Individual Claims History Simulation Machine

Andrea Gabrielli[*] Mario V. Wüthrich[*]

February 26, 2018

Abstract

The aim of this project is to develop a stochastic simulation machine that generates individual claims histories of non-life insurance claims. This simulation machine is based on neural networks to incorporate individual claims feature information. We provide a fully calibrated stochastic scenario generator that is based on real non-life insurance data. This stochastic simulation machine allows everyone to simulate his own synthetic insurance portfolio of individual claims histories and back-test his preferred claims reserving method.

Keywords: Claims reserving, individual claims, claims cash flows, micro-level stochastic reserving, loss reserving, claims simulation, neural network reserving, individual claims features, individual claims covariates, chain-ladder.

1 Description of the project

The aim of this project is to develop a stochastic simulation machine that generates individual claims histories of non-life insurance claims. These individual claims histories should depend on individual claims feature information such as the line of business concerned, the claims code involved or the age of the injured. This feature information should influence the reporting delay of the individual claim, the claim amount paid, its individual cash flow pattern as well as its settlement delay. The resulting (simulated) individual claims histories should be as 'realistic' as possible so that they may reflect a real insurance claims portfolio. These simulated claims then allow us to back-test classical aggregate claims reserving methods, such as the chain-ladder method, as well as to develop new claims reserving methods which are based on individual claims histories. The latter has become increasingly popular in actuarial science, see [1, 3, 5, 7, 8, 10, 11, 13] for recent developments. A main shortcoming in this field of research is that there is no publicly available individual claims history data. Therefore, there is no possibility to back-test the proposed individual claims reserving methods. For this reason, we believe that this project is very beneficial to the actuarial community because it provides a common ground and publicly available (synthetic) data for research in the field of individual claims reserving.

This paper is divided into 4 sections. In this first section we describe the general idea of the simulation machine as well as the chosen data used for model calibration. In Section 2 we

[*] ETH Zurich, RiskLab, Department of Mathematics, 8092 Zurich, Switzerland

describe the design of our individual claims history simulation machine using neural networks. Section 3 focuses on the calibration of these neural networks. In Section 4 we carry out a use test by comparing the real data to the synthetically generated data in a chain-ladder claims reserving analysis. Appendix A presents descriptive statistics of the real data. Since the real insurance portfolio is confidential, we also design an algorithm to generate synthetic insurance portfolios of a similar structure as the real one, see Appendix B. Finally, in Appendix C we provide sensitivity plots of selected neural networks.

1.1 Description of the simulation machine

The simulation machine is programmed in the language R. The corresponding `.zip`-folder can be downloaded from the website:

`https://people.math.ethz.ch/~wmario/simulation.html`

This `.zip`-folder contains all parameters, a file `readme.pdf` which describes the use of our R-functions, as well as the two R-files `Functions.V1` and `Simulation.Machine.V1`. The first R-file `Functions.V1` contains the two R-functions `Feature.Generation` and `Simulation.Machine`. The former is used to generate synthetic insurance portfolios (this is described in more detail in Appendix B) and the latter to simulate the corresponding individual claims histories (this is described in the main body of this manuscript). The R-file `Simulation.Machine.V1` demonstrates the use of these two R-functions, also providing a short chain-ladder claims reserving analysis.

1.2 Procedure of developing the simulation machine

In recent years, neural networks have become increasingly popular in all fields of machine learning. They have proved to be very powerful tools in classification and regression problems. Their drawbacks are that they are rather difficult to calibrate and, once calibrated, they act almost like black boxes between inputs and outputs. Of course, this is a major disadvantage in interpretation and getting deeper insight. However, the missing interpretation is not necessarily a disadvantage in our project because it implies that, in back-testing other methods, the true data generating mechanism cannot easily be guessed.

To construct our individual claims history simulation machine, we design a neural network architecture. This architecture is calibrated to real insurance data consisting of $n = 9'977'298$ individual claims that have occurred between 1994 and 2005. For each of these individual claims, we have full information of 12 years of claims development as well as the relevant feature information. Together with a portfolio generating algorithm (Appendix B), one can then use the calibrated simulation machine to simulate as many individual claims development histories as desired.

