Abstract

Addressing data privacy concerns within the insurance sector is essential to harness the full potential of predictive modeling. This research endeavors to explore the generation of synthetic insurance datasets via the implementation of Generative Adversarial Networks (GANs), specifically using the Multi-Categorical Wasserstein GAN with Gradient Penalty (MC-WGAN-GP) variant. A key aspect of this study is to examine how the integration of expert knowledge during the GAN training process influences the quality of the data subsequently generated. To assess the outcomes, we train Extreme Gradient Boosting (XGBoost) models on both real and GAN-generated data, with and without the integration of expert knowledge. The predictive results of these models are then compared for analysis. The findings of this study indicate that the MC-WGAN-GP exhibits a substantial capacity to identify patterns between dependent and independent variables in the datasets. However, it does not achieve perfect replication of the training data yet. Furthermore, the assimilation of expert knowledge during the GAN training phase markedly improves the predictive performance of models trained on larger datasets, although this advantage does not extend to smaller dataset subsets. These results underscore the potential of GANs to address data privacy issues within the insurance industry, while demonstrating the substantial value of incorporating domain-specific expertise into synthetic data generation processes. Simultaneously, the findings suggest an imperative for further research efforts aimed at refining these methodologies and exploring their broad applicability across varied datasets and predictive modeling paradigms.

Improving Synthetic Property & Casualty Data Generation through Expert Input in Generative Adversarial Networks (Version July 9, 2023)

Erman Acar (UvA), Jan Janiszewski (UvA), Georg Maerz (Afairi AG)

July 9, 2023

Github Repository

https://github.com/Afairi/Afairi.io-Research-GAN

1 Introduction

The evolution of artificial intelligence (AI) and data science has ushered transformative changes in various industries, including insurance underwriting and pricing. However, the ongoing challenges associated with data privacy and reliability continue to pose significant hurdles. The insurance industry's data, particularly risk profiles of customers, are both highly confidential and competitively valuable, thereby presenting substantial complications for actuarial departments and limiting potential growth into new markets or product areas. This predicament is exacerbated by stringent privacy legislations, such as the General Data Protection Regulation (GDPR), which restrict data sharing between insurance entities and confine the scope of academic research due to a dearth of accessible realistic data. Although the acquisition of customer risk data from other insurers or market providers is feasible, the substantial effort required for data obfuscation and GDPR compliance renders such exchanges less advantageous (Liapakis, 2018; Cote et al., 2020).

These circumstances underscore the urgent need for reliable and privacy-compliant synthetic data generation methods. Among various AI models, Generative Adversarial Networks (GANs) present a compelling solution, especially in their ability to generate high-quality synthetic data. In particular, the Multi-Categorical Wasserstein Generative Adversarial Network with Gradient Penalty (MC-WGAN-GP) shows significant potential to create realistic, multicategorical data, a characteristic inherent in insurance databases (Kuo, 2019; Cote et al., 2020).

This thesis contributes to the growing field of synthetic data generation in insurance by exploring the impact of integrating the knowledge of actuarial experts during the training of GAN models. To our knowledge, the application of such an approach to GAN training in an insurance context is a not well-explored area. This thesis aims to fill this void (Prentzas and Hatzilygeroudis, 2011, 2016).

Our research provides a detailed examination of the MC-WGAN-GP model for synthetic insurance data generation and examines the effectiveness of including actuarial expert knowledge across different data set sizes. The insights derived from this study could serve as a valuable input for future AI applications within the insurance industry, particularly concerning data privacy issues.

In pursuit of our objective, we propose two main research questions:

- *Research Question 1*: Can a GAN model trained on insurance claim data accurately replicate the underlying distribution and relationship between the dependent (claim count) and independent variables as measured by Poisson Deviance[1]?

[1] The Poisson Deviance, premised on the assumption of Poisson-distributed data, offers a reliable evaluation of a model's fit. It quantifies the congruence between the model and observed data, with lower values indicating superior model performance. For a comprehensive explanation, refer to Section 3.7.1

- *Research Question 2*: Does the inclusion of actuarial expert knowledge into the input data during the training of the GAN model lead to improved preservation of the distribution and relationships in the synthetic data generated by the model?

Based on these research questions, we have the following working hypotheses:

- *Hypothesis 1.1*: The Extreme Gradient Boosting (XGBoost) model built on the data generated by the MC-WGAN-GP provides predictions on the outcome variable (claim count) that are significantly different from those made by a dummy model predicting the average claim count for all respective policies in the data set, as evaluated by Poisson Deviance.
- *Hypothesis 1.2*: Compared to Hypothesis 1.1, the XGBoost model built on the data generated by the MC-WGAN-GP yields predictions that are statistically comparable to those made by a model trained on the original data set, as evaluated by Poisson Deviance.
- *Hypothesis 2.1*: The XGBoost model built on the data generated by the MC-WGAN-GP with expert knowledge included in the training provides more accurate predictions (as evaluated by Poisson Deviance) than the XGBoost model built on the data generated by the MC-WGAN-GP without expert knowledge included.
- *Hypothesis 2.2*: The effect described in Hypothesis 2.1 is even more prominent in a data set subsample of the original data set ($N = 5,000$[2]).

The subsequent sections of this thesis comprise a review of the relevant literature, an overview of the various models and their applications in insurance, a description of our research methodology, and a comprehensive analysis of our experimental results. The thesis concludes with a discussion of the potential implications of our findings and recommendations for future research directions.

2 Related Work

The proliferation of machine learning (ML) and AI has stimulated an increasing interest in the generation of robust synthetic data across various insurance subdisciplines, including pricing, underwriting, and claim management (Balasubramanian et al., 2021). Our study builds on this body of work by exploring the intersection of three crucial areas: ML models in actuarial pricing, data synthesis in insurance, and knowledge representation for insurance. This intersection forms the basis of our methodology, which is based on the understanding that a robust application of these areas could improve the precision, efficacy, and interpretability of insurance models.

2.1 Data Synthesis in Insurance

Early studies on insurance data generation used traditional statistical methods such as resampling to create synthetic data sets (Dichtl et al., 2017; Gabrielli and V. Wüthrich, 2018). However, with the advent of ML and AI, more sophisticated techniques have emerged (Kuo, 2019; Cote et al., 2020; Goodfellow et al., 2014). Introduced by Goodfellow et al. (2014), GANs have revolutionized the generation of synthetic data. The GAN architecture, composed of a generator and a discriminator network in a competitive environment, has enabled the creation of high-quality synthetic data. Despite this, their use in insurance remains limited.

Kuo (2019) first published research on GANs in insurance, demonstrating the potential of the CTGAN algorithm to generate synthetic insurance data. The data generated by the CTGAN successfully mimicked the distributions of the real data. Building on this work, Cote et al. (2020) evaluated different GAN architectures, with the MC-WGAN-GP model emerging as the most effective in capturing both individual variable distributions and their correlations.

In line with these findings, our thesis uses the MC-WGAN-GP model to generate synthetic data based on an insurance dataset, as this particular GAN has been shown to be effective in mimicking both the distributions of single variables and their relationships (Cote et al., 2020).

[2] where N is the data set size

2.2 Knowledge Representation for Insurance

The fusion of expert knowledge within AI models presents a compelling research avenue, known for its propensity to amplify guidance during learning, increase model interpretability, and increase model performance (Prentzas and Hatzilygeroudis, 2007; Marling et al., 2002; Munoz-Avila et al., 1999; Prentzas and Hatzilygeroudis, 2011, 2016). This is particularly essential in sectors such as insurance, where clarity of the rationale of the model and the justification of decisions play a pivotal role (Fissler et al., 2022).

Over the years, the AI and ML landscapes have seen remarkable progress in terms of assimilating expert knowledge into learning models, with a particular emphasis on rule-based systems (Prentzas and Hatzilygeroudis, 2011, 2016). As defined by (Prentzas and Hatzilygeroudis, 2007), these methods fall primarily into two groups: rule-based reasoning (RBR) and case-based reasoning (CBR). RBR offers a generalized comprehension of the domain, whereas CBR encapsulates detailed knowledge. While rule-based systems generate solutions from the ground up, case-based systems take advantage of established scenarios to tackle analogous new cases. Given the diverse strengths and weaknesses of both RBR and CBR, composite or integrated methods that merge the two have led to innovative and potent results (Marling et al., 2002; Munoz-Avila et al., 1999).

Hybrid methodologies can be segmented into three primary categories: sequential, co-, and embedded processing. Sequential processing involves the successive integration of different knowledge representation techniques, culminating in an information flow from the preliminary to the final representation. Co-processing refers to a cooperative approach where the assimilated components concurrently work towards the final output. On the contrary, embedded processing involves embedding a component based on a particular representation into one or more components predicated on another (Prentzas and Hatzilygeroudis, 2007, 2011).

In insurance practice, numerous strategies for incorporating expert knowledge have been examined. As an illustration, (Byczkowska-Lipińska et al., 2009) proposed an expert-knowledge-driven system to assess the potential of insurability in medical insurance, based on expert rules. (Hsieh and Wang, 2011) further extended this research by introducing the Linguistic Descriptions Evaluating Algorithm, a life insurance risk assessment tool based on a multitude of linguistic approaches (e.g., linguistic logic, uncertainty number modeling, fuzzifications, and defuzzification schemes).

Our approach to enhancing the performance of the MC-WGAN-GP model is based on the sequential integration of various knowledge representation methodologies. Initially, actuarial knowledge is incorporated through case-based rules. Subsequently, group-based rules formulated by the actuary are introduced, either supplementing or superseding the previously developed case-based rules. This two-step process aims to improve the quality of our GAN by integrating expert knowledge into the learning mechanism, as detailed in previous research (Prentzas and Hatzilygeroudis, 2007, 2011).

2.3 ML Models in Actuarial Pricing

Different ML models can be used to compare the quality of different data sources (e.g., GAN generated data vs. real data) with each other. Traditionally, GLMs have been widely used for actuarial pricing in nonlife insurance (Parodi, 2014; Goldburd et al., 2020). Despite their flexibility in dealing with various error distributions and their capacity to use link functions, GLMs are intrinsically linear and, as such, struggle to capture intricate nonlinear relationships.

In light of the challenges associated with Generalized Linear Models (GLMs) for actuarial pricing, several studies have explored the potential of alternative ML and deep learning algorithms to augment GLM-based techniques (Wüthrich and Merz, 2019; Schelldorfer and Wuthrich, 2019; Richman and Wüthrich, 2022; Noll et al., 2020). Some actuarial investigations, for instance, have proposed training a feed-forward Artificial Neural Network (ANN) concurrently with a GLM to optimize the latter's performance (Wüthrich and Merz, 2019; Schelldorfer and Wuthrich, 2019; Richman and Wüthrich, 2022). However, a comparative study of ML algorithms demonstrated the superior performance of XGBoost models over both GLM and ANN-based models, specifically in the context of Motor Third-Party Liability (MTPL) risk prediction (Noll et al., 2020).

Motivated by this evidence, our research opts for an XGBoost as the model of choice to train on both real and GAN-generated data. This selection would enable a robust comparison of the quality of the respective data sources.

3 Methodology

3.1 Research Design

In this study, we use a comparative analysis to examine the performance of different synthetic data generation pipelines. Each pipeline incorporates an ML model trained on actual or GAN-generated data, with the inclusion or exclusion of expert knowledge during the training process serving as a key distinguishing factor.

Our research design incorporates five distinct modeling pipelines, each assessed in different sizes of the training set ($N = 5,000$; $N = 433,728$). This approach engenders a comprehensive understanding of the varying behaviors of the models in different data size scenarios (visual representation provided in Figure 1). The individual pipeline configurations are as follows:

- Pipeline 1 employs a baseline dummy model that predicts a uniform average claim count (derived from the training set) for each policy in the test dataset.
- Pipeline 2 utilizes an XGBoost model trained on actual training data to predict the claim count, serving as a reference model.
- Pipeline 3 introduces a GAN trained on the real data. The synthetic data generated by this GAN is then used to train an XGBoost model
- Building on the third pipeline, Pipeline 4 incorporates additional actuarial expert knowledge during the GAN training process by augmenting the training dataset with expert knowledge variables, thereby enhancing the quality of the synthetic data generation process.

In the interest of unbiased evaluation, all pipelines maintain consistent data preprocessing procedures, XGBoost model hyperparameters, and test datasets.

Given that our dependent variable (claim count) follows a Poisson distribution, we opt for the Poisson Deviance as our model performance evaluation metric (Fissler et al., 2022). This choice is justified by the inapplicability of other commonly used metrics in GAN research, such as the root mean squared error (RMSE) and the mean absolute error (MAE), to Poisson-distributed variables (Cote et al., 2020; Kuo, 2019; Fissler et al., 2022).

3.2 Hardware and Computational Resources

The research methodology is designed for parallel execution across various computing platforms, ranging from high-performance supercomputers to local systems. The primary implementation language is Python, in combination with the PyTorch framework.

For high-performance computing, the methodology is tested on the Snellius supercomputer, which significantly accelerated the model training process (SURF, 2021). However, for accessibility and replicability, the methodology is also designed to run on a more modest machine, such as the MacBook Air M2 with an Apple M2 silicon processor. Although the MacBook Air M2's computational capabilities are lower compared to the supercomputer, the training process is still feasible with extended durations.

3.3 Data

The data set for this study is derived from the French motor third-party liability (MTPL) insurance portfolio, available on the OpenML platform (Feurer et al., [n. d.]). This data set comprises 678,013 vehicle insurance policies and twelve distinct variables. These include the policy number (*IDpol*), which is typically associated with a customer, vehicle, or a combination of both, as well as the claim count (*ClaimNb*), measuring the number of claims made by a customer within a specified exposure time frame (also named *Frequency* in the actuarial context). Other variables include total exposure in years (*Exposure*), rea code (*Area*), power of the vehicle (*VehPower*), age of the vehicle in years (*VehAge*), age of the driver in years (*DrivAge*), bonus-malus level (*BonusMalus*), vehicle brand (*VehBrand*), fuel type (*VehGas*), population density (*Density*), and regions in France (*Region*).

In data pre-processing, particular care is taken to ensure consistency between different variables. Any identified inconsistencies are dealt with using appropriate strategies, ensuring the quality of the data.

