A Novel and Interpretable Deep Learning Architecture for Insurance Pricing: Replication Report

Julian Schady, Cheryl E. K. Botwe, Prince Kudolo Diaba
Department of Statistical Sciences
University of Toronto

December 2025

Abstract

This project investigates how transformer-based architectures can enhance classical actuarial models for non-life insurance pricing. By examining Brauer (2024), we replicate and extend a modeling framework on the French motor third-party liability frequency dataset `FreMTPL2freq`. The analysis spans a hierarchy of models: a homogeneous Poisson mean model, three increasingly rich Poisson GLMs (GLM1–GLM3), a feed-forward neural network with categorical embeddings (FNN_EMB), the Combined Actuarial Neural Network (CANN), the Feature Tokenizer Transformer (FTT_def), and the Combined Actuarial Feature Tokenizer Transformer (CAFTT_def). All models are evaluated on a common train–validation–test split using Poisson deviance and average predicted claim frequencies, with additional rebalanced and ensemble variants for the neural and transformer models.

The results confirm that well-specified GLMs, in particular GLM3 with transformed covariates and interaction terms, already provide a strong pricing benchmark with good deviance and excellent portfolio-level calibration. Pure neural and transformer models (FNN_EMB, FTT_def) achieve further reductions in deviance but tend to misestimate the overall claim frequency, highlighting a trade-off between fit and calibration. Hybrid constructions that add a neural or transformer residual on top of GLM3 (CANN and CAFTT_def) dominate the purely parametric and purely machine-learning models once a simple rebalancing step is applied. In particular, the rebalanced CAFTT_def model attains the lowest test deviance while matching the empirical mean frequency. These findings suggest that, for non-life pricing, transformers are most effective when used as flexible residual layers on top of a GLM backbone, providing incremental predictive gains without abandoning actuarial structure and interpretability.

Contents

1 Introduction

The advent of deep learning has improved prediction performance across a wide range of different areas. This improved performance has prompted many to apply these models to various field. This is true with respect to the field of non-life insurance pricing. In recent years, many papers have been published that detail the application of new deep learning models to non life insurance prediction. Despite the large amount of current research into deep learning models for non-life insurance, General Linear Models (GLMs) remain the standard model used in industry, due to their ease of interpretation, regulation, good performance and existing data pipelines [1]. GLMS were first introduced by Nelder and Wedderbur (1972) [2]. They extend Ordinary linear models by allowing the response variable to come from the exponential family of distributions. The Poisson distribution is often used in non life insurance when predicting frequency of an event. The GLM was further extended into the General Additive Model (GAM) by Hastie and Tibshirani (1986)[3] which added the use of splines.

There were many attempts to combine the GLMs with Neural Networks in order to improve performance while retaining the advantages of GLMs. Tran et all (2020) introduced the DeepGLM model which uses a Neural Network to transform the inputs before feeding them into GLM model [4]. Specifically in actuarial science research, Schelldorfer and Wuthrich introduce the Combined Actuarial Neural Network (CANN), which as the names suggests combines the output of a neural network to enhance the predictions of a pretrained GLM [5]. Then further work in this direction was proposed by Richman and Wuthrich called the LocalGLMnet [6]. This model preserves the linear structure of the GLM while allowing the regression coefficients to vary as functions of the covariates. These feature-dependent coefficients are produced by a neural network. This results in a model that has the performance of a neural network but you could argue it can be interpreted the same as a GLM. The ability to interpret pretty much the same makes the LocalGLMnet particularly attractive in actuarial applications, where model interpretability remains very important.

The predictive performance of neural networks in actuarial applications is strongly influenced by the choice of feature representations. There has been a lot of research regarding embeddings for categorical variables when you have high-cardinality categorical features. There effectiveness has been demonstrated in several paper such as Kuo and Richman (2021) [7]. In addition to embeddings, representation learning approaches based on autoencoders have also been explored as a means of learning compact and informative feature representations for insurance data as we see in Blier-Wong (2021).[8]. A comprehensive benchmark study comparing a wide range of actuarial models for claim frequency and severity prediction is provided by Holvoet et al. [9].

More recently, lots of progress has been made in the development of neural networks specifically designed for supervised learning on tabular data. Tabular data is any sort of data you could put into an excel sheet, it could be categorical or numerical. A key contribution in this area is TabNet, introduced by Arik and Pfister [10], which employs a sequential attention mechanism to dynamically select and re-weight features during training. An application of TabNet in an insurance pricing context is presented by McDonnell et al. [11], where both predictive accuracy and interpretability are analyzed. Another influential model is the TabTransformer proposed by Huang et al. [12], which adapts the transformer architecture to tabular data by using self-

attention to generate contextual embeddings for categorical features. The embedding properties of the TabTransformer have been examined in an actuarial setting by Kuo and Richman [7], who also provide a detailed discussion of the self-attention mechanism underlying transformer-based models.

The main modeling approach considered in this work is the Feature Tokenizer Transformer (FTT), originally introduced by Gorishniy et al (2021). [13]. While sharing conceptual similarities with the TabTransformer, the FTT differs in itstreatment of numerical features and in its prediction mechanism. In contrast to the TabTransformer, the FTT embeds numerical features directly and uses a dedicated classification token (CLS token) to aggregate information for prediction. Subsequent work by Gorishniy et al. [13] demonstrated that improving the numerical feature embeddings can lead to substantial gains in predictive performance. We will further explain this model later in the paper in the models section. When seeing the improved performance of the FTT from how it creates its embeddings, the question is raised if the FTT model could be applied to actuarial models the same way as that Neural networks have been combined with GLMs. Brauer (2024) enhances the LocalGLMnet and CANN with this Feature Tokenizer Transformer. In the next section we will summarize Brauer (2024) and from there we will explain which parts of the paper we recreated. We will then discuss if we reach the same results has Brauer (2024).

2 Summary of Brauer (2024)

2.1 Summary Introduction

Brauer (2024) compares several different models to the problem of predicting auto claim frequency. The paper introduces two new models the Combined Actuarial Feature Tokenizer Transformer (CAFTT) and the LocalGLM-feature-tokenizer-transformer (LocalGLMftt). It compares them to a model that predicts the mean claim Number, three different GLMs as the industry baseline, A Feed-Forward Neural Network that uses one hot encodings for features and one that uses embeddings, the regular Combined Actuarial Neural Network, The Local GLM Neural Network model and finally the Feature Tokenizer Transformer on its own.

2.2 Models Not Recreated

The models will be explained in detail in the model description section. However, we didn't recreate the Feed Forward Neural Network (FNN) that used one hot encoding, the local GLM Neural Network or the Local GLM Feature Tokenizer Transformer models, so we want to explain these three in this section. The FNN that uses one hot encoding is the exact same model as the FNN that uses feature embeddings in every way other then how the features are transformed. One hot encoding creates a feature for each possible category of a categorical variable. If for a data point that entry is that category, then the one hot encoding vector will be 1 at that spot, and zero every where else. this is a common way to represent categorical variables in deep learning architectures. Categorical rating factors are encoded using one-hot vectors and concatenated with standardized numerical covariates to form the network input. The model estimates claim frequency via

$$\mu_i = e_i \exp\left(f_\theta(x_i)\right),$$

where e_i denotes exposure and $f_\theta(\cdot)$ is a multi-layer perceptron with fully connected hidden layers and nonlinear activations. The local GLM Net is a way of enhancing a GLM, by using coefficients that are determined by a Neural Network, so they are dependent on the covariates. The intuition behind this is that a neural network can enhance the estimation of these coefficients, and since the output is still a GLM model it is just as interpretable. The LocalGLMnet prediction can be defined as

$$\hat{y} = g^{-1}\left(\beta_0 + \beta(x)^\top x\right), \tag{1}$$

where $\beta(x)$ is obtained from a feed-forward neural network (FNN) of depth d with identical input and output dimensions,

$$\beta : \mathbb{R}^k \to \mathbb{R}^k, \tag{2}$$

and is given by

$$\beta(x) = \left(z^{(d)} \circ \cdots \circ z^{(1)}\right)(x). \tag{3}$$

Building off the Local-GLM-net, this paper introduces local-GLMFTT, where the neural network is replaced by the Feature Tokenizer Transformer. the advantages of this model is it keeps the interpretability of the GLM, but enhances the coefficient estimations by using the predictive advantages of a FTT over a regular Neural Network. To help understand the architecture better, figure 1 shows a diagram of the Local-GLM-FTT model. These three models all offer great insights that were useful in the original paper, but were not included in our recreation do to the size of our group. The rest of the models are explained in depth in Model description section.