1.3 The chosen data

The chosen data has been preprocessed correcting for wrong entries, for instance, an accident date that is bigger than the reporting date, etc. Moreover, we have dropped claims with missing feature components, for instance, if the age of the injured was missing. However, this was a negligible number of claims that we had to drop, and this does not distort the general calibration. The final (cleaned) data set consists of $n = 9'977'298$ individual claims histories. The following feature information is available for each individual claim:

- the claims number ClNr, which serves as a distinct claims identifier;
- the line of business LoB, which is categorical with labels in $\{1, \ldots, 4\}$;
- the claims code cc, which is categorical with labels in $\{1, \ldots, 53\}$ and denotes the labor sector of the injured;
- the accident year AY, which is in $\{1994, \ldots, 2005\}$;
- the accident quarter AQ, which is in $\{1, \ldots, 4\}$;
- the age of the injured age (in 5 years age buckets), which is in $\{15, 20, \ldots, 70\}$;
- the injured part inj_part, which is categorical with labels in $\{1, \ldots, 99\}$ and denotes the part of the body injured;
- the reporting year RY, which is in $\{1994, \ldots, 2016\}$.

For all claims $i = 1, \ldots, n$, we are given the individual claims cash flow $(C_i^{(j)})_{0 \leq j \leq 11}$, where $C_i^{(j)}$ is the payment for claim i in calendar year $\texttt{AY}_i + j$, and where $\texttt{AY}_i$ denotes the accident year of claim i. Note that we only consider yearly payments, i.e. multiple payments and recovery payments within calendar year $\texttt{AY}_i + j$ are aggregated into a single, annual payment $C_i^{(j)}$. This single, annual payment can either be positive or negative, depending on having either more claim payments or more recovery payments in that year. The sum over all yearly payments $\sum_j C_i^{(j)}$ of a given claim i has to be nonnegative because recoveries cannot exceed payments (this is always the case in the considered data). Remark that our simulation machine will allow for recoveries. Finally, for claims $i = 1, \ldots, n$, we are given the claim status process $(I_i^{(j)})_{0 \leq j \leq 11}$ determining whether claim i is open or closed at the end of each accounting year. More precisely, if $I_i^{(j)} = 1$, claim i is open at the end of accounting year $\texttt{AY}_i + j$, and if $I_i^{(j)} = 0$, claim i is closed at the end of that accounting year. Our simulation machine also allows for re-opening of claims, which is quite common in our real data. More description of the data is given in Appendix A.

2 Design of the simulation machine using neural networks

In this section we describe the architecture of our individual claims history simulation machine. It consists of 8 modeling steps: (1) reporting delay T simulation; (2) payment indicator Z simulation; (3) number of payments K simulation; (4) total claim size Y simulation; (5) number of recovery payments K^- simulation; (6) recovery size Y^- simulation; (7) cash flow $(C_i^{(j)})_{0 \leq j \leq 11}$ simulation and (8) claim status $(I_i^{(j)})_{0 \leq j \leq 11}$ simulation. After a global overview of the architecture of our simulation machine, each single step is described in more detail.

To start with, we define the initial feature space $\mathcal{X}_1$ consisting of the six feature components:

$$\mathcal{X}_1 \;=\; \{(\texttt{LoB}, \texttt{cc}, \texttt{AY}, \texttt{AQ}, \texttt{age}, \texttt{inj\_part})\}. \tag{2.1}$$

Observe that we drop the claims number ClNr because it does not have explanatory power. Apart from these six feature values, the only other model-dependent input parameters of our simulation machine are the standard deviations for the total individual claim sizes and the total individual recoveries, see Sections 2.4 and 2.6 below. During the simulation procedure, not all of the subsequent steps (1)-(8) may be necessary, e.g. if we do not have any payments, then there is no need to simulate the claim size or the cash flow pattern. We briefly describe the 8 modeling steps (1)-(8).

(1) In the first step, we use the initial feature space $\mathcal{X}_1$ to model the reporting delay T indicating the annualized difference between the reporting year and the accident year.