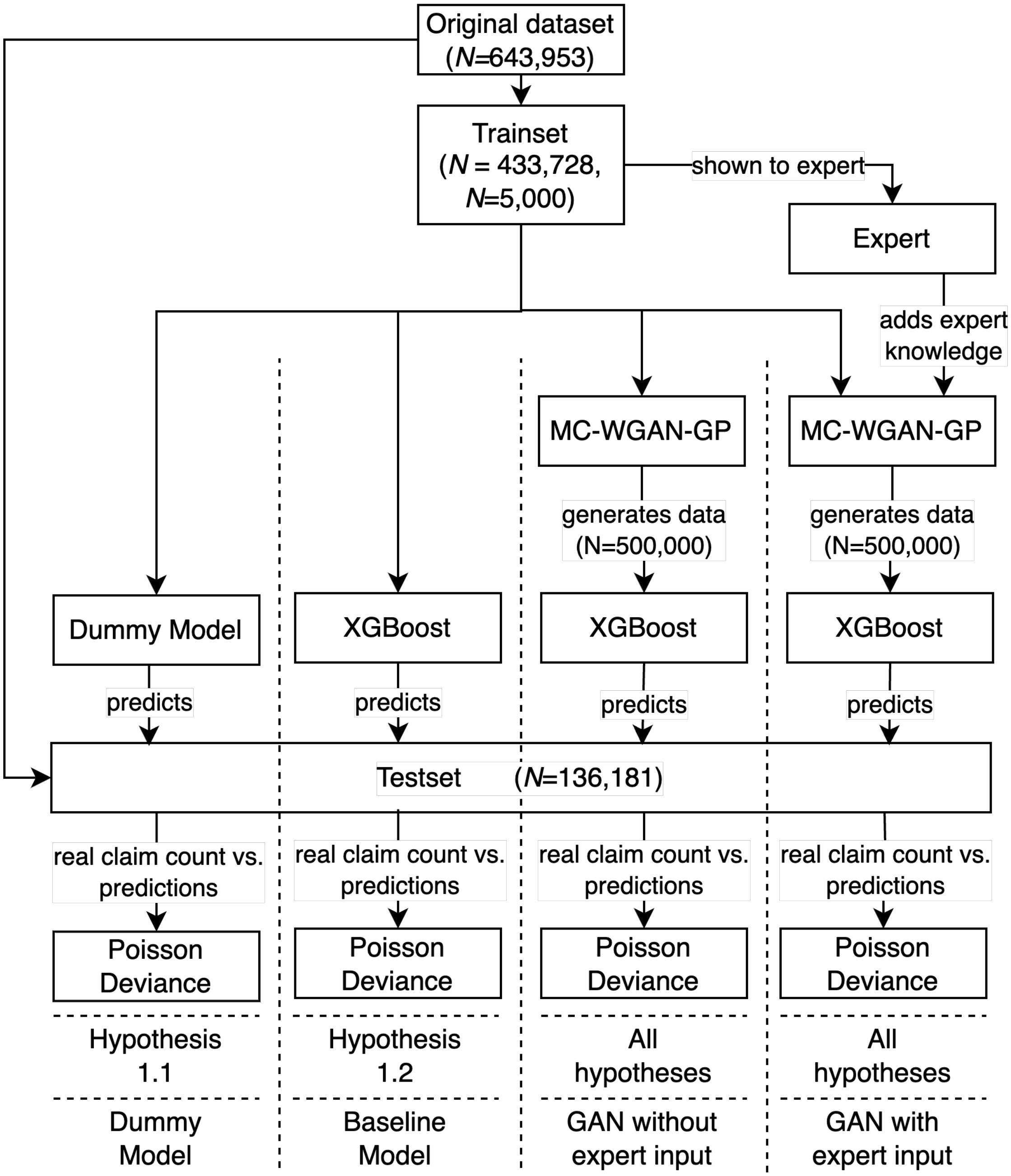

Figure 1: Structure of the different pipelines for each hypothesis

First, the variable *Exposure* deviates from the assumption of expected distribution. Only 24.80% of the policies exhibit the expected exposure of one year, while the majority have exposure durations of less than one year. This is atypical for insurance policy data sets, as policies usually have a standard duration of one year (see Figure 2). As the reasons for this disparity remain unclear, exposure columns are excluded from the XGBoost model training.

Second, in 36.67% of the cases, it is unclear whether the policies are unique, as they share all the necessary characteristics of the policy with each other except for *Exposure*, policy ID (*IDpol*), and claim count (*ClaimNb*). We hypothesize that these non-unique policies represent vehicle fleets, such as leasing vehicles, rental vehicles, or company-owned vehicles. Consequently, these policies are retained in the data set but grouped when the data set is split into training and test sets to ensure the independence of the test set. No other exceptions or inconsistencies are identified during the analysis (see Appendix A for distribution figures).

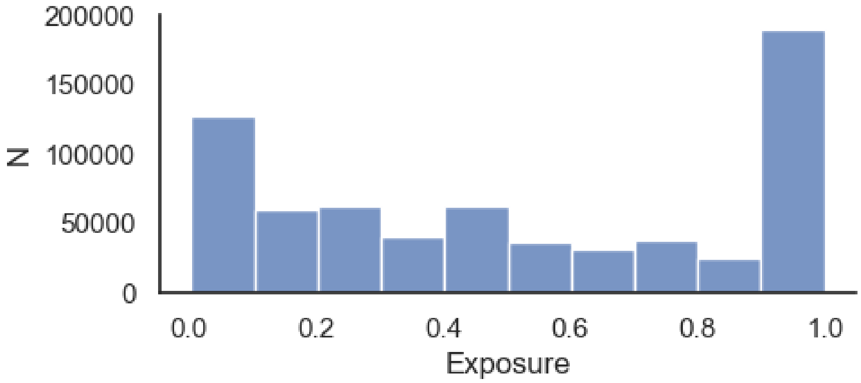

Figure 2: Distribution of exposure values up to one year

The dependent variable *ClaimNb* is further examined. The data set reveals that 95% of the policies ($N = 643,953$) have no claims filed (i.e., $ClaimNb = 0$), while 4.75% ($N = 32,178$) of the policies have one claim filed. A small proportion of policies, amounting to 0.25% ($N = 1,882$), have more than one claim filed, with the maximum number of claims reaching 16 (see Appendix A for a distribution of *ClaimNb* values below 5).

3.3.1 Data Preprocessing

A standardized data preprocessing pipeline is implemented, based on the recommendations of previous research (Noll et al., 2020; Cote et al., 2020). This includes transformations and standardizations of several variables to ensure data quality and comparability between different modeling pipelines.

The common data preparation steps include the following transformations:

- **Policy ID (*IDpol*):** *IDpol* is dropped from the data set because it does not contribute to the modeling process.
- **Claim Count (*ClaimNb*):** The claim count is capped at values exceeding 4 claims; to facilitate GAN training, it is also converted into a categorical variable for GAN training.
- ***Exposure*:** For XGBoost training, the *Exposure* is not used. However, to support GAN training, it is used with values exceeding one year capped for training.
- ***Area*:** The categorical alphabetic representation of the *Area* variable is transformed into a continuous variable.
- **Vehicle Age (*VehAge*):** To avoid excessive skewness in the data, the vehicle's age is capped at 20 years.
- **Vehicle Power (*VehPower*):** To mitigate potential outliers, *VehPower* is capped at values exceeding 9.

- **Driver Age (*DrivAge*):** To limit the influence of extreme values, the age of the driver (*DrivAge*) is capped at 90 years.
- ***BonusMalus*:** As per prior recommendations, *BonusMalus* levels exceeding 150 are capped.
- ***Density*:** To alleviate the impact of skewed distributions, a logarithmic transformation is applied to the *Density*.

Following the data pre-processing steps, the numerical features are standardized, ensuring a consistent scale across all models. Categorical variables are encoded using one-hot vectors, the dimensionality of the vector corresponding to the number of unique categories, as suggested by previous research (Cote et al., 2020). After preprocessing, the data are partitioned into training ($N = 433,728$), validation ($N = 74,044$), and test set ($N = 136,181$).

3.4 Incorporating Expert Knowledge

We postulate that the incorporation of expert knowledge into the GAN model can reduce overfitting, thus improving the quality of the synthetic data generated. Overfitting is a significant issue in risk modeling, especially when there is an imbalance between positive and negative outcomes. This imbalance often leads to biases and errors in estimation. The introduction of expert knowledge, specifically the accurate depiction of the relationships between dependent and independent variables, can potentially rectify these inaccuracies (Yazici et al., 2020).

For this thesis, we obtain domain-specific expertise from a certified actuarial professional serving as the Head of the Motor Third-Party Liability (MTPL) Department at Generali Insurance in Germany. We adopt a multi-step process to effectively integrate her expertise into our research framework. Our proposed rule integration methodology in this study incorporates aspects of both the embedded processing and co-processing strategies for knowledge representation and is described in detail in the following subsections. See Appendix C for a complete account of the expert knowledge incorporated, including all specified rules and adjustments.

3.4.1 Scope Definition

During the initial phase, the actuarial expert identifies variables that have a significant relationship with the dependent variable (claim count). These variables, which include *Density*, *DrivAge* (Driver Age), *BonusMalus*, and *VehAge* (Vehicle Age), were defined by the actuary as areas within the scope of her knowledge.

3.4.2 Idea Generation

During this phase, a range of Generalized Linear Models (GLMs) are trained on the training data to explore potential relationships between each respective variable and the claim count. The expert actuary guides the selection of models to be trained. For example, for *VehAge*, multiple polynomial GLMs with log link and different degrees were trained to encapsulate the polynomial relationship between claim count and vehicle age. An illustration of such a model, showing the relationship $ClaimCount = \beta_0 + \beta_1 \times VehAge + \beta_2 * VehAge^2$, can be found in Figure 3.

3.4.3 Representation Selection

Utilizing visualizations of the models created in the previous step, the actuary identifies the representation that aligns most accurately with her domain knowledge and expectations. This involves selecting the model that best characterizes her understanding of the relationship between the claim count and a given independent variable (e.g., *VehAge*). Consequently, the expert picks various GLM models that optimally illustrate the relationships between the claim count and the independent variables, as exemplified by the case of vehicle age in Figure 4.

3.4.4 Representation Adjustment

During this phase, the actuarial expert implements modifications to the representations of the established relationships for the selected variables. Consider the case of the *VehAge* variable. Drawing from

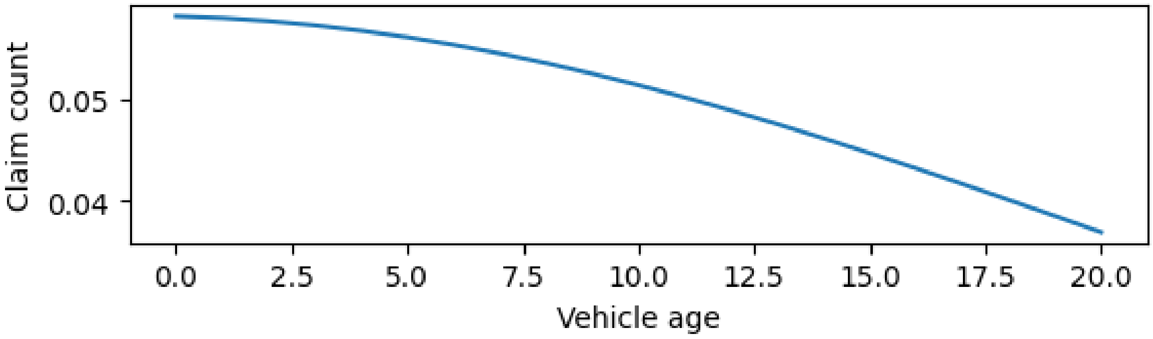

Figure 3: One of multiple potential GLMs to capture the relationship between vehicle age and claim count

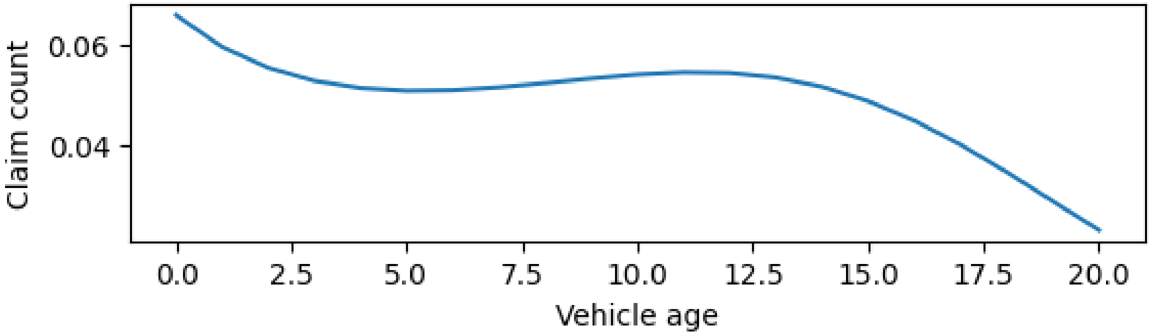

Figure 4: The final GLM chosen to capture the relationship between vehicle age and claim count

her experience, the actuary posited that the likelihood of filing a claim does not increase for relatively new vehicles (those under 5 years of age). Consequently, she suggested maintaining a constant average claim count of 0.05 for vehicles within this age bracket. This expert insight is codified as a rule that superseded the model's original predictions for this particular subgroup. By formulating rules and modifications for certain variable ranges or categories, she is able to incorporate nuanced changes that account for specific patterns or tendencies discerned in the data.

3.4.5 Additional Rules

For each variable, the actuary could propose additional rules which are to be incorporated into the data set. This provision enables us to capture further insights that may have been overlooked or not adequately represented by strictly data-driven approaches. For example, the expert recommended the introduction of a variable designed to differentiate policies of customers who possessed a *BonusMalus* below 100 from those with a *BonusMalus* exceeding 100. This distinction provides a refined perspective on the potential impact of *BonusMalus* on claim count (see Appendix C for all additional rules).

3.4.6 Integration of Expert Knowledge in the GAN

To incorporate the expert knowledge into the GAN training, all the gathered expert knowledge is standardized and systematically assigned to the corresponding customers in our data set. This mapping is based on the values of the independent variables that each customer possesses and results in the creation of multiple new columns within the GAN training data set, at least one for each variable.

The enriched training data set is then used to train the discriminator within our GAN model (see Figure 1). This approach ensures that the discriminator is trained with both the original data and the added expert knowledge, thereby bolstering its performance and accuracy.

Importantly, this method only indirectly trains the generator with expert knowledge, since the generator does not have direct access to this knowledge. However, it benefits from this knowledge

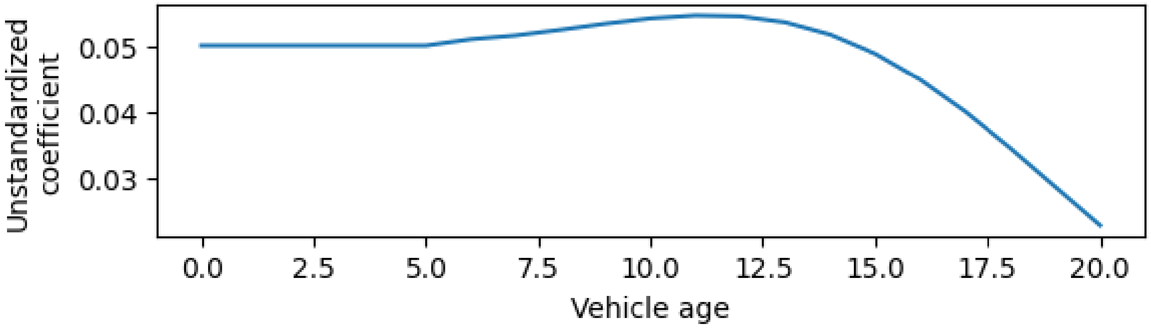

Figure 5: The final relationship representation with Vehicle age <= 5.00 overwritten by actuarial decision to keep the coefficient at 0.05

through the improved feedback received from the enhanced discriminator. This indirect mode of knowledge inclusion harnesses the adversarial dynamic within the GANs, resulting in the generator being guided towards generating synthetic data that not only retains the inherent patterns present in the original data set, but also incorporates the additional insights provided by the expert knowledge. Thus, the synthetic data produced by the generator are reflections of both the original data structure and the specialized insights of the expert.