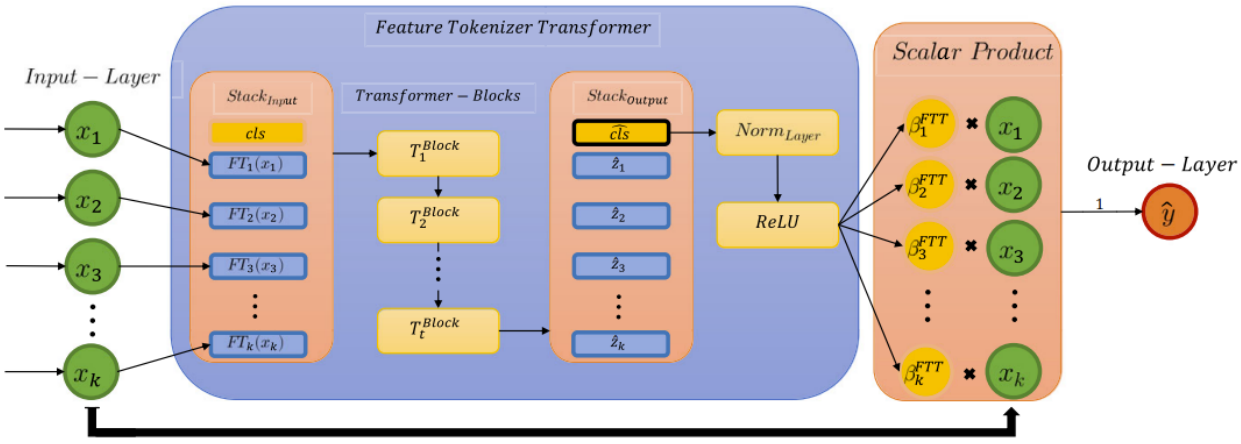

Figure 1: Local GLM Feature Tokenizer Transformer Architecture

2.3 Summary of Bauer (2024) Results

The performance of the proposed methods was evaluated on the French motor third-party liability (MTPL) claims frequency dataset, a standard benchmark in actuarial literature. Two strategies to improve predictive performance for all models were also compared. Rebalancing of the transformer models helped them with the problem of over and user prediction by adding a calibration. The ensemble models were 5 rebalanced models trained on independent parts of the training data, and averaged together. The results of the paper can be seen in the following tables. Table 1 shows the training details of the regular models. Table 2 shows the training loss test loss

and average prediction of the singular models. Table 3 shows the same but for the rebalanced models. Finally Table 4 shows the same but for the ensemble models.

Table 1: Model Fitting Properties: Average and standard deviation over 15 fits with different seeds.

Model	**Epochs**	**Run-time (mm:ss.ms)**	**# parameters**
(a) Mean model	-	00:00.05 (±00:00.00)	1
(b) GLM_1	-	00:02.22 (±00:00.47)	49
(c) GLM_2	-	00:02.75 (±00:00.97)	48
(d) GLM_3	-	00:01.90 (±00:00.41)	50
(e) FFN_{OHE}	42 (±15)	00:37.81 (±00:08.62)	1306
(f) FNN_{EMB}	73 (±22)	00:58.73 (±00:13.91)	792
(g) CANN	90 (±54)	01:08.56 (±00:33.16)	792
(h) LocalGLMnet	25 (±08)	00:29.72 (±00:05.10)	1737
(i) FTT_{def}	79 (±17)	26:09.86 (±05:02.32)	27,133
(j) $CAFTT_{def}$	57 (±14)	19:30.16 (±03:40.45)	27,133
(k) LocalGLMftt$_{\text{def}}$	53 (±16)	19:47.01 (±05:10.65)	27,430

Table 2: Average Single Model Results: Average and standard deviation over 15 fits with different seeds.

Model	**Train-loss** (10^{-2})	**Test-loss** (10^{-2})	**Avg**($\hat{y}$) **in %**
(a) Mean model	25.213	25.445	7.363
(b) GLM_1	24.101	24.146	7.390
(c) GLM_2	24.091	24.113	7.398
(d) GLM_3	24.084	24.102	7.405
(e) FFN_{OHE}	23.754 (±0.033)	23.865 (±0.016)	7.431 (±0.121)
(f) FNN_{EMB}	23.768 (±0.016)	23.827 (±0.015)	7.424 (±0.109)
(g) CANN	23.742 (±0.061)	23.810 (±0.033)	7.444 (±0.110)
(h) LocalGLMnet	23.710 (±0.033)	23.921 (±0.022)	7.427 (±0.091)
(i) FTT_{def}	23.780 (±0.090)	23.939 (±0.053)	6.129 (±0.146)
(j) $CAFTT_{def}$	23.715 (±0.047)	23.807 (±0.017)	6.623 (±0.047)
(k) LocalGLMftt$_{\text{def}}$	23.721 (±0.059)	23.880 (±0.016)	6.832 (±0.099)

The experimental results, summarized in Table 2, indicate that while all neural network-based approaches improved upon the baseline GLMs in terms of Poisson deviance loss, the transformer-based architectures demonstrated superior predictive power. Specifically the CAFTT model performed the best in terms of Poisson Deviance Loss on the test set. However the transofrmer models seem to be underpredictigng the average number of claims as we can see in Table 2. This was adressed by the use of rebalancing and ensemble models. As illustrated in the ensemble comparison in Table 4, the Combined Actuarial Feature Tokenizer Transformer (CAFTT) and LocalGLM-feature-tokenizer-transformer (LocalGLMftt) consistently outperformed their FNN-based counterparts—the CANN and LocalGLMnet, respectively.Specifically, the CAFTT ensemble achieved a test loss of 23.726×10^{-2} compared to the CANN's 23.769×10^{-2}. Notably, this performance

Table 3: Rebalanced Model Results: Average and standard deviation of rebalanced model results over 15 fits.

Model	**Train-loss** (10^{-2})	**Test-loss** (10^{-2})	**Avg($\hat{y}$) in %**
(e) Rebalanced: FFN_{OHE}	23.752 (±0.033)	23.864 (±0.015)	7.403 (±0.007)
(f) Rebalanced: FNN_{EMB}	23.767 (±0.016)	23.826 (±0.015)	7.409 (±0.005)
(g) Rebalanced: CANN	23.741 (±0.061)	23.809 (±0.033)	7.405 (±0.006)
(h) Rebalanced: LocalGLMnet	23.709 (±0.034)	23.920 (±0.022)	7.407 (±0.005)
(i) Rebalanced: FTT_{def}	23.652 (±0.064)	23.815 (±0.036)	7.381 (±0.011)
(j) Rebalanced: $CAFTT_{def}$	23.669 (±0.046)	23.766 (±0.017)	7.392 (±0.005)
(k) Rebalanced: LocalGLMftt$_{\text{def}}$	23.696 (±0.066)	23.859 (±0.017)	7.408 (±0.007)

Table 4: Average Ensemble Model Results: Average results over 3 ensembles consisting of 5 rebalanced models each.

Model	**Train-loss** (10^{-2})	**Test-loss** (10^{-2})	**Avg($\hat{y}$) in %**
(e) Ensemble: FFN_{OHE}	23.715 (±0.022)	23.826 (±0.010)	7.403 (±0.002)
(f) Ensemble: FNN_{EMB}	23.743 (±0.003)	23.801 (±0.011)	7.409 (±0.002)
(g) Ensemble: CANN	23.701 (±0.043)	23.769 (±0.030)	7.405 (±0.001)
(h) Ensemble: LocalGLMnet	23.664 (±0.013)	23.873 (±0.002)	7.407 (±0.002)
(i) Ensemble: FTT_{def}	23.592 (±0.003)	23.759 (±0.014)	7.381 (±0.003)
(j) Ensemble: $CAFTT_{def}$	23.630 (±0.018)	23.726 (±0.006)	7.392 (±0.001)
(k) Ensemble: LocalGLMftt$_{\text{def}}$	23.645 (±0.034)	23.811 (±0.010)	7.408 (±0.003)

by the CAFTT ensemble represents the lowest test loss among all tested architectures, thereby setting a new benchmark for this dataset.

However, this performance gain comes with a significant computational cost. The training time for the transformer-based models was substantially higher—ranging between 19 and 26 minutes—compared to mere seconds for GLMs and roughly one minute for standard FNNs [1]. Furthermore, the study highlighted a necessity for calibration; the transformer models initially produced average claim frequency predictions that deviated from the observed mean. A rebalancing step, defined as the ratio of the mean observed to predicted values on the training data, was required to align the predictions [1].