(2) For the second step, we extend the initial feature space $\mathcal{X}_1$ by including the additional information of the reporting delay T, i.e. we set

$$\mathcal{X}_2 \;=\; \{(\texttt{LoB}, \texttt{cc}, \texttt{AY}, \texttt{AQ}, \texttt{age}, \texttt{inj\_part}, T)\}. \tag{2.2}$$

We use $\mathcal{X}_2$ to model the payment indicator Z determining whether we have a payment or not.

(3) For the third step, we set $\mathcal{X}_3 = \mathcal{X}_2$ and model the number of (yearly) payments K.

(4) In the fourth step, we extend the feature space $\mathcal{X}_3$ by including the additional information of the number of payments K, i.e. we set

$$\mathcal{X}_4 \;=\; \{(\texttt{LoB}, \texttt{cc}, \texttt{AY}, \texttt{AQ}, \texttt{age}, \texttt{inj\_part}, T, K)\}, \tag{2.3}$$

which is used to model the total individual claim size Y.

(5) In the fifth step, we model the number of recovery payments K^-. We therefore work on the extended feature space

$$\mathcal{X}_5 \;=\; \{(\texttt{LoB}, \texttt{cc}, \texttt{AY}, \texttt{AQ}, \texttt{age}, \texttt{inj\_part}, T, K, Y)\}\,. \tag{2.4}$$

(6) In the sixth step, we model the total individual recovery Y^-. Therefore, we set $\mathcal{X}_6 = \mathcal{X}_5$. We understand the total individual claim size Y to be net of recovery Y^-. Thus, the total payment from the insurance company to the insured is $Y + Y^-$, paid in $K - K^-$ yearly payments. The total recovery from the insured to the insurance company is Y^-, paid in K^- yearly payments.

(7) In the seventh step, the task is to generate the cash flows $(C_i^{(j)})_{0\le j\le 11}$. Therefore, we have to split the total gross claim amount $Y+Y^-$ into $K-K^-$ positive payments and the total recovery Y^- into K^- negative payments and distribute these K payments among the 12 development years. For this modeling step, we use different feature spaces $\mathcal{X}_{7a}, \ldots, \mathcal{X}_{7g}$, all being a subset of

$$\mathcal{X}_7 \;=\; \left\{\big(\texttt{LoB}, \texttt{cc}, \texttt{AY}, \texttt{AQ}, \texttt{age}, \texttt{inj\_part}, T, K, Y, K^-, Y^-\big)\right\}, \tag{2.5}$$

see Section 2.7 below for more details.

(8) In the last step, we model the claim status process $(I_i^{(j)})_{0\le j\le 11}$, where we use the feature space

$$\mathcal{X}_8 \;=\; \left\{\Big(\texttt{LoB}, \texttt{AQ}, T, (C^{(j)})_{0\le j\le 11}\Big)\right\}.$$

Each of these 8 modeling steps (1)-(8) consists of one or even multiple feature-response problems, for which we design neural networks. In the end, the full individual claims history simulation machine consists of 35 neural networks. We are going to describe this neural network architecture in more detail next. We remark that some of these networks are rather similar. Therefore, we present the first neural network in full detail, and for the remaining neural networks we focus on the differences to the previous ones.

2.1 Reporting delay modeling

To model the reporting delay, we work with the initial feature space $\mathcal{X}_1$ given in (2.1). Let $n_1 = n = 9'977'298$ be the number of individual claims in our data. We consider the (annualized) reporting delays T_i, for $i = 1, \ldots, n_1$, given by

$$T_i \; = \; \texttt{RY}_i - \texttt{AY}_i \; \in \; \mathcal{T} = \{0, \ldots, 11\},$$

where $\texttt{AY}_i$ is the accident year and $\texttt{RY}_i$ the reporting year of claim i.
The three feature components `LoB`, `cc` and `inj_part` are categorical. For neural network modeling, we need to transform these categorical feature components to continuous ones. This could be done by dummy coding, but we prefer the following version because it leads to less parameters. We replace, for instance, the claims code `cc` by the sample mean of the reporting delay restricted to the corresponding feature label, i.e. for claims code $\texttt{cc} = a$, we set