3.5 Utilization of the Generative Adversarial Network

Informed by the research conducted by Cote et al. (2020) who showed that the MC-WGAN-GP outperforms other GAN architectures, our investigation focused on enhancing the performance of the Multicategorical Wasserstein Generative Adversarial Network with Gradient Penalty (MC-WGAN-GP).

GANs comprise a pair of adversarial neural networks: the generator, which produces synthetic data intended to emulate the real data, and the discriminator, which determines the authenticity of the data points. Mutual competition between these two networks facilitates the improvement of the data quality of the synthetic data, achieving greater similarity to the actual data (Cote et al., 2020).

3.5.1 Wasserstein Generative Adversarial Network

The Wasserstein Generative Adversarial Network (WGAN) addresses the prevalent challenge of training instability often encountered in traditional GANs. This variant replaces the typical binary classification discriminator with a critic that generates real-valued outputs and uses the Wasserstein distance to compute the distance between real and generated distributions (Arjovsky et al., 2017).

3.5.2 Multicategorical Wasserstein Generative Adversarial Network with Gradient Penalty

Our MC-WGAN-GP model expands on the WGAN-GP structure and incorporates the method of handling multicategorical variables suggested by Camino et al. (2018). In our design, every categorical variable is processed through a dense layer, followed by a softmax activation layer in the generator. The outputs of these layers are subsequently concatenated to produce the final output of the generator (Cote et al., 2020).

3.5.3 Adjustments to the models

Our research on the best MC-WGAN-GP architecture involves various modifications to the model, including adjustments to the final layer of the generator, the introduction of a different penalty layer, and the replacement of the discriminator with a critic. Of these modifications, the incorporation of a sigmoid layer as the last layer of the critic (that is, transforming it into a discriminator) demonstrated notable enhancements in the performance of the MC-WGAN-GP model. Consequently, the scope of our hyperparameter search expanded to include the analysis of substituting the critic with a discriminator within the GAN training process.

3.5.4 Hyperparameter Optimization

In the effort to optimize the hyperparameters, the model architecture suggested by Camino et al. (2018) is utilized as a reference point.

To determine the optimal hyperparameters for the MC-WGAN-GP, a grid search technique is applied. The selection of the best hyperparameters is mainly informed by previous research (Cote et al., 2020), given the resource constraints associated with training, and can be found in the study GitHub repository. The validation of these hyperparameters is performed using the XGBoost model, which is previously defined on the validation set. The MC-WGAN-GP that generated the data leading to the most accurate GLM predictions is identified as the optimal model.

The scope of our grid search included parameters such as batch size, loss penalty, batch normalization decay for the discriminator and generator, discriminator's leaky ReLU's parameter, noise size, the ratio of discriminator updates per generator update, L2 regularization for the discriminator, and sizes of the discriminator's hidden layers. In particular, the standard leaky ReLU activation function, used as the final layer of the discriminator, is replaced with a sigmoid function for certain grid search iterations.

Experiments included changes to the model and training protocol, such as replacing the last generator layer from a leaky ReLU layer to a sigmoid layer and omitting the gradient penalty in some iterations. Attempts are also made to balance the positive and negative cases in the data set by undersampling policies without claims for the initial 500 epochs of the GAN training process. Unfortunately, these changes did not enhance the quality of the model's data generation.

Training of the MC-WGAN-GP is carried out for 15,000 epochs in mini-batches, utilizing the binary cross-entropy loss function. Both the generator and the discriminator/critic shared the same zero-sum objective function (Cote et al., 2020).

The final structure of the MC-WGAN-GP generator used in this study consists of three fully connected layers, supplemented with batch normalization and ReLU, followed by a fully connected layer. The terminal layer utilizes a leaky ReLU for continuous variables and a Gumbel function for categorical variables. The final architecture of the discriminator comprises two fully connected layers, enhanced with a ReLU function and terminated by a sigmoid activation function (refer to Figure 6 for details).

Hyperparameter tuning is performed across two different data set sizes, large and small, to assess the scalability and efficiency of our model. More details on these data sets are available in the *Data* section.

The performance results of the model with different hyperparameters on large and small data sets will be presented in the subsequent results section.

Appendix D contains the optimal parameters identified for each of the hypotheses.

3.6 XGBoost Model

To assess the quality of the different data sets (real and generated by the GAN models), we trained an XGBoost model, a popular choice for structured and tabular data, on these data sets. Various versions of the same XGBoost model, all retaining consistent hyperparameters, are trained on different data sets. These data sets included the synthetic data generated by the GAN model with and without expert knowledge, as well as the original data set.

In line with common practice in the insurance sector, we assumed that the claim count follows a Poisson distribution (Noll et al., 2020). As such, an XGBoost model presuming a Poisson distribution is trained on the input data, maintaining the same model architecture for both synthetic and real data sets.

Instead of undertaking a comprehensive grid search for the XGBoost hyperparameters, we used the findings of previous research that had already identified optimal parameters to train the XGBoost model on our data set. Therefore, we conducted only a limited grid search, comparing the performance of the hyperparameters suggested by Martínez de Lizarduy Kostornichenko (2021) with those by König and Loser (2020) on the large training data set. Appendix E contains the optimal parameters identified.

This systematic approach to model development and evaluation provides a robust method for comparing the quality of the data sets generated by our GAN models. By training and testing the XGBoost model on the synthetic and real data sets, we can objectively assess the resemblance of the generated data to the original data set. Moreover, this comparison allows us to evaluate the impact of expert knowledge on the quality of the synthetic data produced by the GAN models, highlighting the benefits and potential limitations of this approach.

3.7 Evaluation

3.7.1 Poisson Deviance

The assessment of the predictive efficacy of our model is carried out by applying the Poisson Deviance. This statistical measure is frequently used to evaluate the quality of fit in Poisson regression models, especially in contexts involving count data (Fissler et al., 2022). Poisson Deviance furnishes a comparison between the fit offered by the model under examination and that offered by a 'perfect' model. The latter serves as an idealized benchmark, representing a model that would perfectly predict the observed data.

Within the framework of our investigation, the calculation of the Poisson Deviance took the form of the following equation:

$$D = 2 \sum \left[y_i \log \left(\frac{y_i}{\hat{\mu_i}} \right) - (y_i - \hat{\mu_i}) \right],$$

where y_i represents the observed data (ground truth) and $\hat{\mu_i}$ represents the prediction of the model.

A lower Poisson Deviance value indicates a more favorable fit of the model to the data, thereby denoting superior model performance; a Poisson Deviance value of 0 indicates an ideal fit. Notably, while the Poisson Deviance is sensitive to the mean of the distribution, it does not heavily penalize errors that exhibit a bias above the mean.

3.7.2 Evaluation Design

The proposed methodology's efficacy is appraised via comparative performance analysis of identical XGBoost models, each trained on one of three distinct data sets - the original data set, a synthetic data set generated by the MC-WGAN-GP that incorporates expert knowledge, and another synthetic data set produced by the MC-WGAN-GP without such expertise. For the original data set, we created 100 unique bootstrap versions for both model training and evaluation. In the case of synthetic data, the GAN is used to generate 100 separate iterations of 100,000 synthetic policies, each of which is used for model training. Subsequent evaluations of these trained models are performed against the 100 bootstrap versions of the test set.

To ensure the reliability of our results, a one-sided, two-sample t-test is adopted to verify both hypotheses. Poisson deviance is computed for each of the 100 bootstrap versions of the test set for each pair of variables under consideration. A p-value below the conventional significance threshold of 0.05 is interpreted as grounds to reject the null hypothesis in favor of the alternative.

Our choice of a one-sided t-test was based on the main objective of the study, namely to determine whether the GAN can produce synthetic data that do not underperform the original data set in terms of predictive capacity (Hypothesis 1) and whether the inclusion of expert knowledge can increase the quality of the synthetic data (Hypothesis 2).

In particular, the independence and normality assumptions of the t-test are considered met in our scenario. The bootstrap versions of the test set are independent, satisfying the independence requirement. Invoking the Central Limit Theorem justified the normality assumption given the large number of bootstrap versions (Liapounoff, 1900).

For hypothesis testing, our attention is focused on the models that demonstrated superior performance during the hyperparameter search. In this context, we define $P_{ne,l}$ and P_0 as the Poisson Deviance for predictions made by the XGBoost model trained on synthetic data without expert knowledge and the original data, respectively. Hypothesis 1, which solely concerns the MC-WGAN-GP trained on the large data set without expert knowledge, frames the null and alternative hypotheses as follows:

$$H_0 : P_{ne,l} = P_0$$

$$H_a : P_{ne,l} < P_0$$

For Hypothesis 2, we introduce $P_{e,s}$, $P_{e,m}$, and $P_{e,l}$, representing Poisson Deviances from predictions by the model trained on synthetic data, designed by the GAN trained with expert knowledge for small, medium, and large data sets, respectively. In a similar vein, $P_{ne,s}$, $P_{ne,m}$, and $P_{ne,l}$ symbolize the Poisson Deviances from the predictions of the model trained on synthetic data without expert knowledge for the small, medium, and large data sets, respectively. Hence, for Hypothesis 2, the null and alternative hypotheses are as follows:

$$H_0 : P_{e,s} = P_{ne,s}; P_{e,l} = P_{ne,l}$$
$$H_a : P_{e,s} < P_{ne,s}; P_{e,l} < P_{ne,l}$$

Hypothesis 2 is considered validated only if all the conditions delineated in the alternative hypothesis are satisfied for all data sets.

Given the nature of the evaluation, potential limitations encompass the inherent assumptions of the selected hypothesis testing methodology, reliance on the XGBoost model's predictive accuracy, and the use of a singular, albeit intricate, method for synthetic data set generation. However, these limitations are mitigated by the rigor of the evaluation procedure and the robustness of the chosen metrics, affording a comprehensive appraisal of the model performance.

4 Results

The findings of our study provide evidence in support of Hypotheses 1.1 and 2.1. A comparative analysis of the XGBoost models, particularly their performance metrics, revealed nuanced insights, particularly related to the effect of synthetic data generation and the inclusion of expert knowledge.

Hypothesis 1.1 posited that the XGBoost model built on the data generated by the MC-WGAN-GP would produce significantly different predictions on the outcome variable, namely the claim count, compared to a dummy model predicting the average claim count for all respective policies in the data set. As evidenced by our results, this hypothesis is supported [$t(49) = -3.384$, $p < 0.001$]. We found a significant difference between the performance of the models, with the XGBoost model trained on GAN data producing a Poisson Deviance (PD) of 0.317 ($SD = 0.002$), which is statistically different from the dummy model ($PD = 0.319$, $SD = 0.002$).

Hypothesis 1.2, on the other hand, predicted that the XGBoost model built on synthetic data would have performance metrics statistically indistinguishable from those of a model trained on the original data set. Contrary to this hypothesis, we found that the Poisson Deviances were statistically different with a Poisson Deviance of 0.317 ($SD = 0.002$) for the model trained on GAN data versus a Poisson Deviance of 0.304 ($SD = 0.002$) for the model trained on the original data set, $t(49) = 39.789$, $p < 0.001$. See Figure 7 and Table 1 for a comparison of the performance of the model.

When it comes to the use of expert knowledge in the GAN training phase, **Hypothesis 2.1** expected that the XGBoost model built on synthetic data generated with expert knowledge would provide more accurate predictions than the model built on data generated without expert knowledge. Our results indeed indicate a significant improvement in the model's performance when it is trained on synthetic data generated with expert knowledge. The Poisson Deviance decreased to 0.319 ($SD = 0.002$), better than the value observed for the model trained without expert knowledge, $PD = 0.321$, $SD = 0.002$, $t(49) = -3.499$, $p < 0.001$, (Figure 8 provides a visual representation of Poisson Deviances for Hypothesis 2.1).

Training set size	Data source	Poisson Deviance
433,728	Dummy model	0.319 (0.318, 0.320)
433,728	Synthetic	0.317 (0.316, 0.318)
433,728	**Real data**	**0.302 (0.301, 0.302)**

Table 1: Comparison of the different pipelines for research question 1 on their main metrics with confidence intervals (in brackets); best performing model in bold

Lastly, **Hypothesis 2.2**, asserting that the positive effect of expert knowledge would be more prominent on a subsample of the original data set, is not corroborated by the results, as shown by Poisson Deviance, $t(49) = 21.222$, $p = 1.000$. This suggests that while expert knowledge improves the quality of synthetic data, its impact may not be as pronounced when the size of the data set is reduced. The Poisson Deviance increased to 0.372 ($SD = 0.015$), worse than the value observed for the model trained without expert knowledge ($PD = 0.326$, $SD = 0.002$). However, since both models do not outperform the dummy model ($PD = 0.319$), these results must be carefully interpreted (Figure 8 provides a visual representation of Poisson Deviances for Hypothesis 2.2).

To dig into the dynamics of our models, we conducted an in-depth analysis centered on the variable distributions within a sample generated by the GAN models. In particular, the GAN model, which is trained on data devoid of expert input, appeared to falter when it came to generating policies associated with 3 or 4 claims. In stark contrast, the GAN model that incorporated expert knowledge demonstrated the capability to successfully generate such policies (see Appendix A for the detailed distribution of *ClaimNb*).

In addition, we performed an analysis that focused on the mean error per ventile. We partitioned the data into ten distinct ventiles based on the predictions rendered by each model, with the highest and lowest claim count predictions falling into the 10th and 1st ventiles, respectively. Subsequently, we computed the residuals between the model predictions and the actual claim counts. These residuals were then collated by prediction ventile group, yielding average residuals per ventile for each model.

The analysis revealed noteworthy findings. With the larger dataset, the model that integrated expert knowledge outperformed its counterpart, particularly in its predictions for the ventiles with exceptionally high projected claim counts. In particular, this model did not overestimate the claim count for these groups to the same extent as the model without expert input (see ventiles 19 and 20 in Figure 10). This observation corroborates our underlying hypothesis, suggesting that the incorporation of expert knowledge can mitigate the risk of model overfitting.