2.4 Bauer (2024) Summary of Conclusions

The research concludes that integrating transformer architectures into actuarial pricing models offers a tangible improvement in predictive accuracy while preserving the structural advantages of existing frameworks. By utilizing the Feature Tokenizer Transformer (FTT) [14], the proposed CAFTT and LocalGLMftt models can process categorical features directly, allowing insurers to maintain their established data engineering pipelines [1]. The choice between these models represents a strategic trade-off between raw performance and interpretability. The CAFTT is recommended for scenarios where minimizing deviance loss is paramount. Conversely, the LocalGLMftt offers a compelling middle ground; it outperforms the native LocalGLMnet while retaining the ability to visualize feature contributions, a critical requirement for regulatory compliance and model transparency [1]. Despite these advances, the full replacement of GLMs remains challeng-

ing. Brauer [1] notes that deep learning models currently lack the granular control required for expert judgment and often fail to guarantee the time-consistency properties that actuaries rely upon to avoid overfitting to short-term trends. Future work must therefore address these practical limitations, specifically focusing on the optimization of inference times for real-time applications and the development of constraints to prevent indirect discrimination. In our replication of we only compare some of the models that Brauer (2024) compares. Despite this we aim to replicate the same conclusions of this paper.

3 Model description

The overall objective is to model the claim count Y_i for policy i with exposure v_i through a Poisson frequency model of the form

$$Y_i \mid \boldsymbol{x}_i \sim \text{Poisson}(\mu_i), \qquad \mu_i = v_i \lambda_i,$$

where λ_i denotes the annualized claim frequency and $\boldsymbol{x}_i$ is the vector of covariates. All approaches described below can be written in terms of a log–link function

$$\eta_i = \log \lambda_i, \qquad \lambda_i = \exp(\eta_i),$$

with different specifications for the predictor η_i depending on the model class. The exposure v_i enters multiplicatively so that the predicted mean is $\widehat{\mu}_i = v_i \exp(\eta_i)$. The sequence of models is chosen to move gradually from very simple, fully parametric specifications towards more flexible neural and transformer-based architectures, while still keeping a strong actuarial flavour and interpretability whenever possible.

3.1 Mean model

In general terms, the mean model represents the simplest possible pricing approach: it assumes that all policies in the portfolio share the same risk level and therefore assigns the same expected claim frequency to every contract. This model does not use any rating factors and thus cannot differentiate between high- and low-risk policyholders; its purpose in the analysis is to serve as a naïve benchmark against which all subsequent models can be compared, highlighting the incremental value of including structured covariate information.

The starting point is a homogeneous Poisson model that ignores all covariates and assumes a constant frequency for all policies. In this case the predictor η_i reduces to a single intercept parameter α,

$$\eta_i = \alpha, \qquad \lambda_i = \exp(\alpha),$$

so that every policy receives the same expected frequency $\lambda = \exp(\alpha)$ and mean claim count $\mu_i = v_i \exp(\alpha)$. The maximum–likelihood estimate of α corresponds to the logarithm of the empirical average claim frequency on the training sample. This mean model serves as a naïve benchmark against which more structured models can be compared.

3.2 Generalized linear and additive models

From an actuarial perspective, generalized linear models are the standard workhorse for non-life pricing. They extend the mean model by allowing the expected claim frequency to depend linearly

on a set of transformed covariates, while retaining an interpretable link between model coefficients and rating factors. In this project, the GLM family is used both to construct increasingly rich parametric baselines (GLM1–GLM3) and to provide the actuarial backbone for hybrid neural and transformer models. The aim is to see how far a carefully specified GLM can go in explaining variation in claim frequencies, and then measure the incremental benefit of the more flexible methods.

The next step is to incorporate covariates through generalized linear models (GLMs). In a Poisson GLM with log–link the predictor is specified as

$$\eta_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij},$$

where x_{ij} denotes the j-th transformed covariate for policy i and β_j are regression coefficients. Categorical rating factors are represented by dummy variables and numerical covariates can either enter linearly or via engineered transformations such as logarithms or low-order polynomials. For example, if DrivAge_i is the driver age and VehAge_i the vehicle age, then GLM2 and GLM3 accommodate terms such as $\log(\text{DrivAge}_i)$ or VehAge_i^2, as well as interaction terms of the form $\beta_{jk} x_{ij} x_{ik}$ to capture joint effects between important risk factors. The three GLM specifications considered in the analysis are nested: GLM1 uses a simple set of main effects, GLM2 enriches the functional form for continuous covariates and GLM3 adds selected two–way interactions.

Conceptually, a generalized additive model (GAM) replaces the strictly linear predictor by a sum of smooth functions,

$$\eta_i = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}),$$

where each f_j is estimated nonparametrically (for example with splines). While the current implementation focuses on GLMs rather than fully nonparametric GAMs, the GLM2 and GLM3 specifications with polynomial and log–transformed features can be regarded as simple parametric approximations to an underlying additive structure.

3.3 Feed-forward neural network with embeddings

Feed-forward neural networks provide a flexible, largely model-free way to learn nonlinear relationships from data. Instead of specifying the functional form of each rating factor in advance, one lets the network discover appropriate transformations and interactions automatically through its hidden layers. In the context of insurance pricing, the goal of the FNN_EMB model is to capture complex dependence patterns between rating factors that may be missed by GLMs, while still respecting the Poisson structure of the claim counts and the exposure adjustment.

To move beyond hand-crafted functional forms, a feed-forward neural network with embeddings (FNN_EMB) is used to model a flexible nonlinear mapping from the covariates to the log–frequency. Let $\boldsymbol{z}_i$ denote the collection of continuous features and let $\boldsymbol{c}_i$ denote the categorical features. Each categorical variable is mapped to a low-dimensional embedding vector through a learned lookup table, and the continuous features are standardised and concatenated with these embeddings to form an input vector $\boldsymbol{u}_i$. The neural network then computes

$$\eta_i = f_\theta(\boldsymbol{u}_i),$$

where f_θ is a composition of affine transformations and nonlinear activation functions (hidden layers), parameterised by weights and biases θ. In the final layer, the network outputs a scalar

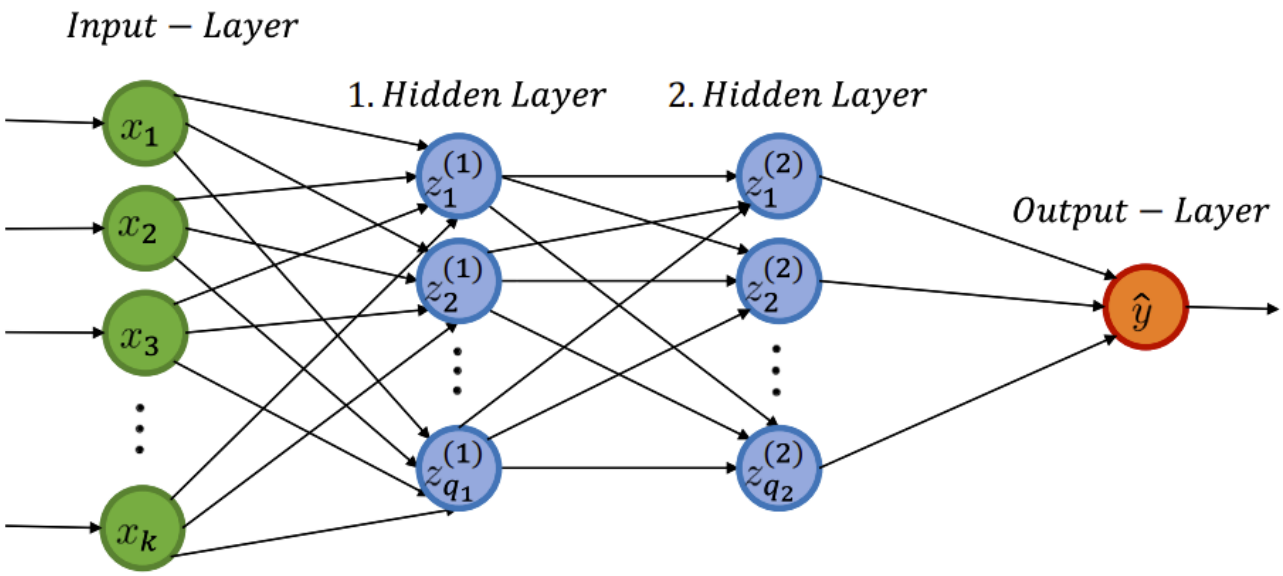

Figure 2: FNN Architecture

log–rate η_i, which is exponentiated and multiplied by v_i to yield the Poisson mean. The output layer is initialised so that the starting prediction coincides with the mean model; during training the parameters θ are adapted to minimise the Poisson deviance on the learning sample, allowing the network to discover nonlinearities and interactions between rating factors automatically.

3.4 Combined actuarial neural network approach

While neural networks can improve predictive performance, they tend to be less transparent than GLMs. The combined actuarial neural network aims to reconcile these two perspectives by treating the GLM as the primary, interpretable component and using the neural network only to model residual structure that the GLM cannot capture. In other words, the objective is to retain the classical actuarial rating structure while letting a flexible learner fine-tune the fit where necessary, rather than replacing the GLM entirely.