$$a \; \mapsto \; a^* \; = \; a^*(a) \; = \; \frac{\sum_{i=1}^{n_1} T_i \, \mathbb{1}_{\{\texttt{cc}_i = a\}}}{\sum_{i=1}^{n_1} \mathbb{1}_{\{\texttt{cc}_i = a\}}} \; \in \; \mathbb{R}, \tag{2.6}$$

where $\texttt{cc}_1, \ldots, \texttt{cc}_{n_1}$ are the observed claims codes. By slight abuse of notation, we obtain a $d = 6$ dimensional feature space $\mathcal{X}_1$ where we may assume that all feature components of $\mathcal{X}_1$ are continuous. Such feature pre-processing as in (2.6) will be necessary throughout this section for the components `LoB`, `cc` and `inj_part`: we just replace T_i in (2.6) by the respective response variable. Note that from now on this will be done without any further reference.
The above procedure equips us with the data

$$\mathcal{D}_1 \; = \; \{(\boldsymbol{x}_1, T_1), \ldots, (\boldsymbol{x}_{n_1}, T_{n_1})\},$$

with $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_1} \in \mathcal{X}_1$ being the observed features and $T_1, \ldots, T_{n_1} \in \mathcal{T}$ the observed responses. For an insurance claim with feature $\boldsymbol{x} \in \mathcal{X}_1$, the corresponding reporting delay $T(\boldsymbol{x})$ is modeled by a categorical distribution

$$\mathbb{P}\left[T(\boldsymbol{x}) = t\right] \; = \; \pi_t(\boldsymbol{x}), \qquad \text{for } t \in \mathcal{T}.$$

This requires that we model probability functions of the form

$$\pi_t : \; \mathcal{X}_1 \; \to \; [0, 1], \qquad \boldsymbol{x} \; \mapsto \; \pi_t(\boldsymbol{x}),$$

satisfying normalization $\sum_{t \in \mathcal{T}} \pi_t(\boldsymbol{x}) = 1$, for all $\boldsymbol{x} \in \mathcal{X}_1$. We design a neural network for the modeling of these probability functions and we estimate the corresponding network parameters from the observations $\mathcal{D}_1$.

We choose a classical feed-forward neural network with multiple layers. Each layer consists of several neurons, and weights connect all neurons of a given layer to all neurons of the next layer. Moreover, we use a non-linear activation function to pass the signals from one layer to the next. The first layer, consisting of the components $x_1, \ldots, x_d$ of a feature $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathcal{X}_1$, is called input layer (blue circles in Figure 1). In our case, we have $d = 6$ neurons in this input layer. The last layer is called output layer (red circles in Figure 1) and it contains the categorical probabilities $\pi_0(\boldsymbol{x}), \ldots, \pi_{11}(\boldsymbol{x})$. In between these two layers, we choose two hidden layers having q_1 and q_2 hidden neurons, respectively (black circles in Figure 1 with $q_1 = 11$ and $q_2 = 15$).