In contrast, an examination of the smaller dataset yielded a starkly different pattern. The GAN model bereft of expert input resulted in an XGBoost model that exhibits a consistent tendency toward underprediction of claim counts. This behavior suggests potential underfitting within the GAN, which in turn might have impeded the generation of rare claim events. In contrast, the XGBoost model constructed from GAN data incorporating expert input manifested a pronounced tendency to overestimate claim counts. This overestimation is particularly evident in ventiles associated with higher predicted claim counts (see ventile 10 to 20 in Figure 11), which might suggest a scenario of overfitting in which the model generated an inordinately large number of customers with filed claims.

The confidence intervals for the aforementioned comparisons all supported the findings, remaining consistent within the established intervals (see Table 2). A detailed interpretation of these results, along with relevant visual representations, is provided in the following sections.

Training set size	Data source	Expert knowledge	Poisson Deviance
433,728	Synthetic	No	0.321 (0.320, 0.321)
433,728	**Synthetic**	**Yes**	**0.319 (0.319, 0.320)**
5,000	**Synthetic**	**No**	**0.321 (0.320, 0.321)**
5,000	Synthetic	Yes	0.372 (0.368, 0.377)

Table 2: Comparison of the different pipelines for research question 2 on their main metrics with confidence intervals (in brackets); best performing models in bold

5 Discussion

The present research makes a substantial contribution to the growing corpus of research focusing on synthetic data generation and data privacy, specifically within the insurance industry's context. Using MC-WGAN-GP, our study underscores the capabilities of GANs to produce synthetic insurance data sets that maintain the inherent structure and relationships evident in the original data. When the synthetically generated data were utilized to train XGBoost models, they facilitated claim count predictions reflective of the dependent and independent variables' interplay to a significant extent. Although not perfectly capturing all nuances of the original data set's intricate relationships, the quality of prediction is sufficiently high. This underscores the potential of GANs in addressing data privacy concerns while maintaining data utility.

Our investigation also brings to light the potential role of expert knowledge incorporation in enhancing the quality of synthetic data during GAN training, as evidenced by the augmented predictive accuracy of models trained on such data. This serves to accentuate the integral role of domain-specific knowledge in bolstering the performance of machine learning techniques, including GANs.

The research further illuminates potential drawbacks in integrating expert knowledge into GAN training. Specifically, the study reveals that the beneficial impact of the incorporation of expert knowledge on model performance is attenuated when the size of the data set is reduced. This suggests the existence of a certain data set size threshold necessary for the optimal manifestation of expert knowledge benefits. We observed an unexpected decline in model performance when the MC-WGAN-GP is trained on smaller data sets supplemented with expert knowledge, emphasizing the importance of adequate data set size.

Understanding the relationship between the effectiveness of expert knowledge integration and data set size emerges as a key area for future exploration. Identifying the critical thresholds in data set size that influence expert knowledge's beneficial effects could be advantageous. Moreover, alternative methods of integrating expert knowledge into the training process warrant exploration. For instance, direct inclusion of expert knowledge through the addition of a dedicated layer in the generator architecture represents a promising alternative. However, potential complications associated with the backpropagation phase of the generator's training require careful consideration and navigation.

Despite these challenges, the study offers several strengths, including gaining a comprehensive understanding of the functioning of MC-WGAN-GP under different scenarios of inclusion of expert knowledge and data set sizes. Additionally, the systematic and comprehensive integration of expert knowledge greatly enhanced the modeling process by refining the relationships between dependent and independent variables. We also established a streamlined process to capture the knowledge of actuaries, a strategy that could be applicable to other AI research areas.

The findings have the potential to shape more accurate and reliable data generation processes in the insurance industry, thereby facilitating data sharing among insurers and researchers and creating non-confidential data sets for research purposes. Our study's exploration of incorporating expert knowledge in GAN training may also lead to advancements in GAN models across various industries, thereby paving the way for more sophisticated data-driven decision-making processes.

6 Conclusion

Our research successfully navigates the intersection of synthetic data generation, data privacy, and the insurance industry, offering notable insights. Using the MC-WGAN-GP, we have demonstrated the potential of synthetic insurance datasets to closely emulate the inherent structure and relationships of the original data, thus underlining the importance of GANs in data privacy endeavors.

Crucially, our findings elevate the importance of expert knowledge in bolstering the quality of synthetic data generation, setting the stage for further exploration of integration of domain-specific expertise in machine learning methodologies. We revealed an intricate interplay between the effectiveness of this integration of expert knowledge and the size of the training dataset, thereby identifying a new direction for future research.

These findings offer potential avenues for generating non-confidential datasets and promote improved data sharing practices among various stakeholders. The integration of expert knowledge into GAN training, as demonstrated in this research, heralds a future of enhanced data-driven decision making across various industries. Future research, based on our findings, should contribute to expanding the boundaries of understanding GAN applications, not just within the insurance industry but also across broader spheres.

In summary, this research lays the foundation for a future where synthetic data generation, powered by expert knowledge, is not only feasible but delivers optimal results. This study paves the way for the systematic and beneficial integration of domain expertise into GAN training, promising significant improvements in results.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*. PMLR, 214–223.

Ramnath Balasubramanian, Ari Libarikian, and Doug McElhaney. 2021. https://www.mckinsey.com/industries/financial-services/our-insights/insurance-2030-the-impact-of-ai-on-the-future-of-insurance

Liliana Byczkowska-Lipińska, Mariusz Szydło, and Piotr Lipiński. 2009. *Expert Systems in the Medical Insurance Industry.* Springer Berlin Heidelberg, Berlin, Heidelberg, 189–199. https://doi.org/10.1007/978-3-642-04462-5_19

Ramiro Camino, Christian Hammerschmidt, and Radu State. 2018. Generating multi-categorical samples with generative adversarial networks. *arXiv preprint arXiv:1807.01202* (2018).

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).

Hubert Dichtl, Wolfgang Drobetz, and Martin Wambach. 2017. A bootstrap-based comparison of portfolio insurance strategies. *The European Journal of Finance* 23, 1 (2017), 31–59.

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

Tobias Fissler, Christian Lorentzen, and Michael Mayer. 2022. Model comparison and calibration assessment: user guide for consistent scoring functions in machine learning and actuarial practice. *arXiv preprint arXiv:2202.12780* (2022).

Andrea Gabrielli and Mario V. Wüthrich. 2018. An individual claims history simulation machine. *Risks* 6, 2 (2018), 29.

Mark Goldburd, Dan Khare, Anand amd Tevet, and Dmitriy Guller. 2020. Generalized Linear Models for Insurance Rating.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

Chih Hsun Hsieh and Paul P Wang. 2011. Linguistic evaluation system and insurance. *New Mathematics and Natural Computation* 7, 03 (2011), 383–411.

Daniel König and Friedrich Loser. 2020. https://www.kaggle.com/code/floser/glm-neural-nets-and-xgboost-for-insurance-pricing

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

Xenofon Liapakis. 2018. A GDPR Implementation Guide for the Insurance Industry. *International Journal of Reliable and Quality E-Healthcare (IJRQEH)* 7, 4 (2018), 34–44.

Alexandre Liapounoff. 1900. Sur une proposition de la théorie des probabilités. *Izvestija Rossijskoj akademii nauk. Serija matematičeskaja* 13, 4 (1900), 359–386.

Cynthia Marling, Mohammed Sqalli, Edwina Rissland, Hector Muñoz-Avila, and David Aha. 2002. Case-based reasoning integrations. *AI magazine* 23, 1 (2002), 69–69.

Viktor Martínez de Lizarduy Kostornichenko. 2021. *Comparative performance analysis between Grandient Boosting models and GLMs for non-life pricing.* Master's thesis.

Héctor Munoz-Avila, David W Aha, Len Breslow, and Dana Nau. 1999. HICAP: An interactive case-based planning architecture and its application to noncombatant evacuation operations. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*. 870–875.

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

Pietro Parodi. 2014. *Pricing in general insurance.* CRC press.

Jim Prentzas and Ioannis Hatzilygeroudis. 2007. Categorizing approaches combining rule-based and case-based reasoning. *Expert Systems* 24, 2 (2007), 97–122.

Jim Prentzas and Ioannis Hatzilygeroudis. 2011. Neurules-a type of neuro-symbolic rules: An overview. In *Combinations of Intelligent Methods and Applications: Proceedings of the 2nd International Workshop, CIMA 2010, France, October 2010*. Springer, 145–165.

Jim Prentzas and Ioannis Hatzilygeroudis. 2016. Assessment of life insurance applications: an approach integrating neuro-symbolic rule-based with case-based reasoning. *Expert Systems* 33, 2 (2016), 145–160.

Ronald Richman and Mario V Wüthrich. 2022. LocalGLMnet: interpretable deep learning for tabular data. *Scandinavian Actuarial Journal* (2022), 1–25.

Jürg Schelldorfer and Mario V Wuthrich. 2019. Nesting classical actuarial models into neural networks. *Available at SSRN 3320525* (2019).

SURF. 2021. Dutch National Supercomputer Snellius. https://www.surf.nl/en/dutch-national-supercomputer-snellius

Mario V Wüthrich and Michael Merz. 2019. Yes, we CANN! *ASTIN Bulletin: The Journal of the IAA* 49, 1 (2019), 1–3.

Yasin Yazici, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, and Vijay Chandrasekhar. 2020. Empirical analysis of overfitting and mode drop in gan training. In *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 1651–1655.

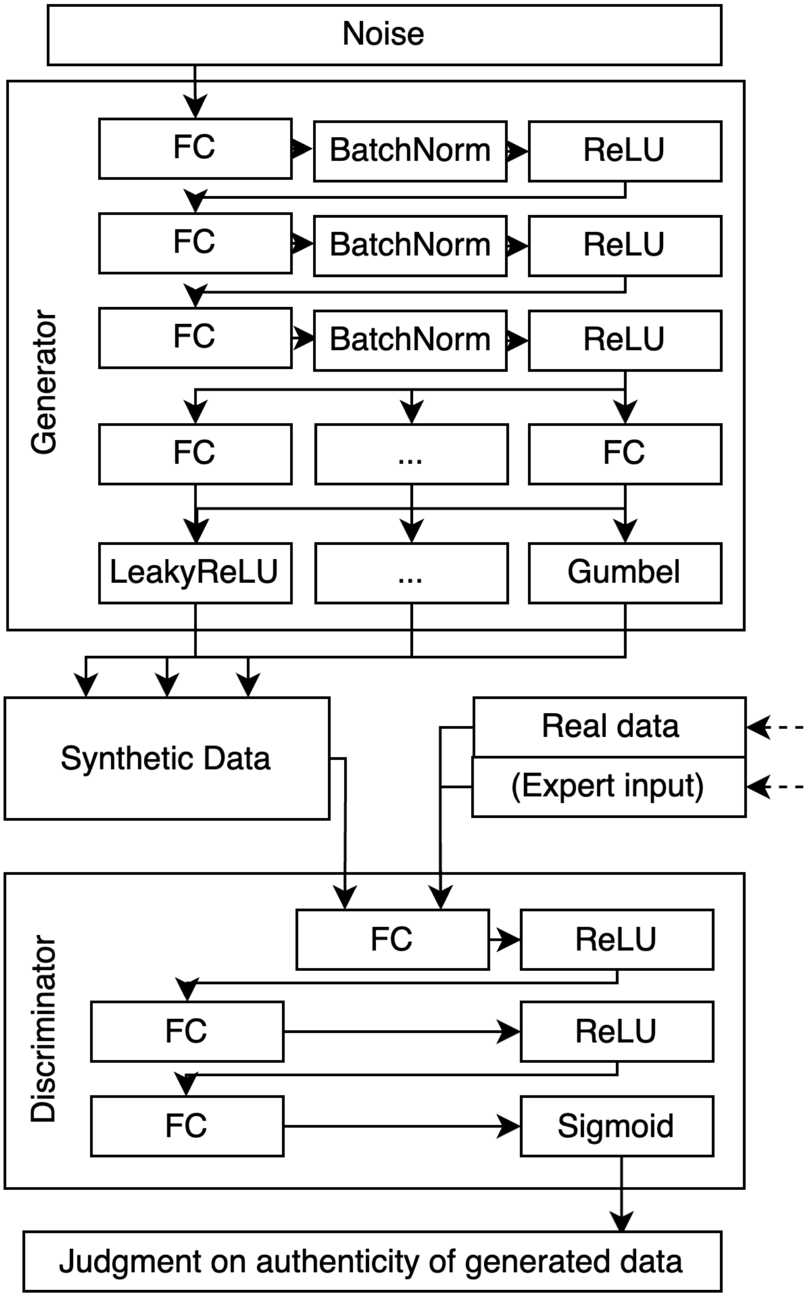

Figure 6: The architecture of the final version of the MC-WGAN-GP

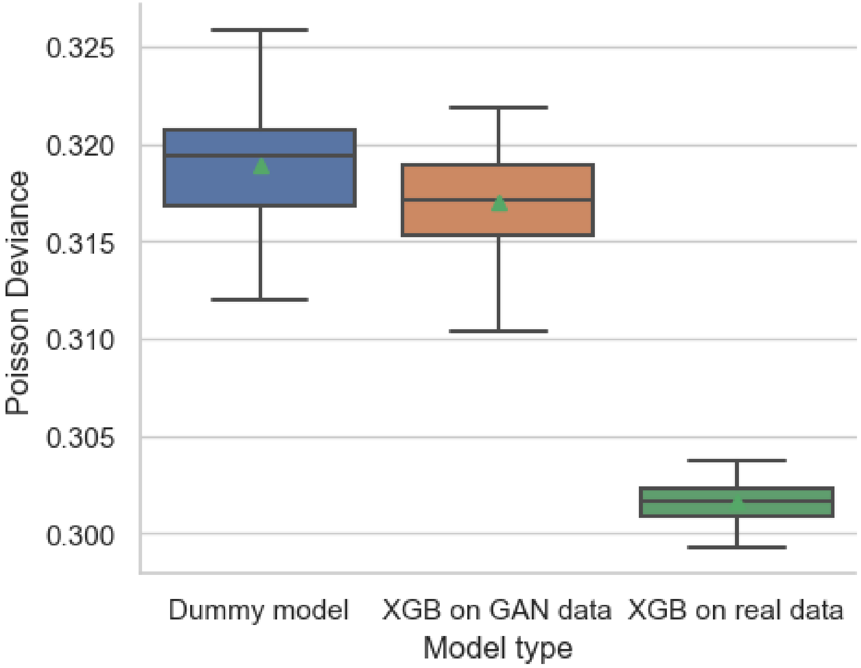

Figure 7: Research Question 1: Comparison of the Poisson Deviance across the different models compared. The plot's box encapsulates the interquartile range with the median indicated by a central black line. 'Whiskers' represent the distribution outside the interquartile range. The mean is denoted by a green triangle. Lower Deviance values indicate a better fit.