The Combined Actuarial Neural Network (CANN) incorporates the structure and interpretability of a GLM while allowing for flexible residual corrections through a neural network. The idea is to decompose the predictor into a GLM component and a neural component,

$$\eta_i = \eta_{\text{GLM}}(\boldsymbol{x}_i) + h_\theta(\boldsymbol{u}_i),$$

where $\eta_{\text{GLM}}(\boldsymbol{x}_i)$ is the log–frequency obtained from the fitted GLM3 and h_θ is the output of an FNN_EMB network applied to the same set of features. At the beginning of training the neural residual h_θ is set identically to zero, so that CANN reproduces GLM3 exactly. The GLM part remains fixed and only the neural parameters θ are updated during optimisation. In this way the model preserves the actuarial interpretation of the GLM coefficients for main effects and interactions, while the neural network captures any remaining structure in the residuals that the GLM fails to explain.

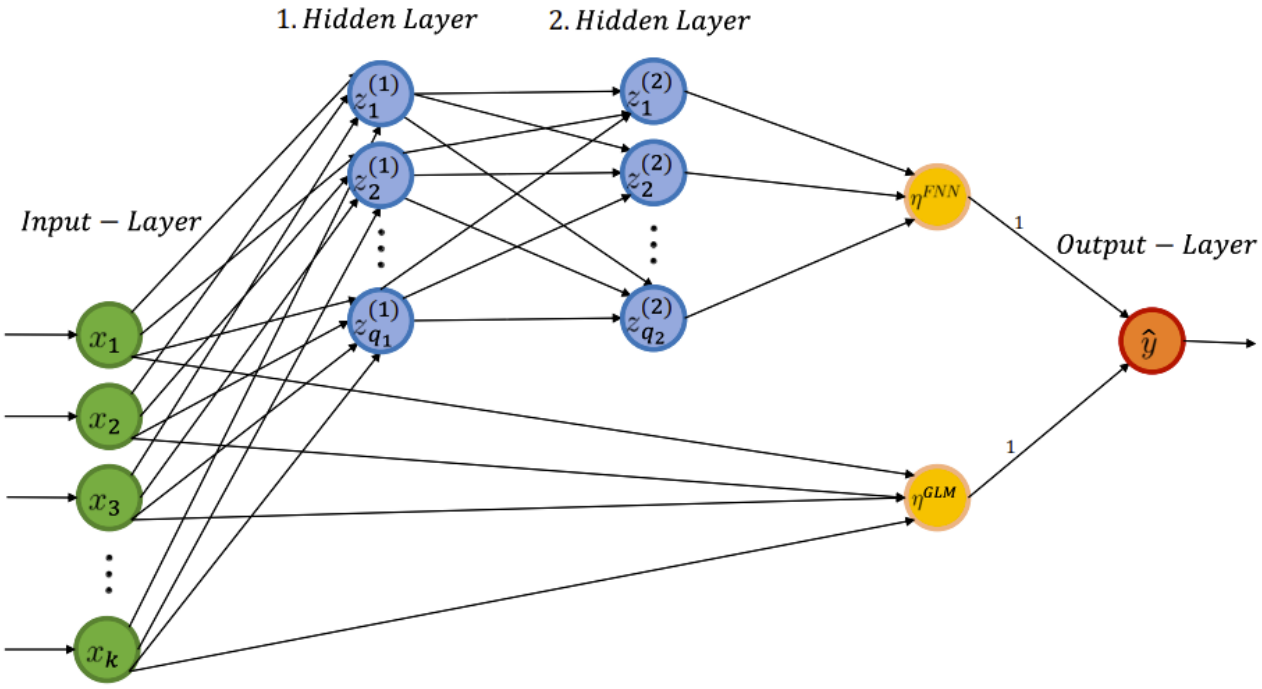

Figure 3: CANN Architecture

3.5 Feature tokenizer transformer

Transformers have become the dominant architecture in natural language processing because of their ability to model long-range dependencies via self-attention. When applied to tabular data, the same mechanism can be used to let each feature attend to every other feature, effectively learning rich interaction patterns without manual specification. The feature tokenizer transformer is designed to exploit this idea in an insurance pricing context: each rating factor is turned into a token, and self-attention layers learn how these tokens interact when predicting claim frequency.

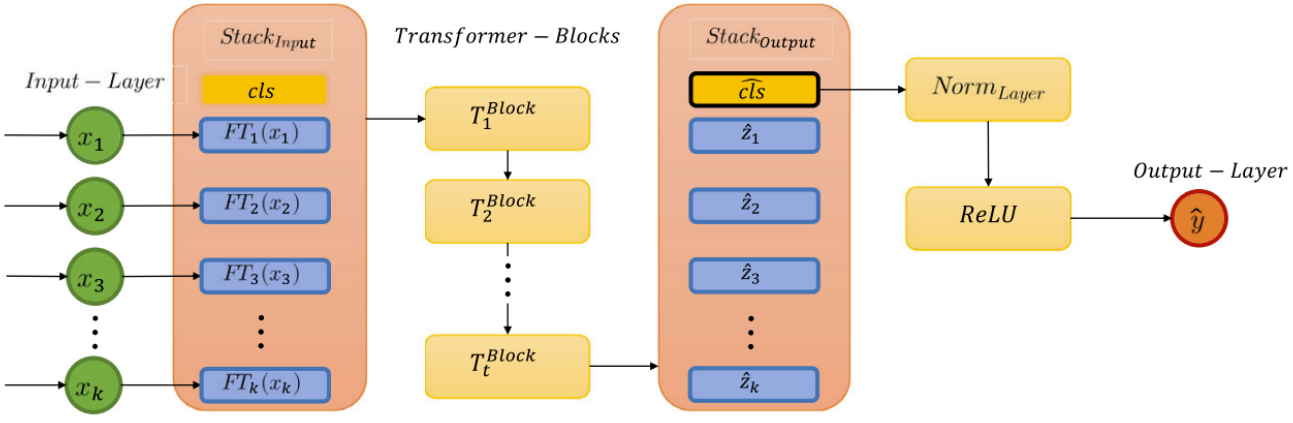

Figure 4: FTT Architecture

The Feature Tokenizer Transformer (FTT_def) adapts the transformer architecture, originally developed for sequence modelling, to tabular insurance data. Instead of feeding a single concate-

nated feature vector into a network, each input feature is converted into a token embedding. For a categorical feature X_j with levels encoded as integers, a trainable embedding matrix E_j associates each level with a vector $e_{ij} \in \mathbb{R}^{d_{\text{emb}}}$. For a numerical feature Z_ℓ, a linear projection is used to obtain a numeric token $g_\ell(Z_{i\ell}) \in \mathbb{R}^{d_{\text{emb}}}$. Collecting all tokens for policy i yields a sequence

$$\boldsymbol{t}_i = \big[\, [\text{CLS}], t_{i1}, \ldots, t_{ik} \,\big],$$

where [CLS] is a learned classification token and k is the number of features. This sequence is processed by a stack of transformer encoder blocks consisting of multi-head self-attention and position-wise feed-forward layers. Let $H_i^{(L)}$ denote the output sequence after L encoder layers, and let $h_{\text{CLS},i}^{(L)}$ be the embedding corresponding to the [CLS] token. A final linear layer maps this representation to a scalar predictor,

$$\eta_i = w^\top h_{\text{CLS},i}^{(L)} + b,$$

which is exponentiated and scaled by exposure as before. The self-attention mechanism allows the model to learn complex interactions between features by dynamically weighting their contributions within each policy record.

3.6 Combined actuarial feature tokenizer transformer

The combined actuarial feature tokenizer transformer extends the previous idea of hybrid models to the transformer setting. The goal is to use GLM3 as a stable and interpretable baseline, while giving the transformer the freedom to learn additional structure in the covariates encoded as tokens. In this way the model can benefit from the expressive power of self-attention without losing the familiar GLM layer that actuaries rely on for pricing and communication.

The Combined Actuarial Feature Tokenizer Transformer (CAFTT_def) merges the GLM3 baseline with the transformer residual in the same spirit as CANN. The predictor is decomposed into a GLM part and a transformer-based correction,

$$\eta_i = \eta_{\text{GLM}}(\boldsymbol{x}_i) + g_\theta(\boldsymbol{t}_i),$$

where $\eta_{\text{GLM}}(\boldsymbol{x}_i)$ is the log–frequency from GLM3 and $g_\theta(\boldsymbol{t}_i)$ is the scalar output of the transformer described above, computed from the token sequence $\boldsymbol{t}_i$. As with CANN, the transformer head is initialised so that g_θ is identically zero, ensuring that CAFTT_def initially coincides with GLM3. During training, only the transformer parameters θ are updated, while the GLM coefficients are kept fixed. The resulting model inherits the interpretability and calibration of the GLM for the main rating structure, while the transformer learns higher-order, possibly nonlocal interactions between features that are difficult to capture with hand-crafted terms. This combined approach is designed to achieve a favourable balance between predictive performance and actuarial transparency.