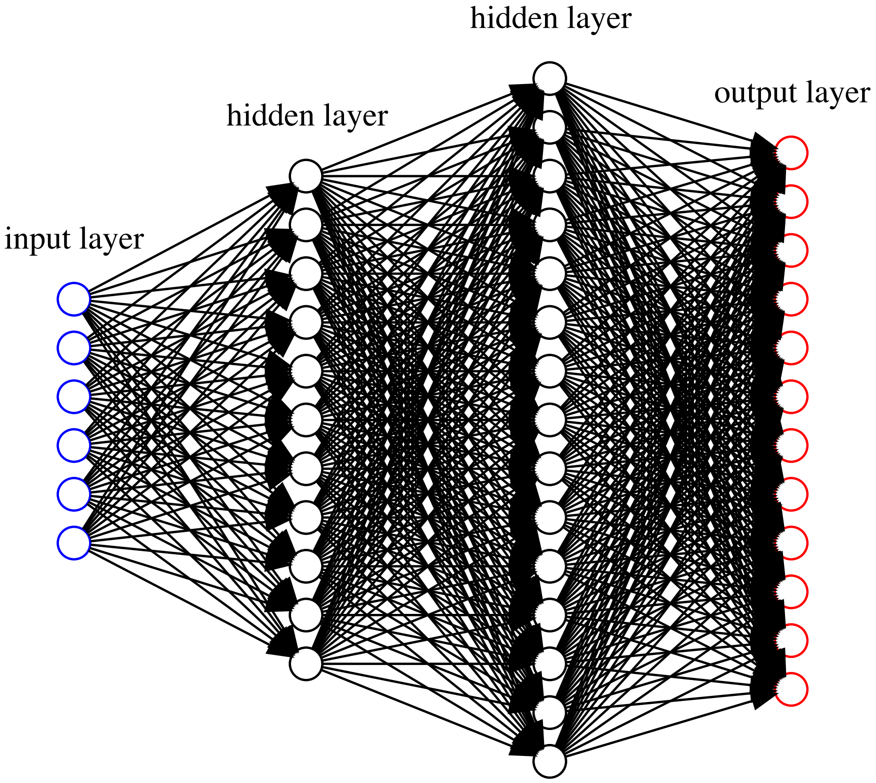

Figure 1: Deep neural network with two hidden layers: the first column (blue circles) illustrates the $d = 6$ dimensional feature vector $\boldsymbol{x}$ (input layer), the second column gives the first hidden layer with $q_1 = 11$ neurons, the third column gives the second hidden layer with $q_2 = 15$ neurons and the fourth column gives the output layer (red circle) with 12 neurons.

More formally, we choose the q_1 hidden neurons $z_1^{(1)}, \ldots, z_{q_1}^{(1)}$ in the first hidden layer as follows

$$z_j^{(1)} \;=\; z_j^{(1)}(\boldsymbol{x}) \;=\; \phi\left(w_{j,0}^{(1)} + \sum_{l=1}^{d} w_{j,l}^{(1)}\, x_l\right), \qquad \text{for all } j = 1, \ldots, q_1,$$

for given weights $(w_{j,l}^{(m)})_{j,l,m}$ and for the hyperbolic tangent activation function

$$\phi(x) \;=\; \tanh(x).$$

The range of $\phi(x)$ is $[-1, 1]$, making it a centered version of the sigmoid activation function. We have $\phi' = 1 - \phi^2$, which is a useful property in the gradient descent method described in Section 3, below.

The activation is then propagated in an analogous fashion to the q_2 hidden neurons $z_1^{(2)}, \ldots, z_{q_2}^{(2)}$ in the second hidden layer, that is, we set

$$z_j^{(2)} \;=\; z_j^{(2)}(\boldsymbol{x}) \;=\; \phi\left(w_{j,0}^{(2)} + \sum_{l=1}^{q_1} w_{j,l}^{(2)}\, z_l^{(1)}(\boldsymbol{x}) \right), \qquad \text{for all } j = 1, \ldots, q_2.$$

For the 12 neurons $\pi_0(\boldsymbol{x}), \ldots, \pi_{11}(\boldsymbol{x})$ in the output layer, we use the multinomial logistic regression assumption

$$\pi_t(\boldsymbol{x}) \;=\; \frac{\exp\left\{\mu_t(\boldsymbol{x})\right\}}{\sum_{s\in\mathcal{T}} \exp\left\{\mu_s(\boldsymbol{x})\right\}}, \qquad \text{for all } t \in \mathcal{T}, \tag{2.7}$$

with regression functions $\boldsymbol{x} \mapsto \mu_t(\boldsymbol{x})$ for all $t \in \mathcal{T}$ given by

$$\mu_t(\boldsymbol{x}) \;=\; \beta_0^{(t)} + \sum_{j=1}^{q_2} \beta_j^{(t)} z_j^{(2)}(\boldsymbol{x}),$$

for given weights $(\beta_j^{(t)})_{j,t}$. We define the network parameter $\boldsymbol{\alpha}$ of all involved parameters by

$$\boldsymbol{\alpha} \;=\; \left(w_{1,0}^{(1)}, \ldots, w_{q_2,q_1}^{(2)}, \beta_0^{(0)}, \ldots, \beta_{q_2}^{(11)} \right)' \;\in\; \mathbb{R}^{q_1(d+1)+q_2(q_1+1)+12(q_2+1)}.$$