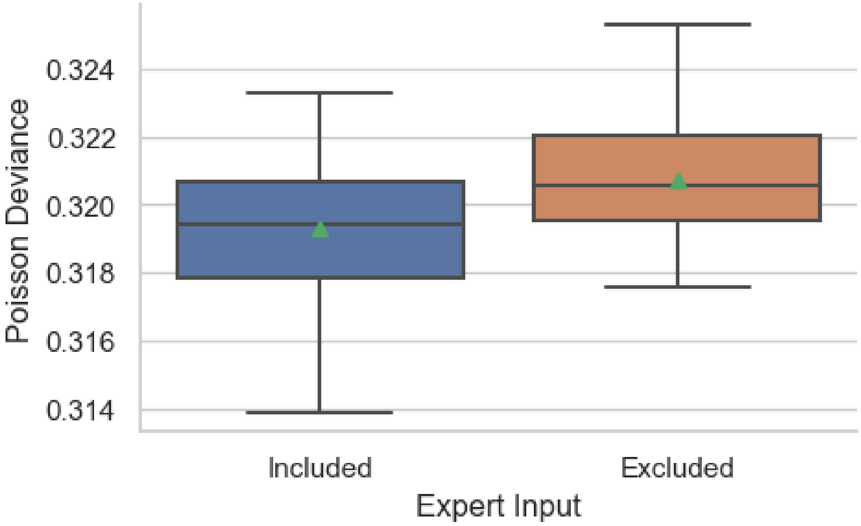

Figure 8: Hypotheis 2.1: Comparison of the Poisson Deviance across the XGBoost models trained on large data set. The plot's box encapsulates the interquartile range with the median indicated by a central black line. 'Whiskers' represent the distribution outside the interquartile range. The mean is denoted by a green triangle. Lower Deviance values indicate a better fit.

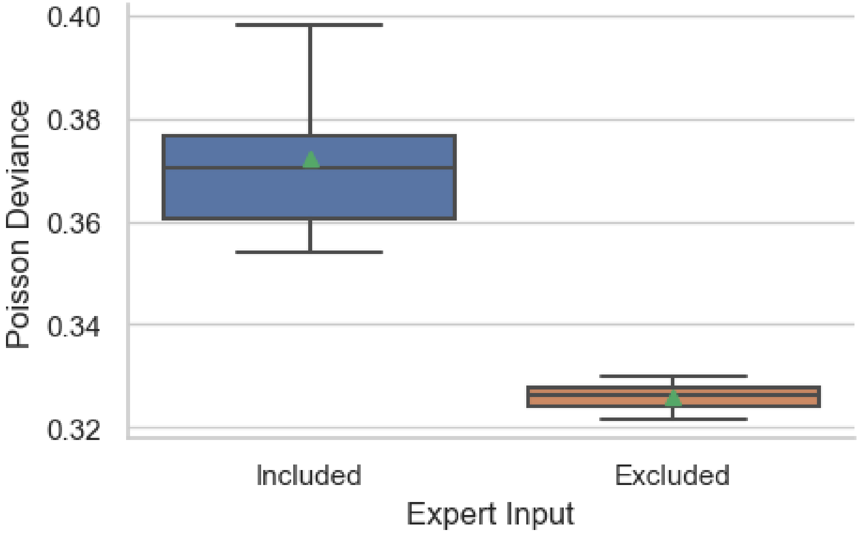

Figure 9: Hypotheis 2.2: Comparison of the Poisson Deviance across the XGBoost models trained on small dataset. The plot's box encapsulates the interquartile range with the median indicated by a central black line. 'Whiskers' represent the distribution outside the interquartile range. The mean is denoted by a green triangle. Lower Deviance values indicate a better fit.

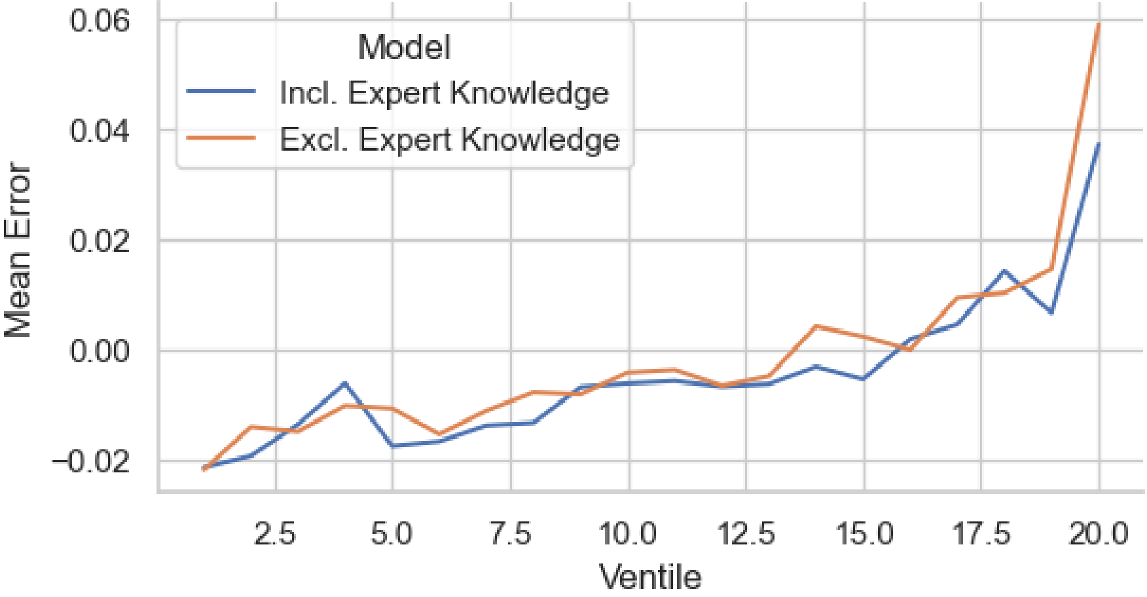

Figure 10: MAE per ventile (large dataset scenario). Split into ventiles was built on predicted claim count values by the XGBoost model trained in respective pipeline.

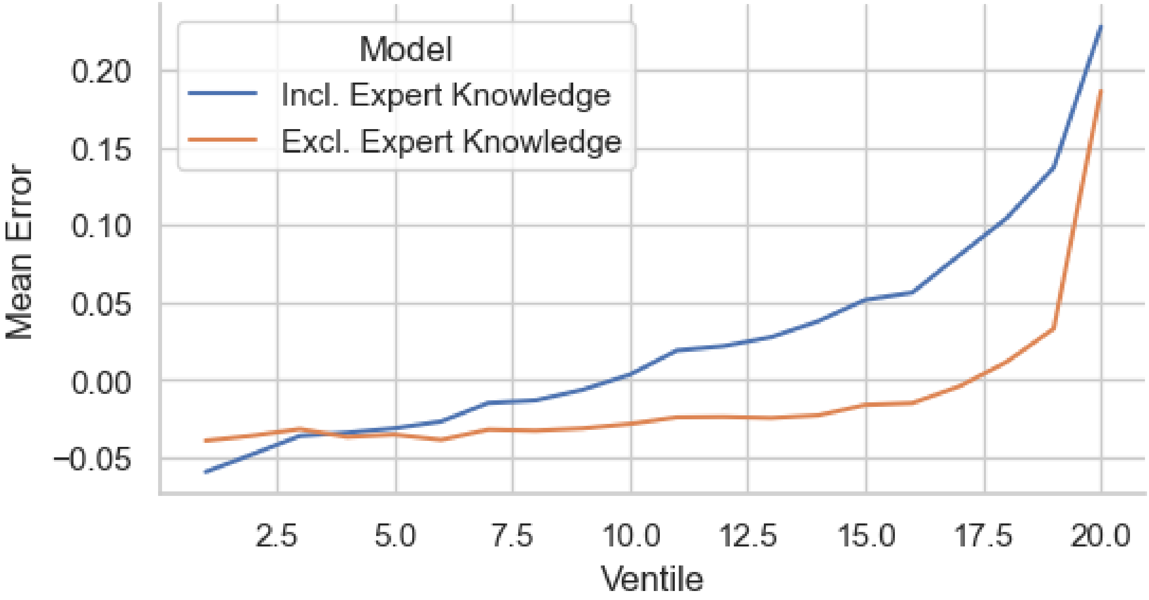

Figure 11: MAE per ventile (small dataset scenario). Split into ventiles was built on predicted claim count values by the XGBoost model trained in respective pipeline.

A Comparison of Distributions for Full Dataset Scenario

In order to understand the impact of expert knowledge on synthetic data generation, we carried out a comparative analysis of variable distributions in three specific datasets in the full dataset scenario, including training, validation, and test set ($N = 678013$). These datasets were the original data, the synthetic data created by the GAN with expert input, and the synthetic data produced by the GAN without expert input. The figures and tables in this Appendix depict the distribution profiles of various variables across these datasets. To ensure visual comparability, we calibrated the size of the synthetic data samples generated by the GANs to correspond with the size of the real dataset for this specific examination. Additionally, any synthetic data points that deviated beyond the boundaries of the real dataset (such as instances where *DrivAge ¡ 18*) were modified to align within the range defined by the real dataset.

A.1 *ClaimNb*

Data	Real	Generated: GAN with expert input	Generated: GAN without expert input
$N(ClaimNb = 0)$	643953	639395	639708
$N(ClaimNb = 1)$	32178	36286	37090
$N(ClaimNb = 2)$	1784	2182	1215
$N(ClaimNb = 3)$	82	149	0
$N(ClaimNb = 4)$	16	1	0

Table 3: Comparison of distributions of *ClaimNb*

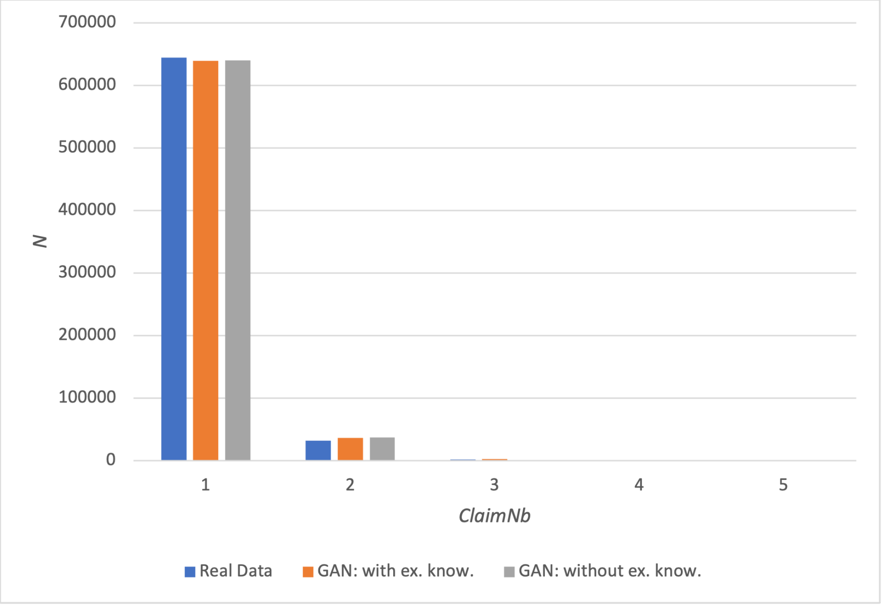

Figure 12: Comparison of distributions of the different datasets generated (real dataset vs. GAN generated vs. GAN generated with expert knowledge included) for *ClaimNb*, where N is the case count.

A.2 *Other variables*

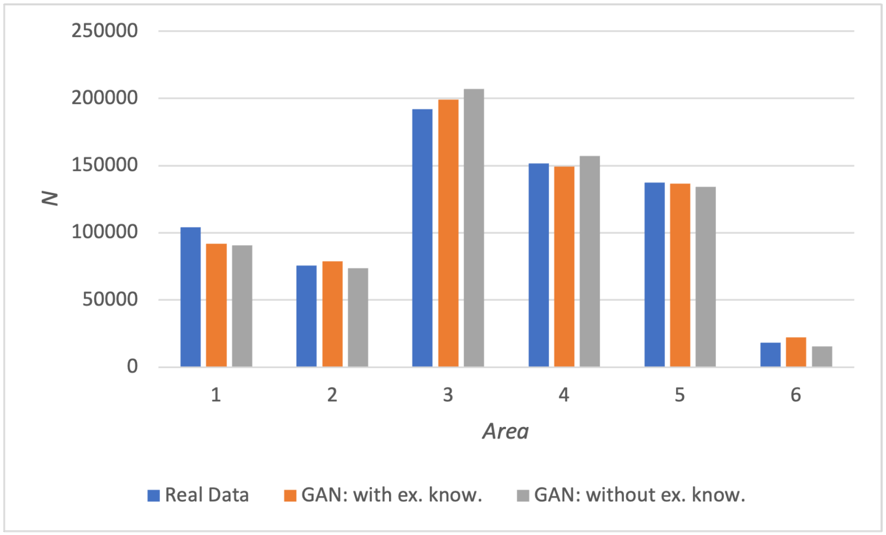

Figure 13: Comparison of distributions of the different datasets generated (real dataset vs. GAN generated vs. GAN generated with expert knowledge included) for *Area*, where N is the case count.

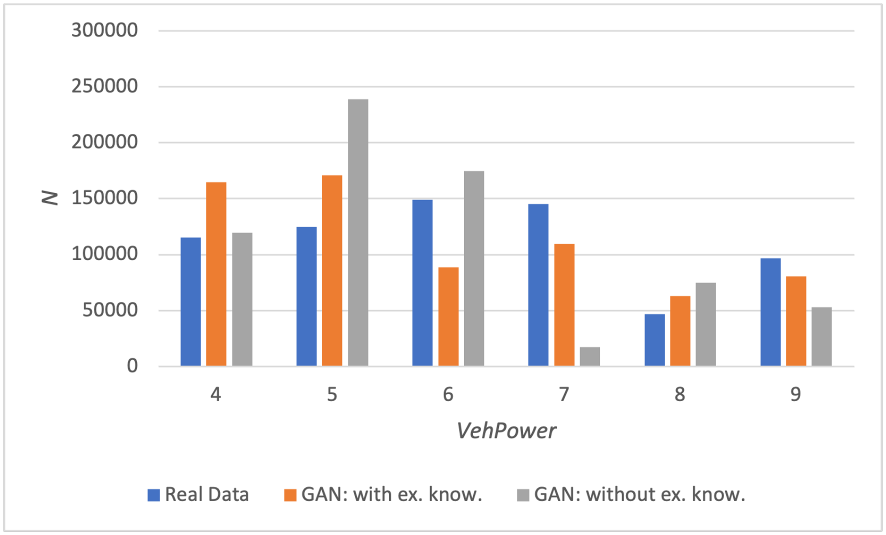

Figure 14: Comparison of distributions of the different datasets generated (real dataset vs. GAN generated vs. GAN generated with expert knowledge included) for *VehPower*, where N is the case count.