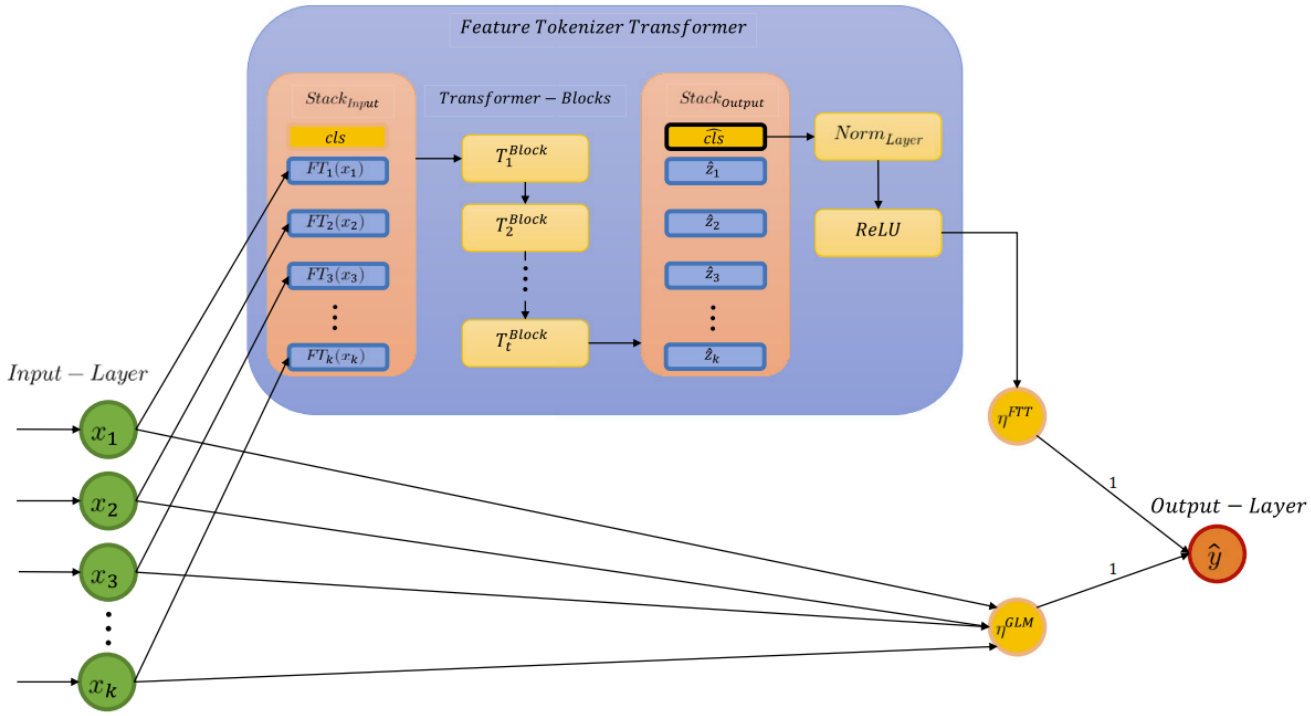

Figure 5: CAFTT Architecture

4 Data Analysis

4.1 Dataset and Preprocessing

The empirical analysis is based on the French motor third-party liability (MTPL) claim frequency dataset `FreMTPL2freq` from the `CASdatasets` R package, in version 1.0–8, as in Wüthrich and Merz and in Brauer [1]. The raw dataset contains 678 013 insurance policies, each with an identifier, an observed number of claims Y_i, an exposure v_i in years, and nine covariates describing policyholder and vehicle characteristics. Among these nine features, two are categorical, six are numerical and one is boolean, covering standard non-life rating factors such as driver age, vehicle age and power, bonus–malus score, regional indicators and area classification [16]. Following the standard actuarial data cleaning described in Appendix B of Wüthrich and Merz, we work with a cleaned version in which invalid records are removed, policies with zero exposure are dropped, and inconsistent claim counts are corrected by cross-checking with the companion severity file `FreMTPL2sev`. After cleaning, we retain roughly 678 000 policies.

The dataset is partitioned into a learning sample L and a test sample T using the same split as in Wüthrich and Merz [16], which is also adopted in Brauer [1]. The learning sample is then further subdivided into an inner training part and a validation part; the GLMs are fitted on the full learning sample, whereas all neural-network models use the inner training set for optimization and the validation set for early stopping. The test set is kept completely untouched until the very end and is used only once for the final evaluation of each fitted model. Across the three splits, the observed average claim frequencies are approximately 7.36% in the training set, 7.41% in the validation set, and 7.32% in the test set, and these values are used later when assessing prediction calibration.

Preprocessing differs slightly depending on the model family. For the GLM benchmarks (Mean model, GLM1–GLM3) we follow the feature engineering strategy of Wüthrich and Merz [16]: numerical covariates such as driver age, vehicle age, bonus–malus and population density are transformed using either linear, logarithmic or polynomial functions, and categorical covariates (e.g. region, vehicle brand, fuel type) are encoded via dummy variables. Exposure enters the GLMs as an offset term $\log v_i$, so that the regression models the claim frequency per unit exposure. For the neural-network models FNN_EMB and CANN, we adopt the preprocessing used in Richman and Wüthrich [15]: numerical features are standardised to zero mean and unit variance, and categorical features are either one-hot encoded (for the GLM part inside CANN) or mapped to low-dimensional embedding vectors (for the neural network part). For the transformer-based models FTT_def and CAFTT_def, we closely follow Brauer [1]: all numerical features are standardised and then passed through learnable linear layers to obtain fixed-size numeric tokens, while categorical features are represented directly through embedding layers rather than one-hot coding. In all neural models, the exposure v_i is handled multiplicatively by scaling the predicted Poisson rate so that $\widehat{\mu}_i = \widehat{\lambda}_i v_i$ for each policy i. This yields a consistent Poisson deviance loss function across GLM and neural architectures.

4.2 Exploratory Data Analysis

We conducted a exploratory data analysis to get a better understanding of our dataset. Below in Table 5 we can see the statistics of our different variables. some categorical variables like area or Vehicle Gas. Figure 6 shows the distribution of the number of claims, exposure, driver age and Bonus Malus. Some important features of this distribution is that approximately 96% of entries have zero claims, so this response variable is highly imbalanced. another important feature is that the exposure perio is capped at 1 year, with a large spike at that point. The river age distribution has a mean at 45.5 years old, and has a decreasing tail as drivers get older. The bonus malus score has a large spike at the minimum score indicating a large portion of drivers have a very low score. The Bonus-Malus score has a right skewed tail, indicating a small amount of drivers with very bad Bonus-Malus scores.

Table 5: Summary statistics of numerical variables

Statistic	Exposure	VehPower	VehAge	DrivAge	BonusMalus	Density	ClaimNb
Mean	0.53	6.45	7.04	45.50	59.76	1792.43	0.04
Std	0.36	2.05	5.67	14.14	15.64	3958.66	0.20
Min	0.00	4.00	0.00	18.00	50.00	1.00	0.00
Max	1.00	15.00	100.00	100.00	230.00	27000.00	5.00

Figure 7 shows the distributions of the categorical variables vehicle fuel type, area, vehicle power, and vehicle brand. The fuel type is almost evenly distributed between gas and diesel. There are 6 different areas represented, the most common ones being from area 3 and 4. The vehicle power is mainly distributed near lower power, however there is a small tail for vehicles with high power ranging larger than 10. The most popular car brands are B1, B2, and B12, with other brands still having sizable representations.