The classification model for the tuples $(\boldsymbol{x}, T(\boldsymbol{x}))_{\boldsymbol{x}\in\mathcal{X}_1}$ is now fully defined and there remains the calibration of the network parameter $\boldsymbol{\alpha}$ and the choice of the hyperparameters q_1 and q_2. Assume for the moment that q_1 and q_2 are given. In order to fit $\boldsymbol{\alpha}$ to our data $\mathcal{D}_1$, we aim to minimize a given loss function $\boldsymbol{\alpha} \mapsto \mathcal{L}(\boldsymbol{\alpha})$. Therefore, we assume that $(\boldsymbol{x}_1, T_1), \ldots, (\boldsymbol{x}_{n_1}, T_{n_1})$ are drawn independently from the joint distribution of $(\boldsymbol{x}, T(\boldsymbol{x}))$. The corresponding deviance statistics loss function of the categorical distribution of our data $\mathcal{D}_1$ is then given by

$$\mathcal{L}(\boldsymbol{\alpha}) \;=\; \mathcal{L}_{\mathcal{D}_1}(\boldsymbol{\alpha}) \;=\; -2\log\left(\prod_{i=1}^{n_1} \sum_{t\in\mathcal{T}} \mathbb{1}_{\{T_i=t\}} \pi_t(\boldsymbol{x}_i) \right) \;=\; -2\sum_{i=1}^{n_1} \sum_{t\in\mathcal{T}} \mathbb{1}_{\{T_i=t\}} \log \pi_t(\boldsymbol{x}_i). \tag{2.8}$$

The optimal network parameter $\boldsymbol{\alpha}$ is found by minimizing this deviance statistics loss function. We come back to this problem in Section 3.2.1, below. Since for different hyperparameters q_1 and q_2 we get different network structures, every pair (q_1, q_2) corresponds to a separate model. The choice of appropriate hyperparameters q_1 and q_2 is discussed in Section 3.3, below.

After the calibration of q_1, q_2 and $\boldsymbol{\alpha}$ to our data $\mathcal{D}_1$, we can simulate the reporting delay $T(\boldsymbol{x})$ of a claim with given feature $\boldsymbol{x} \in \mathcal{X}_1$ by using the resulting categorical distribution given by (2.7). This simulated value will then allow us to go to the next modeling step (2), see (2.2).

We close this first part by the following remark: Our choice to work with two hidden layers may seem arbitrary since we could also have chosen more hidden layers or just one of them. From a theoretical point of view, one hidden layer would be sufficient to approximate a vast collection of regression functions to any desired degree of accuracy, provided that we have sufficiently many hidden neurons in that layer, see [2, 4]. However, these models with large-scale numbers of hidden neurons are known to be difficult to calibrate, and it is often more efficient to use fewer neurons but more hidden layers to get an appropriate complexity in the regression function.

2.2 Payment indicator modeling

In our real data, we observe that roughly 29% of all claims can be settled without any payment. For this reason, we model the claim sizes by compound distributions. First, we model a payment indicator Z that determines whether we have a payment or not. Then, conditionally on having a payment, we determine the exact number of payments K. Finally, we model the total individual claim size Y for claims with at least one payment.