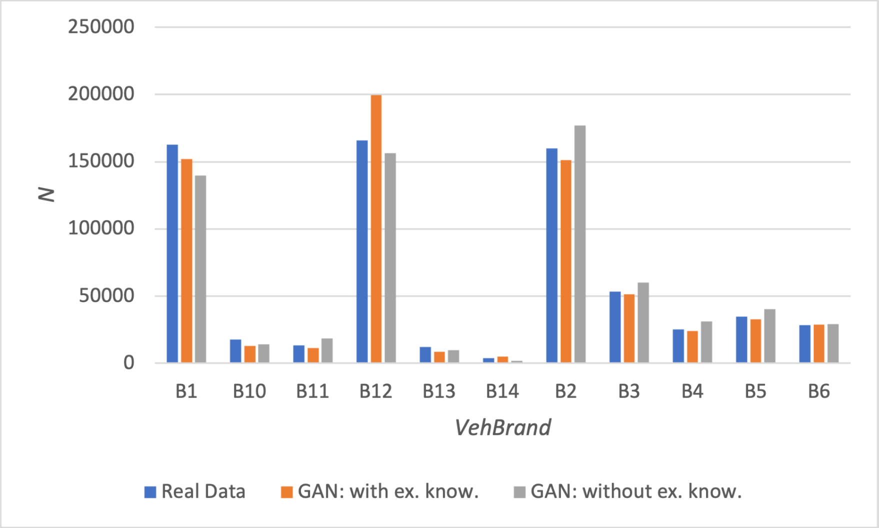

Figure 15: Comparison of distributions of the different datasets generated (real dataset vs. GAN generated vs. GAN generated with expert knowledge included) for *VehBrand*, where N is the case count.

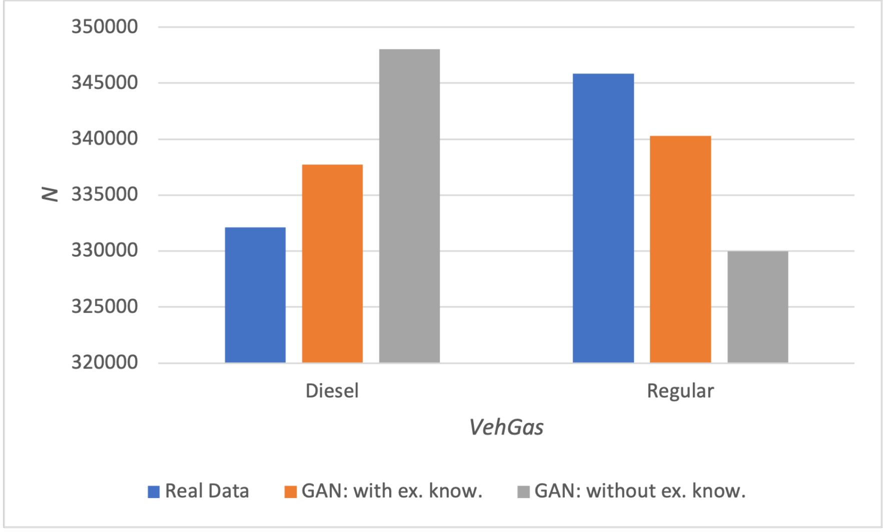

Figure 16: Comparison of distributions of the different datasets generated (real dataset vs. GAN generated vs. GAN generated with expert knowledge included) for *VehGas*, where N is the case count.

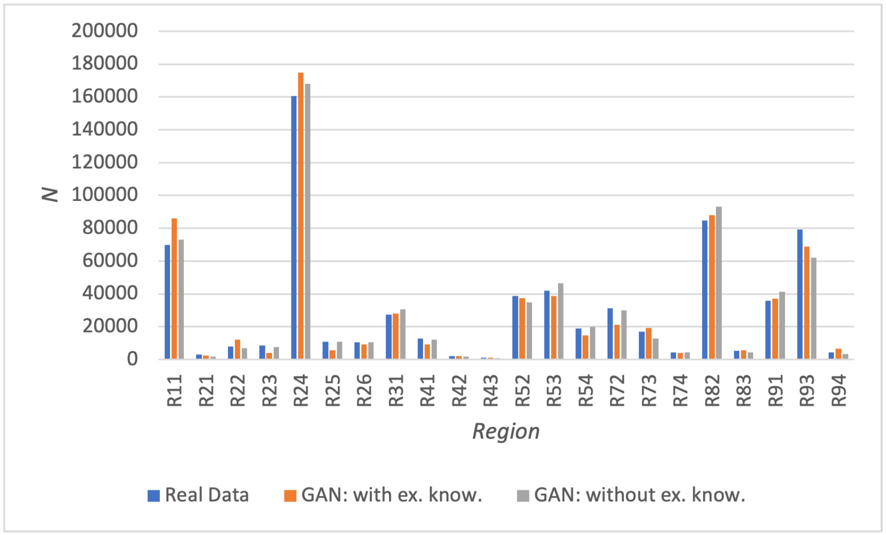

Figure 17: Comparison of distributions of the different datasets generated (real dataset vs. GAN generated vs. GAN generated with expert knowledge included) for *Region*, where N is the case count.

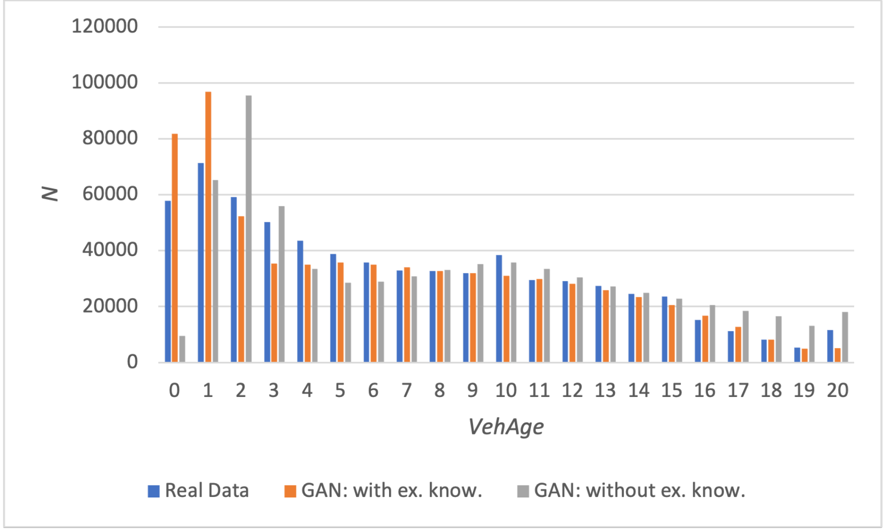

Figure 18: Comparison of distributions of the different datasets generated (real dataset vs. GAN generated vs. GAN generated with expert knowledge included) for *VehAge*, where N is the case count.

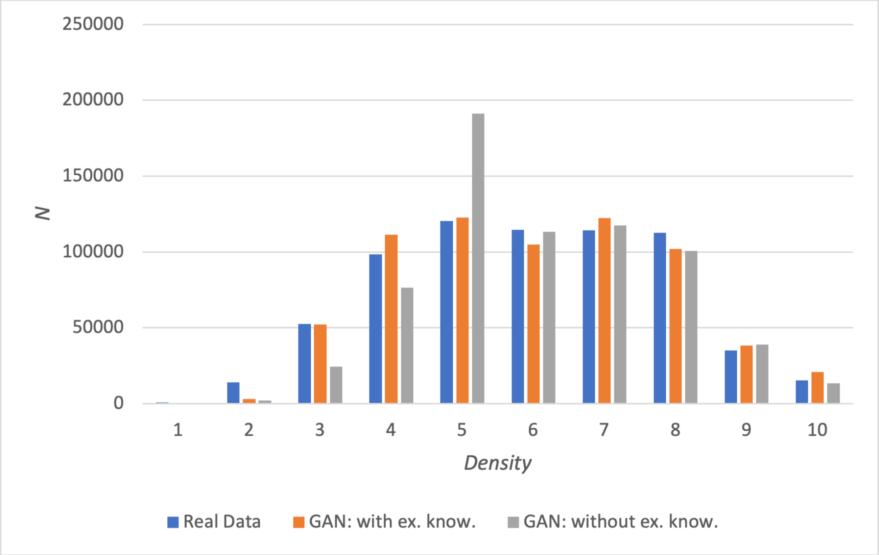

Figure 19: Comparison of distributions of the different datasets generated (real dataset vs. GAN generated vs. GAN generated with expert knowledge included) for *Density*, where N is the case count.

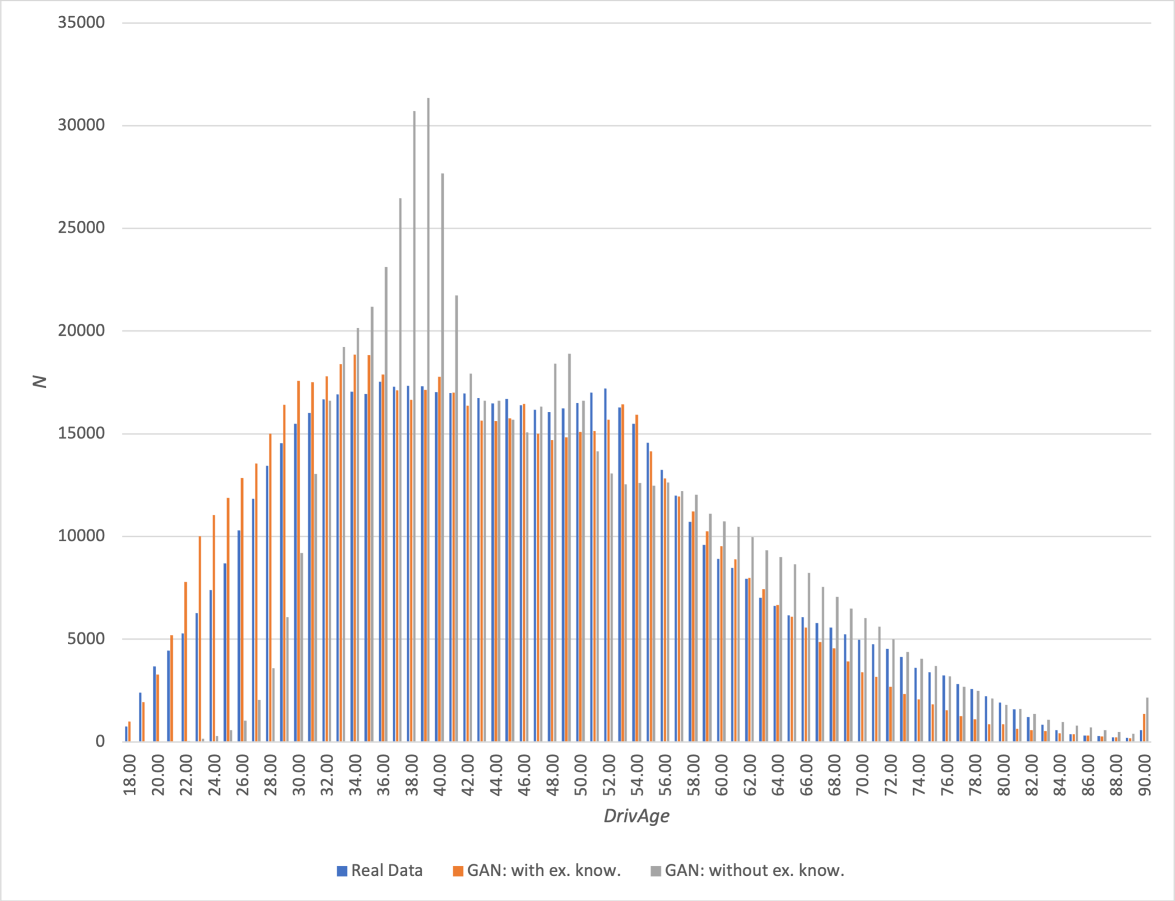

Figure 20: Comparison of distributions of the different datasets generated (real dataset vs. GAN generated vs. GAN generated with expert knowledge included) for *DrivAge*, where N is the case count.

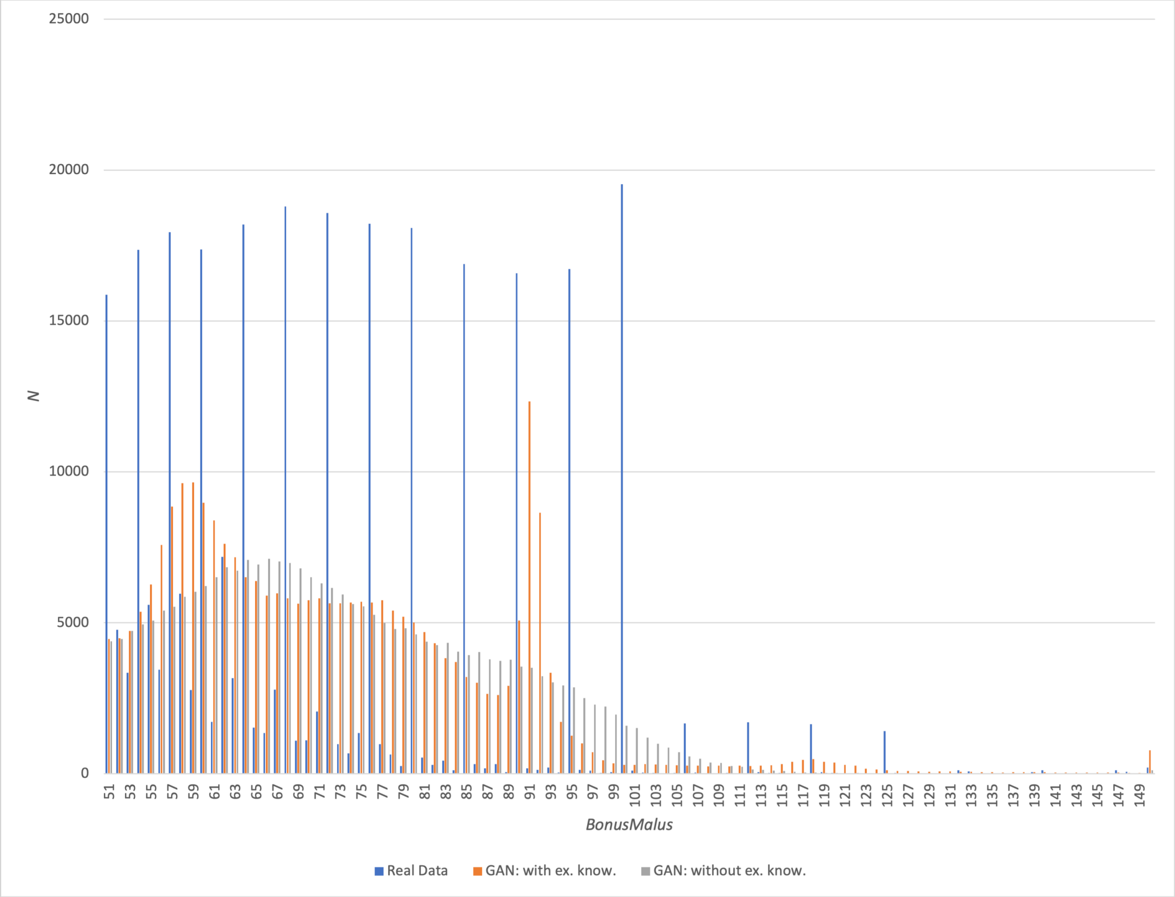

Figure 21: Comparison of distributions of the different datasets generated (real dataset vs. GAN generated vs. GAN generated with expert knowledge included) for *BonusMalus*, where N is the case count, excluding $BonusMalus = 50$ for visualisation purposes.