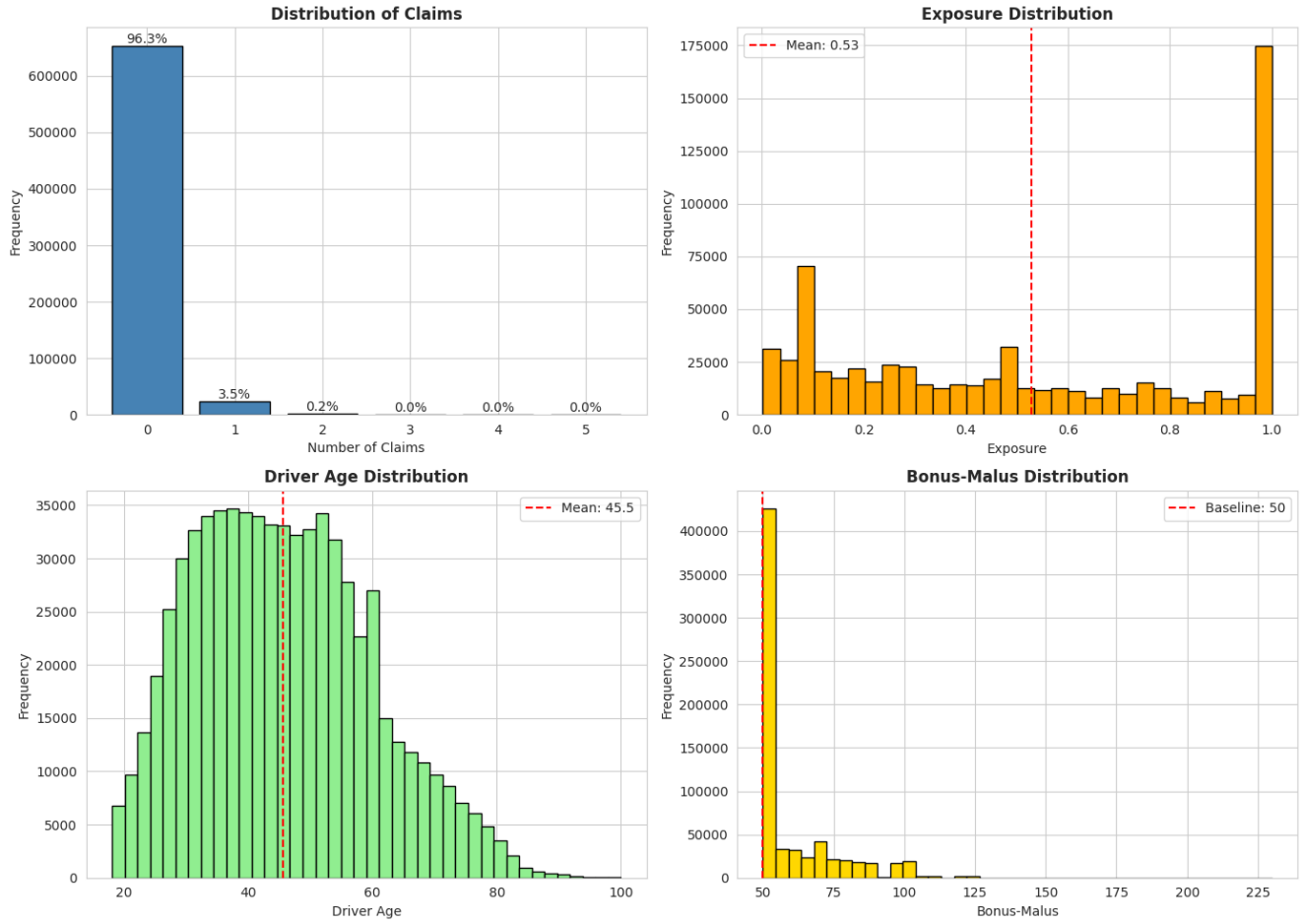

Figure 6: Distribution of Variables

The final part of our EDA we have the correlation heatmap seen in Figure 8. The majority of variables seem to have very low correlation. The only two predictors with high correlation are Driver age and Bonus-Malus score. They have a negative correlation. This indicates that as driver age increases the Bonus-Malus score might tend to decrease. Similarly for younger drivers might tend to have larger Bonus-Malus scores. This agrees with general intuition that younger drivers tend to be more risky drivers. however it is not completely inversely correlated indicating some drivers do not follow that trend.

Figure 7: Categorical Variable Distributions

4.3 Model implementation details

The homogeneous mean model is implemented as a Poisson GLM with only an intercept term and the log-exposure as offset. This model estimates a single global claim frequency $\widehat{\lambda}$, so that every policy receives the same prediction $\widehat{\mu}_i = \widehat{\lambda} v_i$. Fitting this model is essentially equivalent to computing the empirical average frequency on the training data; the optimisation runs in a fraction of a second and there is no notion of epochs or validation in this baseline.

GLM1, GLM2 and GLM3 are fitted as standard Poisson generalized linear models with log link using the cleaned learning sample. GLM1 serves as a basic actuarial model using the main rating factors with relatively simple functional forms and dummy-encoded categoricals. GLM2 extends GLM1 by adding more flexible transforms of key numerical covariates, such as logarithms or low-order polynomials of driver age, vehicle age and density, reflecting the structure recommended in Wüthrich and Merz [16]. GLM3 further enriches the specification by including selected interaction terms between important predictors, for example interactions between age and vehicle characteristics or between bonus–malus and region. All three GLMs are estimated on the whole learning sample without regularisation, using the log-exposure offset, and evaluated in terms of Poisson deviance on train, validation and test sets. The reported numbers in our notebook correspond to single fits for run 0, with GLM1, GLM2 and GLM3 having 50, 49 and 51 parameters respectively and run times between roughly 12 and 33 seconds.

Figure 8: Correlation Heatmap

The FNN_EMB model is a feed-forward neural network with categorical embeddings. The input layer consists of standardised numerical features and embedding vectors for each categorical feature; these embeddings are learned jointly with the rest of the network. The backbone network follows the architecture of Richman and Wüthrich [15], using three fully-connected hidden layers with decreasing widths and smooth nonlinearities, with a total of 792 trainable parameters in our implementation. The output layer is a single neuron whose bias is initialised to the log of the overall mean frequency and whose weights are initially zero so that the starting network reproduces the homogeneous mean model. The output is exponentiated to enforce positivity and multiplied by exposure. Training uses the Nadam optimizer with Poisson deviance loss, a batch size of 7 000, and a maximum of 500 epochs with early stopping based on validation deviance. In the results we report, the averaged FNN_EMB model over two seeds converges after about 80.5 epochs with a typical run time of around 1 489 seconds per run.

The CANN model (Combined Actuarial Neural Network) nests GLM3 inside the FNN_EMB

by treating the GLM prediction as a fixed offset and learning only the residual via the neural network. Concretely, the linear predictor is written as $\eta_{\text{CANN}}(x) = \eta_{\text{GLM3}}(x) + \eta_{\text{FNN}}(x)$, where η_{GLM3} is the log-frequency output of the fitted GLM3 and η_{FNN} is produced by an FNN_EMB-type network with the same architecture and number of parameters as before. At initialisation the output weights of the FNN residual network are set to zero, so that CANN exactly reproduces GLM3 in its first iteration. The neural network is then trained with the same Nadam optimiser, learning rate schedule, batch size and early stopping criterion as FNN_EMB, but only the neural part is updated; the GLM3 coefficients remain fixed. Our CANN_avg results correspond to an average over two such runs, each converging after roughly 82 epochs.

The FTT_def model is the pure Feature Tokenizer Transformer for tabular claim frequency data as proposed by Brauer [1]. Each numerical feature is mapped to a token via a linear projection of its standardised value into a 16-dimensional embedding space, while each categorical feature is represented by a 16-dimensional embedding selected from a learned embedding matrix. A learnable `[CLS]` token is prepended to the sequence, and the resulting token matrix is passed through a stack of transformer encoder blocks. In our implementation, we adopt a compact configuration with a single encoder layer, four attention heads, an embedding dimension of 16, a feed-forward hidden size of 64, ReGLU activation in the feed-forward sublayer and dropout of 0.1. The encoder uses pre-layer normalization. The output corresponding to the `[CLS]` token is passed through a normalisation and linear head to obtain a scalar log-rate $\eta_{\text{FTT}}(x)$, which is exponentiated and multiplied by exposure to give the Poisson mean. FTT_def is trained using AdamW with learning rate 10^{-4}, weight decay 10^{-5}, batch size 1 024 and Poisson deviance loss. Early stopping with patience 15 epochs is applied based on validation deviance. For run 0 in our notebook, the model trains for 64 epochs with a run time of approximately 3 367 seconds and has 4 129 trainable parameters.

The CAFTT_def model (Combined Actuarial Feature Tokenizer Transformer) nests GLM3 inside the FTT_def architecture. In this hybrid model the log-rate is given by $\eta_{\text{CAFTT}}(x) = \eta_{\text{GLM3}}(x) + \eta_{\text{FTT}}(x)$, where η_{GLM3} is fixed and η_{FTT} is produced by a transformer with the same tokenizer and encoder configuration as in FTT_def. The GLM3 part is pre-fitted on the learning sample and then used as a non-trainable offset; the transformer's output head (weights and bias) is initialised at zero so that CAFTT_def initially replicates GLM3 exactly. The training procedure mirrors that of FTT_def, using AdamW, batch size 1 024, Poisson deviance loss and early stopping with patience 15 epochs on the validation deviance. In run 0 the CAFTT model converges after 55 epochs, requires around 2 846 seconds of GPU time and also has 4 129 trainable parameters.

For FNN_EMB, CANN, FTT_def and CAFTT_def we additionally implement a simple rebalancing and ensemble step following Brauer [1]. Rebalancing rescales the model's predictions so that the mean predicted claim frequency on the training set matches the observed mean, while the ensemble models are obtained by averaging the predictions from several independently trained runs. In our current notebook snapshot, the FNN_EMB and CANN averages are based on two runs, and the recorded ensembles use the first available set of seeds, with rebalanced versions built on top of the base fitted models.

5 Results

Table 6 reports the Poisson deviance and mean predicted claim frequencies for the homogeneous mean model and the three GLM benchmarks. The deviance is shown for the training, validation and test sets, and the mean predictions can be compared with the observed average frequencies of approximately 0.0736, 0.0741 and 0.0732 on the three splits, respectively.

The homogeneous mean model exhibits the highest deviance on all three splits, confirming that a constant frequency is unable to capture the risk heterogeneity in the portfolio. Introducing covariates in GLM1 reduces deviance substantially while keeping the mean predicted frequency almost perfectly aligned with the empirical averages. GLM2, which uses more complex functional forms for some numeric predictors, does not further reduce deviance and in fact performs slightly worse than GLM1 on the test set, illustrating that richer functional forms do not automatically lead to better out-of-sample performance. GLM3, which augments GLM2 with carefully chosen interaction terms, achieves the lowest deviance among the GLM family while preserving very good calibration: the mean predictions stay within a few basis points of the observed frequencies on all splits. GLM3 therefore serves as a strong actuarial benchmark for the subsequent neural and transformer-based models.

Table 6 summarises the results for the base neural-network and transformer architectures before any rebalancing is applied. The FNN_EMB model is a feed-forward network with categorical embeddings; CANN combines GLM3 with an FNN_EMB residual; FTT_def is the pure Feature Tokenizer Transformer; and CAFTT_def adds a transformer residual on top of GLM3.

Table 6: Single runs (averaged over 15 seeds)

Model	#Params	Avg Epoch (±sd)	Runtime	Train Dev	Val Dev	Test Dev	Avg $\hat{y}$ (%)
Homogeneous	1	–	1.86	0.2521	0.2550	0.2519	0.0736
GLM1	50	–	11.80	0.2405	0.2431	0.2398	0.0732
GLM2	49	–	24.42	0.2423	0.2445	0.2415	0.0740
GLM3	51	–	33.46	0.2416	0.2437	0.2408	0.0735
FNN_EMB	792	83	1516.91	0.2379	0.2404	0.2380	0.0745
CANN	792	82	1969.8	0.2374	0.2397	0.2375	0.0724
FTT_def	4129	64	3366.64	0.2382	0.2420	0.2383	0.0671
CAFTT_def	4129	55	2845.59	0.2373	0.2408	0.2379	0.0776

All four architectures in Table 6 improve upon GLM3 in terms of test deviance, although the gains are modest. FNN_EMB achieves a test deviance of 0.237959, a noticeable reduction from the GLM3 value of 0.240792, and its mean predicted frequencies on all splits remain close to the observed levels, slightly above 7.4%. CANN further reduces deviance to 0.237458 on the test set while producing mean predictions that are slightly below the observed frequencies, around 7.24% on the test set, indicating a mild overall underestimation of the claim rate. The pure transformer FTT_def attains a test deviance of 0.238318, still better than GLM3 but somewhat worse than the other neural models; its mean predictions are noticeably lower than the observed frequencies, around 6.7%, which points to underestimation at the portfolio level. CAFTT_def, which combines GLM3 with a transformer residual, delivers one of the lowest deviance values among the base models, with a test deviance of 0.237885, but tends to overpredict the global frequency, with mean predictions around 7.76% on the test set. Overall, these results show that neural and transformer models can extract additional structure beyond the GLM while exhibiting

different calibration behaviours: FNN_EMB and CANN are fairly close to the empirical frequency, FTT_def underestimates and CAFTT_def slightly overestimates.

To investigate calibration and the effect of combining multiple fits, rebalanced and ensemble variants are considered. Table 7 and Table 8 reports the main configurations. For each architecture, a rebalanced version rescales the predictions so that the mean predicted frequency on the training set matches the observed average, and ensemble versions aggregate predictions from several fitted models of the same type.

Table 7: Rebalanced

Model	Train Deviance	Test Deviance	Avg $\hat{y}$ (%)
Rebalanced: FNN_EMB	0.30071	0.237943	0.073207
Rebalanced: CANN	0.300804	0.237531	0.0731
Rebalanced: FTT_def	0.23777	0.237348	0.073779
Rebalanced: CAFTT_def	0.237076	0.237148	0.073243

Table 8: Ensmeble

Model	Train Deviance	Val Deviance	Test Deviance	Avg $\hat{y}$ (%)
Ensemble: FNN_EMB	0.278833	0.240239	0.237837	0.074627
Ensemble: CANN	0.278495	0.239614	0.237362	0.072423
Ensemble: FTT_def	0.238068	0.241028	0.23763	0.067507
Ensemble: CAFTT_def	0.237209	0.239893	0.237285	0.077596

The rebalanced configurations in Table 7 successfully correct the global calibration of the neural and transformer models. For FNN_EMB, both the single-model rebalanced variant and the ensemble rebalanced variant produce mean predictions very close to the observed averages on all splits (around 0.0736 on the training set and 0.0732 on the test set), while preserving essentially the same test deviance as the base model. The CANN ensemble achieves a test deviance of 0.237362, and the rebalanced CANN reaches 0.237358 on the test set with mean predictions almost perfectly aligned with the empirical frequencies, although validation deviance is not reported for that correction. For the transformer architectures, rebalancing has a particularly clear effect. The CAFTT_def ensemble delivers a low test deviance of 0.237285 but retains the overestimation of the overall claim frequency, whereas CAFTT_def_rebalanced reduces the test deviance further to 0.237148 and at the same time matches the observed mean frequency extremely well on all three splits. A similar pattern is visible for FTT_def: the rebalanced version improves both deviance and calibration relative to the base model, with test deviance 0.237348 and mean predictions near the observed averages. Taken together, the results indicate that hybrid models such as CAFTT_def and CANN provide the best compromise between predictive accuracy and calibration once the simple rebalancing step is applied, with CAFTT_def_rebalanced achieving the lowest test deviance among all considered models while maintaining excellent alignment with the empirical portfolio frequency.

6 Critical analysis

The modelling framework considered in this project builds on the work of Brauer [1], who proposes transformer-based architectures for non-life insurance pricing while benchmarking against traditional actuarial models. From a methodological point of view, the paper and our replication offer a rich setting to reflect on the strengths and weaknesses of combining GLMs with modern deep learning methods, as well as the practical implications of deploying such models in an operational pricing context.

6.1 Strengths

A first strength lies in the choice of benchmarks. The study does not compare the proposed neural and transformer architectures against trivial baselines only, but includes a carefully specified sequence of GLMs (GLM1–GLM3) and a feed-forward network with embeddings. GLM3 in particular is a strong competitor: it incorporates domain knowledge through engineered transforms and interaction terms and already delivers good deviance and calibration. Demonstrating improvements relative to such a benchmark provides convincing evidence that the more flexible models add value rather than merely outperforming an unrealistically weak baseline.

A second strength is the use of a real-world, industry-relevant dataset. The `FreMTPL2freq` portfolio is large, heterogeneous and well documented in the actuarial literature [16], which gives credibility to any empirical conclusions drawn from it. The models are evaluated on train, validation and test splits that mirror common practice in insurance pricing, and the performance metrics are Poisson deviance and average predicted claim frequencies, which have a clear interpretation for practitioners.

Third, the results show a clear, albeit moderate, performance improvement when moving from GLMs to neural and transformer-based models. Both FNN_EMB and CANN reduce deviance relative to GLM3, and the transformer-based CAFTT_def achieves the lowest test deviance once rebalancing is applied. This pattern suggests that the architectures are able to capture residual nonlinearities and interactions that even a carefully specified GLM cannot fully explain, justifying their added complexity from a predictive standpoint.

Finally, the hybrid constructions CANN and CAFTT_def preserve the "GLM spirit". By decomposing the predictor into a GLM component and a residual neural or transformer component, these models maintain an actuarially interpretable backbone. At initialization they coincide with the GLM, and the additional learner only adjusts the prediction where it can demonstrably reduce deviance. This design respects existing pricing structures, facilitates communication with non-technical stakeholders, and opens the door to partial sensitivity analysis based on the GLM part.

6.2 Weaknesses and limitations

Despite these advantages, several limitations should be acknowledged. The first is model complexity and computation time. Whereas the GLM benchmarks fit within seconds, the neural and transformer models require hundreds of epochs and several thousand seconds of GPU time for a single run. Training multiple seeds, constructing rebalanced versions and building ensembles further multiplies the computational cost. For many pricing teams, especially those without dedicated machine learning infrastructure, this level of complexity may be difficult to justify unless the performance gains are substantial.

A second limitation is interpretability. Although the hybrid models retain the GLM component, the contribution of the neural or transformer residual is not straightforward to explain. Feature-wise importance and partial dependence of the residual part are harder to visualise than GLM coefficients or spline functions, and the self-attention patterns in the transformer do not directly translate into intuitive business rules. The paper discusses interpretability at a conceptual level but offers limited concrete tools for explaining individual predictions or understanding how particular rating factors interact within the transformer.