B Comparison of Relationships between independent variables and *ClaimNb* for Full Dataset Scenario

In order to understand the impact of expert knowledge on synthetic data generation, we carried out a comparative analysis of the relationships between the independent variables and *ClaimNb* across three specific datasets in the full dataset scenario, which includes training, validation, and test set ($N = 678013$). These datasets were the original data, the synthetic data created by the GAN with expert input, and the synthetic data produced by the GAN without expert input. The figures in this appendix depict the relationships between *ClaimNb* and various independent variables in these datasets. To ensure visual comparability, we calibrated the size of the synthetic data samples generated by the GANs to correspond with the size of the real dataset for this specific examination. Furthermore, any synthetic data points that deviated beyond the boundaries of the real dataset (such as instances where *DrivAge ¡ 18*) were modified to align within the range defined by the real dataset.

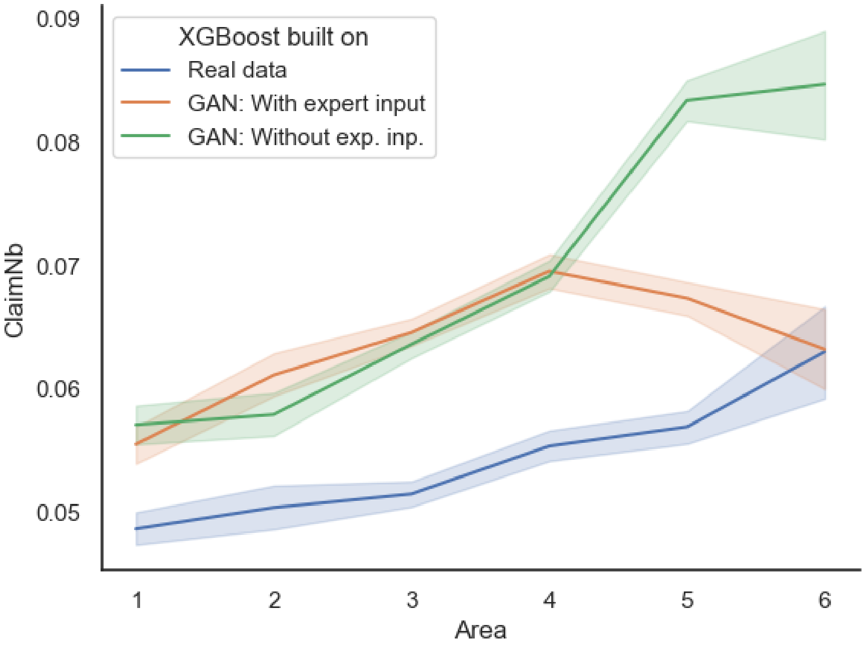

Figure 22: Relationship between *Area* and *ClaimNb* for the three main pipelines in the large dataset scenario.

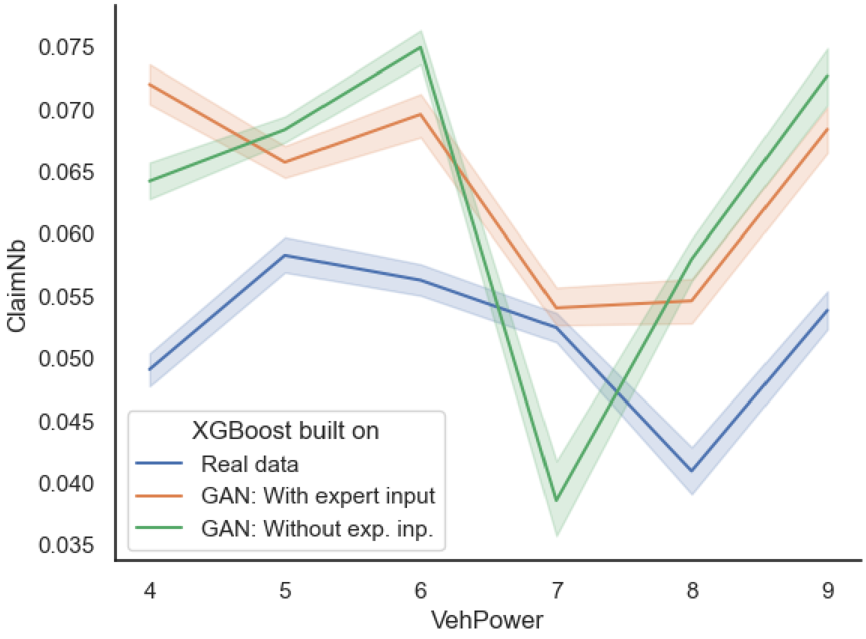

Figure 23: Relationship between *VehPower* and *ClaimNb* for the three main pipelines in the large dataset scenario.

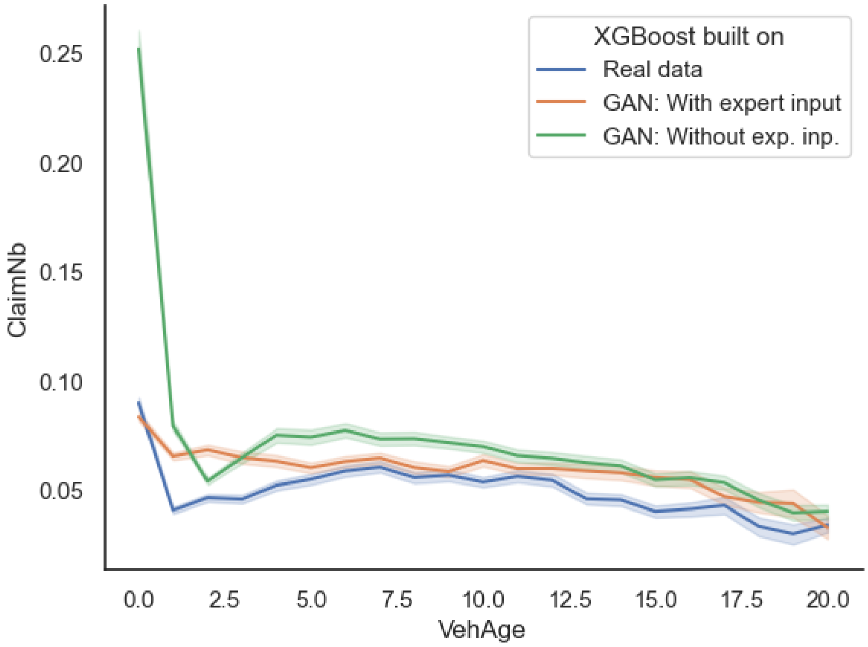

Figure 24: Relationship between *VehAge* and *ClaimNb* for the three main pipelines in the large dataset scenario.

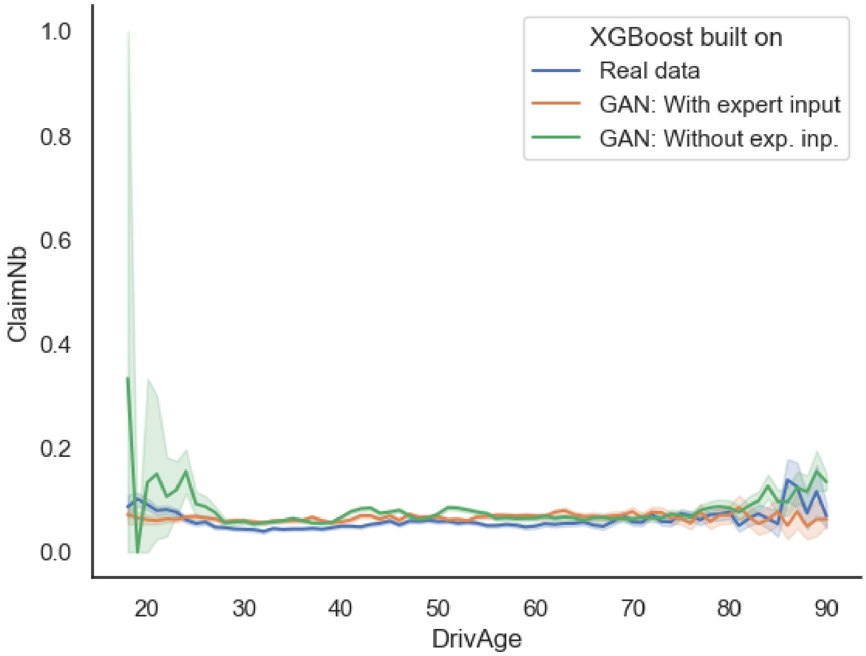

Figure 25: Relationship between *DrivAge* and *ClaimNb* for the three main pipelines in the large dataset scenario.

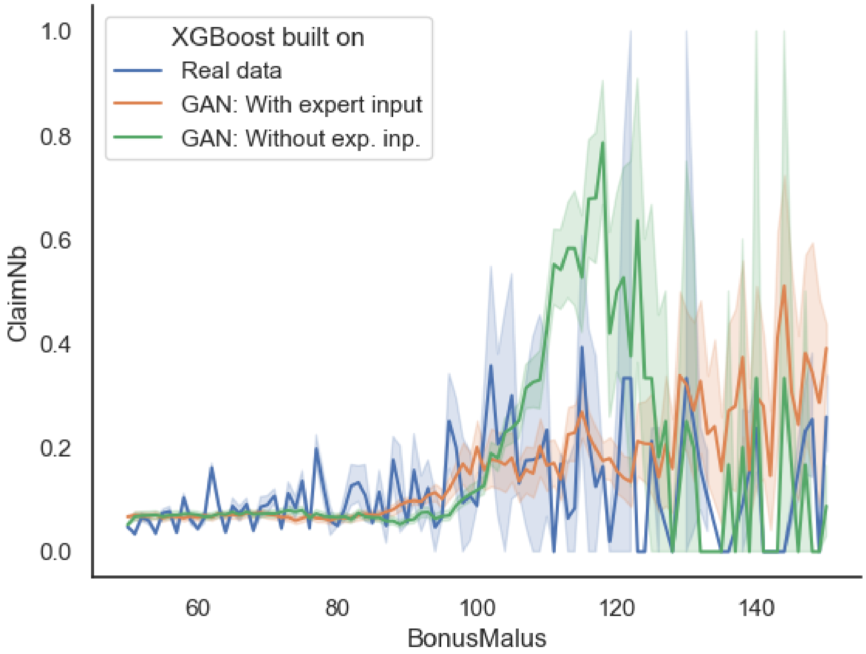

Figure 26: Relationship between *BonusMalus* and *ClaimNb* for the three main pipelines in the large dataset scenario.

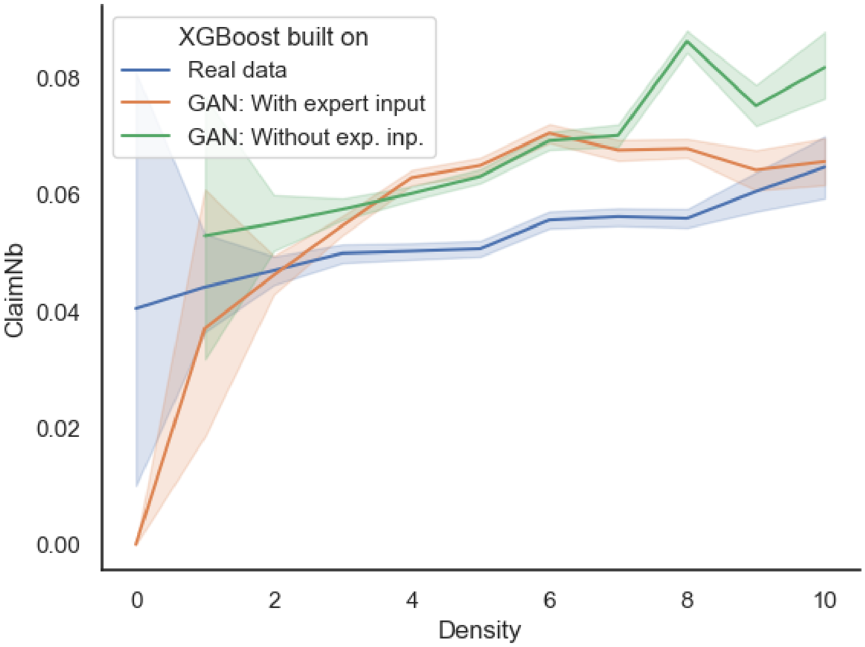

Figure 27: Relationship between *Density* and *ClaimNb* for the three main pipelines in the large dataset scenario.

C Summary of Expert Knowledge Provided

In multiple sessions, the actuarial expert provided her knowledge on MTPL vehicle insurance to us. The summary of the sessions is shared in this appendix. The summary is ordered into the different steps of inclusion of expert knowledge defined in the *Incorporating Expert Knowledge* section in the *Method* section: Idea Generation, Representation Selection, Representation Adjustment, Additional Rules.

C.1 Scope Definition

The actuary suggested that she can provide expert knowledge on the following variables: *VehAge, Density, BonusMalus, VehPower, DrivAge.*

C.2 *VehAge*

- Idea generation: The actuary proposed that the relationship between *VehAge* and *ClaimNb* can be modeled using a polynomial Generalized Linear Model (GLM). The degree of the polynomial (that is, the highest power of *VehAge* in the model) should not be greater than 4. The four different models were trained on the training set and shown to the expert.
- Representation selection: The actuary chose that the model best representing the relationship between *VehAge* and *ClaimNb* is $ClaimNb = \beta_0 + \beta_1 \times VehAge + \beta_2 \times VehAge^2 + \beta_3 \times VehAge^3$.
- Representation adjustment: The actuary decided that for vehicles below 5 years of age, the expected *ClaimNb* should be set at $\widehat{ClaimNb} = 0.05$.
- Additional rules: No additional rules have been suggested (see Figure 28 for the final representation).