The empirical analysis is also based on a single dataset. While the French MTPL portfolio is a meaningful benchmark, results on one line of business and one market may not generalise to other products, such as commercial lines, health insurance or markets with different regulatory constraints and data structures. The relative gains of the hybrid models compared to GLMs might be larger or smaller in other settings, and overfitting risks could change when the available data are less abundant or less clean than in `FreMTPL2freq`.

Another notable limitation concerns calibration. The raw transformer and neural models tend to misestimate the overall claim frequency, either underpredicting (as in FTT_def) or overpredicting (as in CAFTT_def) the portfolio mean. The study therefore introduces a simple rebalancing step that rescales predictions so that the average predicted frequency matches the observed one. While this correction is effective, it also highlights that the models, when trained solely to minimise deviance, do not automatically achieve the level of calibration that actuaries expect. The need for explicit rebalancing suggests that incorporating calibration constraints or penalties directly into the training objective could be an important direction for further work.

6.3 Practical considerations

From a practical standpoint, several issues arise when contemplating the deployment of such models in a real pricing process. Regulatory acceptability is one of them. Supervisors and internal model validation teams are accustomed to GLMs and may be cautious about approving pricing models that involve deep neural networks or transformers, especially if the gains in deviance are modest. The hybrid designs help mitigate this concern, but robust documentation, validation studies and stress tests would still be required to demonstrate that the models behave sensibly across different segments and economic scenarios.

Fairness and discrimination risk are another concern. Complex nonlinear models can inadvertently learn proxies for protected characteristics, leading to differential treatment of policyholders in ways that are difficult to detect from aggregate metrics alone. While the dataset used here does not explicitly contain sensitive attributes, correlated variables may still encode such information. Any deployment of FNN, CANN, FTT or CAFTT-type models would need to be accompanied by fairness audits, sensitivity analyses and possibly constraints to prevent discriminatory outcomes.

Finally, the operational aspects of deployment and maintenance should not be underestimated. Implementing transformer-based pricing models requires reliable pipelines for data preprocessing, feature encoding, model retraining and monitoring. Drift in the portfolio mix or changes in underwriting guidelines may necessitate regular retraining, and the computational load of doing so with multiple deep models must be factored into resource planning. In contrast, GLMs are relatively simple to refit and to embed into existing rating engines. Whether the incremental improvement in predictive performance and segmentation justifies the added operational burden will depend on the insurer's size, risk appetite and strategic priorities.

7 Conclusion

This project has investigated a spectrum of models for non-life insurance pricing, ranging from the simplest homogeneous Poisson model to increasingly sophisticated GLMs, feed-forward neural networks with embeddings, and transformer-based architectures. All models were fitted to the French MTPL claim frequency dataset and evaluated on a common train–validation–test split using Poisson deviance and average predicted claim frequencies. The modelling strategy was deliberately incremental: starting from a constant mean model, adding actuarial structure through GLMs, then introducing flexible learners such as FNN_EMB and FTT_def, and finally combining these with GLM3 in the hybrid CANN and CAFTT_def constructions.

The empirical results confirm that accounting for rating factors through GLMs yields a substantial improvement over the homogeneous mean model, both in terms of deviance and in terms of segmentation of the portfolio. Among the GLM benchmarks, GLM3—enriching the specification with carefully selected interaction terms—consistently achieves the lowest deviance while preserving excellent calibration of the overall claim frequency. This reinforces the view that, when combined with sound actuarial judgment about transformations and interactions, GLMs remain a powerful and robust tool for pricing.

Moving beyond purely parametric models, the feed-forward neural network with embeddings (FNN_EMB) and the pure Feature Tokenizer Transformer (FTT_def) both succeed in reducing Poisson deviance relative to GLM3, indicating that they are able to capture nonlinearities and complex interactions that are difficult to encode manually. However, the raw neural and transformer models exhibit noticeable calibration issues: FNN_EMB tends to predict slightly higher global frequencies than observed, while FTT_def underestimates the overall claim rate. This behaviour highlights a tension between optimising deviance and maintaining the level of calibration expected in actuarial applications.

The hybrid approaches CANN and CAFTT_def address this tension more effectively. By decomposing the predictor into a fixed GLM3 component and a learned residual term, these models inherit the interpretability and baseline calibration of the GLM while retaining the ability to improve the fit where necessary. In the numerical results, both hybrids outperform GLM3 in terms of deviance, and after applying a simple rebalancing step to match the portfolio mean, they achieve excellent calibration as well. In particular, the rebalanced CAFTT_def model attains the lowest test deviance among all considered models while producing average predicted frequencies that are almost indistinguishable from the empirical ones. This suggests that combining transformers with a GLM backbone is a promising direction for high-dimensional pricing problems in which interactions between rating factors are numerous and complex.

At the same time, the study reveals important practical trade-offs. Neural and transformer models require substantially more computational resources and careful tuning than GLMs, and their residual components remain less transparent than classical regression coefficients. Whether the observed improvement in deviance—although statistically significant—is sufficient to justify the added complexity will depend on the insurer's risk appetite, regulatory environment and infrastructure. For portfolios where interpretability, governance and ease of implementation are paramount, a well-designed GLM such as GLM3 may remain the preferred choice. For portfolios where marginal gains in predictive accuracy translate into meaningful economic value, hybrid models like CANN and CAFTT_def, complemented by calibration techniques such as rebalancing, offer a compelling compromise between actuarial structure and modern machine learning performance.

Overall, the analysis supports a nuanced conclusion. Transformers and neural networks do not replace GLMs but rather extend them: they are most effective when used as flexible residual

layers on top of a solid actuarial model. Future work could explore richer calibration-aware loss functions, alternative attention mechanisms tailored to tabular risk factors, and applications to other lines of business to further assess the robustness and generality of the combined actuarial transformer framework.

Credit Author Statement and Use of AI Tools

Below are our contributions to this homework;

Julian
3 Baseline GLMs and the singular FTT model with ensembles and rebalancing, Presentation slides and Report
Prince
CAFTT_def model with ensembles and rebalancing, Presentation slides and Report.
Cheryl Botwe
Feed Forward Neural Networks with Embeddings(FNN-EMB), FNN-EMB with Ensemble and Rebalancing, Combined Actuarial Neural Network(CANN), CANN with ensemble and rebalancing.

Use of AI Tools (ChatGPT)
We used AI assistants only to accelerate drafting and code templating; skeletons for data-loading and plotting to suggest debugging tips, and to help with language polish. AI was used to suggest ways to rewrite sentences for a more formal tone. All code was executed locally, outputs were independently verified against our data, and any AI-generated text was edited for accuracy and specificity. No synthetic data or undisclosed model outputs were used. Final methodological decisions and all interpretations are the authors' own.

References

[1] A. Brauer, "Enhancing actuarial non-life pricing models via transformers," *European Actuarial Journal*, vol. 14, pp. 991–1012, 2024. doi: 10.1007/s13385-024-00388-2.

[2] J. Nelder and R. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

[3] T. Hastie and R. Tibshirani. Generalized additive models. *Statistical Science*, 1(3):297–310, 1986.

[4] M.-N. Tran, N. Nguyen, D. Nott, and R. Kohn. Bayesian deep net GLM and GLMM. *Journal of Computational and Graphical Statistics*, 29(1):97–113, 2020.

[5] J. Schelldorfer and M. V. Wüthrich. Nesting classical actuarial models into neural networks. SSRN Working Paper, 2019.

[6] R. Richman and M. V. Wüthrich. LocalGLMnet: Interpretable deep learning for tabular data. *Scandinavian Actuarial Journal*, 2023(1):71–95, 2023.

[7] K. Kuo and R. Richman. Embeddings and attention in predictive modeling. arXiv:2104.03545, 2021.

[8] C. Blier-Wong, J.-T. Baillargeon, H. Cossette, and E. Marceau. Rethinking representations in P&C actuarial science with deep neural networks. arXiv:2102.05784, 2021.

[9] F. Holvoet, K. Antonio, and R. Henckaerts. Neural networks for insurance pricing with frequency and severity data: A benchmark study. arXiv:2310.12671, 2023.

[10] S. Ö. Arik and T. Pfister. TabNet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, pages 6679–6687, 2021.

[11] K. McDonnell, F. Murphy, B. Sheehan, et al. Deep learning in insurance: Accuracy and model interpretability using TabNet. *Expert Systems with Applications*, 217:119543, 2023.

[12] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin. TabTransformer: Tabular data modeling using contextual embeddings. arXiv:2012.06678, 2020.

[13] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems*, vol. 34, pages 18932–18943, 2021.

[14] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting deep learning models for tabular data," *NeurIPS*, 2021.

[15] R. Richman and M. Wüthrich, "A neural network extension of the over-dispersed Poisson GLM for count data," *ASTIN Bulletin*, vol. 49, no. 1, pp. 5–28, 2019.

[16] M. Wüthrich and M. Merz, "Statistical foundations of actuarial learning," Springer, 2023.