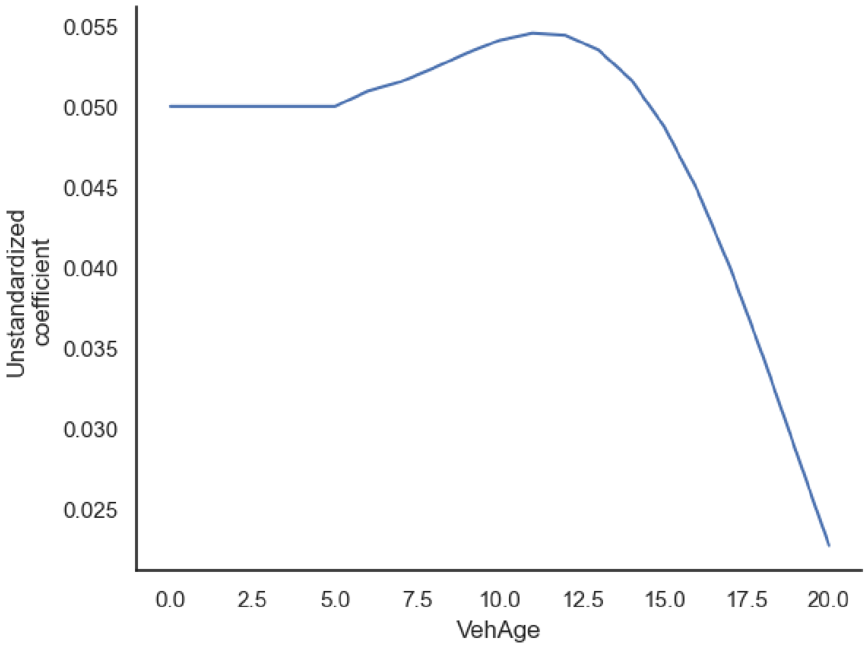

Figure 28: Chosen representation for the relationship between *VehAge* and claim count. The unstandardized coefficient was mapped to the respective users, standardized, and then used for GAN training.

C.3 *Density*

- Idea generation and representation selection: The actuary proposed that the relationship between *Density* and *ClaimNb* is linear. Therefore, it can be modeled using a monomial Generalized Linear Model (GLM). The model that best represents the relationship is $ClaimNb = \beta_0 + \beta_1 \times Density$.

- Representation adjustment: The were no adjustments to the model after it was built

- Additional rules: No additional rules have been suggested (see Figure 29 for the final representation).

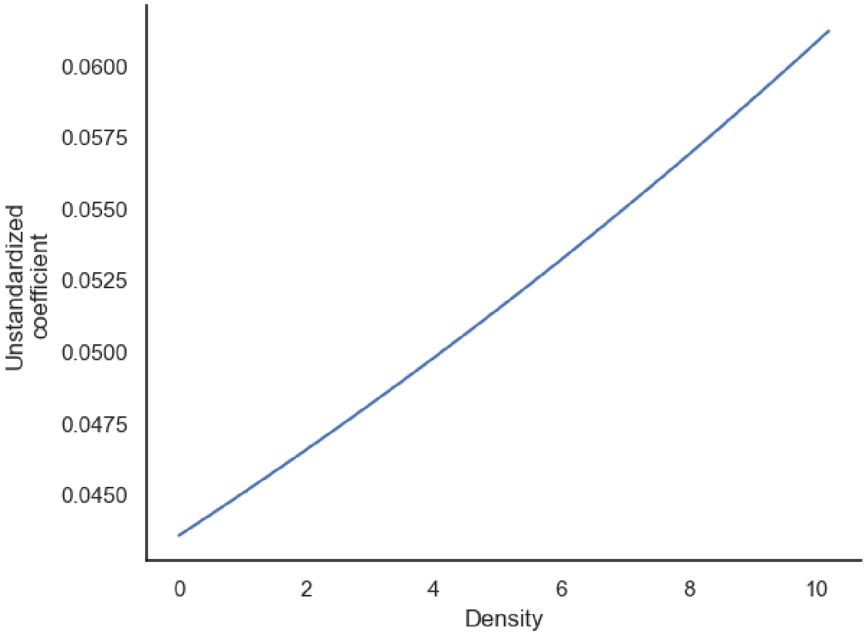

Figure 29: Chosen representation for the relationship between *Density* and claim count. The unstandardized coefficient was mapped to the respective users, standardized, and then used for GAN training.

C.4 *BonusMalus*

- Idea generation and representation selection: The actuary proposed that the relationship between *Density* and *ClaimNb* is linear. Therefore, it can be modeled using a monomial Generalized Linear Model (GLM). The model that best represents the relationship is $ClaimNb = \beta_0 + \beta_1 \times BonusMalus + \beta_2 \times BonusMalus^2$.

- Representation adjustment: There were no adjustments to the model after it was built (see Figure 30 for the final representation of *BonusMalus*).

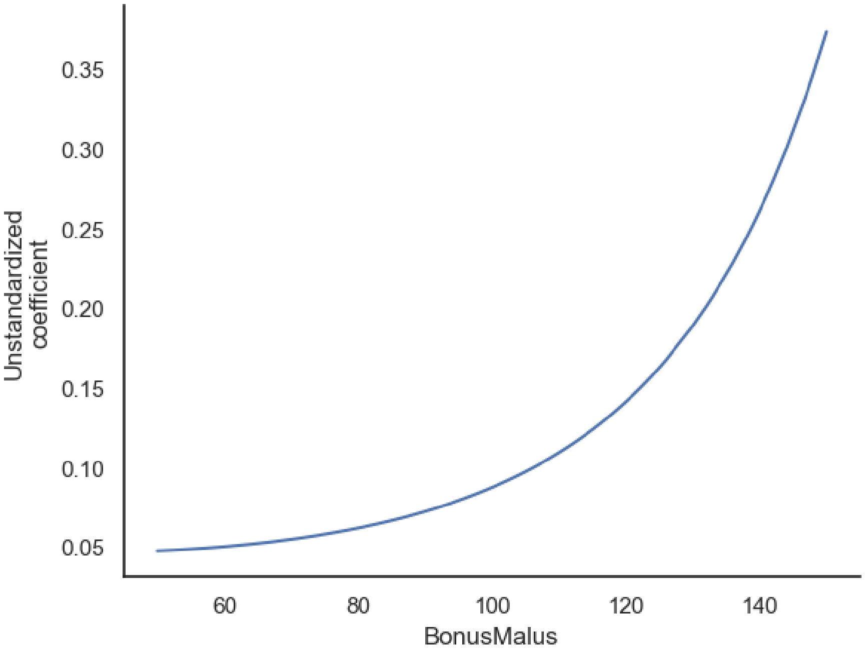

Figure 30: Chosen representation for the relationship between *BonusMalus* and claim count. The unstandardized coefficient was mapped to the respective users, standardized, and then used for GAN training.

- Additional rules: The expert added a variable which indicates if a customer had a bonus malus level below/above 100. This reflects in her eyes the fact that customers above this threshold behave much more risky than customers below this bonus malus level. The split was introduced into the model as grouping the customers into customers with a bonus malus above/below 100 and taking the average claim number. It resulted in a new variable with the coefficients X for customers with a lower bonus malus than 100 and Y for customers with a higher bonus malus than 100 (see Figure 31 for the final representation of the additional *BonusMalus* rule).

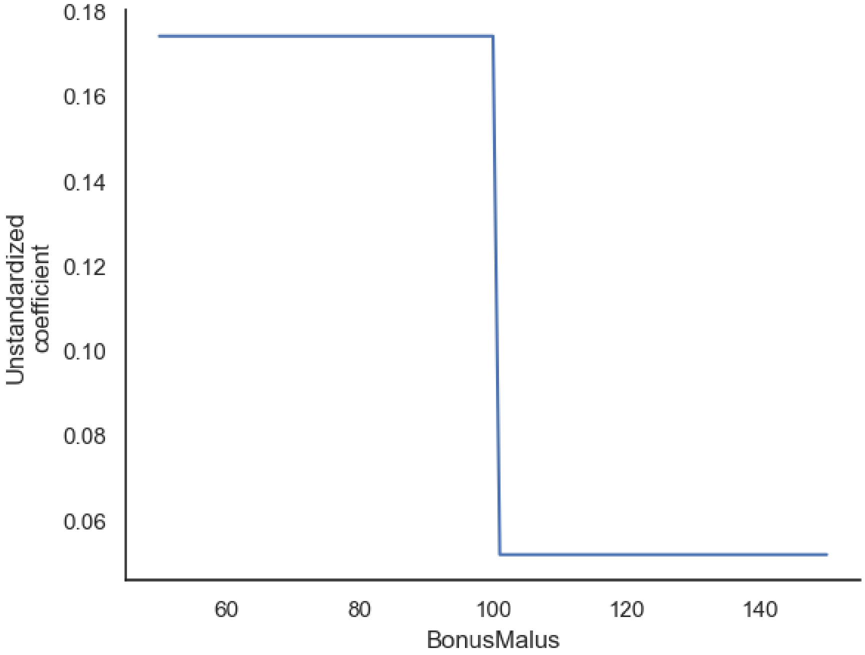

Figure 31: Chosen representation for the relationship between *BonusMalus* and claim count. The unstandardized coefficient was mapped to the respective users, standardized, and then used for GAN training.

C.5 *VehPower*

- Idea generation: The actuary stated that to her knowledge, there would be no relationship between *VehPower* and *ClaimNb*. Therefore, there is no representation in the form of any model and no additional rules are necessary.

- Representation selection: No representation was chosen.

- Representation adjustment: No adjustment was made.

- Additional rules: No additional rules have been suggested.

C.6 *DrivAge*

- Idea generation: The actuary proposed that the relationship between *DrivAge* and *ClaimNb* can be modeled using a polynomial Generalized Linear Model (GLM). The degree of the polynomial (that is, the highest power of *DrivAge* in the model) should be between 3 and 6. The four different models were trained on the training set and shown to the expert.

- Representation selection: The actuary chose that the model best representing the relationship between *VehAge* and *ClaimNb* is $ClaimNb = \beta_0+\beta_1\times DrivAge+\beta_2\times VehAge^2+\beta_3\times DrivAge^3+\beta_4 \times DrivAge^4 + \beta_4 \times DrivAge^4 + \beta_5 \times DrivAge^5$.

- Representation adjustment: No adjustment has been made by the adjustment

- Additional rules: No additional rules have been suggested (see Figure 32 for the final representation of *DrivAge* as expert knowledge).

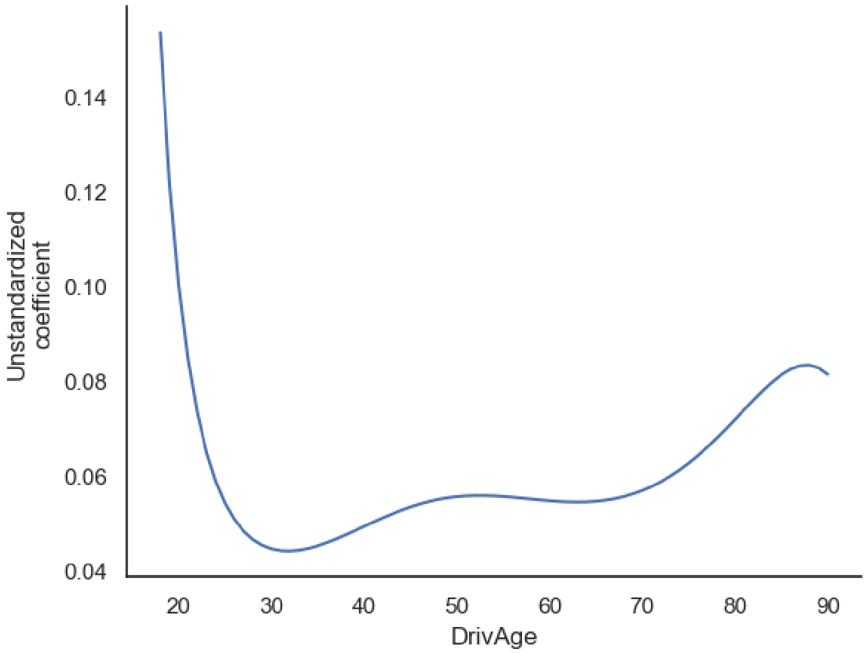

Figure 32: Chosen representation for the relationship between *DrivAge* and claim count. The unstandardized coefficient was mapped to the respective users, standardized, and then used for GAN training.

D Gridsearch of GAN Hyperparameters

To find the best performing model, a hyperparameter grid search was carried out. Due to infrastructural limits (the use of high-capacity Snellius graphical processing units was limited to 50000 system billing units), our grid search was limited to only the combinations of parameters that produced the models which best performed in previous research Cote et al. (2020). For the gridsearch hyperparameters searched, see the respective ganruns.csv file in our GitHub repository (URL: `https://github.com/AfairiJJ/thesis/blob/main/config/ganruns.csv`).

The optimized hyperparameters were: minibatch size for training (batch size), generator train rounds per epoch (Gen. epochs), penalty factor for the loss function (loss penalty), generator batch norm decay (Gen. bn. decay), critic batch norm decay (Cr. bn. decay), if the sigmoid function was used in the laster layer of the critic (Sigmoid). See Table 4 for the best GAN hyperparameters found.

Further parameters that were adopted from Cote et al. (2020) without gridsearch were: Number of epochs (15,000), critic train rounds per epoch (2 per epoch), size of hidden layers in generator (3 × 100), generator learning rate (0.001), critic learning rate (0.001), generator L2 regularization factor (0), generator learning rate (1), critic learning rate (1),

Hyperparameter	GAN for Hypotheses 1.1, 1.2	GAN for Hypothesis 2.1, 2.2
Batch size	100	500
Gen. epochs	1	2
Loss penalty	10	5
Critic bn. decay	0.010	0.000
Gen. bn. decay	0.900	0.500
Critic leak par.	0.200	0.100
Noise size	75	100
Sigmoid layer	True	True
Critic L2 reg.	0.000	0.000
Critic hidden layers	100	100, 100

Table 4: Best hyperparameters found for the GANs for the different hypotheses

E Gridsearch of XGBoost Hyperparameters

A small gridsearch was performed to find the best performing XGBoost model. Since gridsearches on the hyperparameters of the XGBoost model for this dataset have already been conducted, we compared the best performing models evaluated through gridsearches conducted by König and Loser (2020) with the best performing model found by a gridsearch by Martínez de Lizarduy Kostornichenko (2021) to find the overall best performing model. The model of König and Loser (2020) performed better in the large dataset, while the model of Martínez de Lizarduy Kostornichenko (2021) performed better in the small dataset with $N = 5,000$. The hyperparameters shared between the models were the column sub-sample size for each estimator (80%), the row sub-sample size for each estimator (90%), the minimal weight per child for each estimator (10), and the estimator building method (*hist*). The different hyperparameters can be seen in Table 5

Hyperparameter	N of estimators	Max. depth of each estimator	Learnin
XGBoost by König and Loser (2020)	1200	7	0.025
XGBoost by Martínez de Lizarduy Kostornichenko (2021)	500	5	0.02

Table 5: Differing hyperparameters for the two versions of the XGBoost model