In order to model the payment indicator, we work with the $d = 7$ dimensional feature space $\mathcal{X}_2$ introduced in (2.2). Let $n_2 = n_1$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_2} \in \mathcal{X}_2$ be the observed features, where this time the reporting delay T is also included. For all $i = 1, \ldots, n_2$, we define the number of payments by

$$K_i \;=\; \sum_{j=0}^{11} \mathbb{1}_{\left\{C_j^{(i)} \neq 0\right\}}. \tag{2.9}$$

The payment indicator is then given by

$$Z_i \;=\; \mathbb{1}_{\{K_i > 0\}},$$

for all $i = 1, \ldots, n_2$. This provides us with the data

$$\mathcal{D}_2 \;=\; \left\{(\boldsymbol{x}_1, Z_1), \ldots, (\boldsymbol{x}_{n_2}, Z_{n_2})\right\}.$$

For a claim with feature $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathcal{X}_2$, the corresponding payment indicator $Z(\boldsymbol{x})$ is a Bernoulli random variable with

$$\mathbb{P}\left[Z(\boldsymbol{x}) = 1\right] \;=\; \pi(\boldsymbol{x}),$$

for a given (but unknown) probability function

$$\pi: \; \mathcal{X}_2 \;\to\; [0,1], \qquad \boldsymbol{x} \;\mapsto\; \pi(\boldsymbol{x}).$$

Note that this Bernoulli model is a special case of the categorical model of Section 2.1. Therefore, it can be calibrated completely analogously, as described above. In particular, the corresponding deviance statistics loss function to be minimized is given by

$$\mathcal{L}(\boldsymbol{\alpha}) \;=\; \mathcal{L}_{\mathcal{D}_2}(\boldsymbol{\alpha}) \;=\; -2\sum_{i=1}^{n_2} Z_i \log \pi(\boldsymbol{x}_i) + (1 - Z_i)\log(1 - \pi(\boldsymbol{x}_i)), \tag{2.10}$$

where $\boldsymbol{\alpha}$ is the network parameter of the corresponding neural network. From this calibrated model, we simulate the payment indicator $Z(\boldsymbol{x})$, which then allows us to go to the next modeling step. If this indicator is equal to 1, we move to step (3), see Section 2.3; if this indicator is equal to 0, we directly go to step (8), see Section 2.8.

2.3 Number of payments modeling

We use the $d = 7$ dimensional feature space $\mathcal{X}_3 = \mathcal{X}_2$ to model the number of payments, conditioned on the event that the payment indicator Z is equal to one. We define $n_3 \leq n_2$ to be the number of claims with payment indicator equal to one and order the claims appropriately in i such that $Z_i = 1$ for all $i = 1, \ldots, n_3$. Then, we define the number of payments K_i as in (2.9), for all $i = 1, \ldots, n_3$. This gives us the data

$$\mathcal{D}_3 \;=\; \{(\boldsymbol{x}_1, K_1), \ldots, (\boldsymbol{x}_{n_3}, K_{n_3})\}\,.$$

For a claim with feature $\boldsymbol{x} \in \mathcal{X}_3$ and payment indicator $Z = 1$, we could now proceed as in Section 2.1 in order to model the number of payments $K(\boldsymbol{x})$. However, the claims with $K_i = 1$ are so dominant in the data that a good calibration of the categorical model (2.7) becomes difficult. For this reason, we choose a different approach: In a first step, we model the events $\{K(\boldsymbol{x}) = 1\}$ and $\{K(\boldsymbol{x}) > 1\}$, conditioned on $\{Z = 1\}$, and, in a second step, the conditional distribution of $K(\boldsymbol{x})$, given $K(\boldsymbol{x}) > 1$. In particular, in the first step we have a Bernoulli classification problem that is modeled completely analogously to Section 2.2, only replacing the data $\mathcal{D}_2$ by

$$\mathcal{D}_{3a} \;=\; \left\{\left(\boldsymbol{x}_1, \mathbb{1}_{\{K_1=1\}}\right), \ldots, \left(\boldsymbol{x}_{n_3}, \mathbb{1}_{\{K_{n_3}=1\}}\right)\right\}.$$

The case $K(\boldsymbol{x}) > 1$ is then modeled analogously to the categorical case of Section 2.1 with 11 categories and data $\mathcal{D}_{3b} \subset \mathcal{D}_3$ only considering the claims with more than one payment.

The simulation of the random number of payments $K(\boldsymbol{x})$ for a claim with feature $\boldsymbol{x} \in \mathcal{X}_3$, reporting delay T and payment indicator $Z = 1$ needs more care than the corresponding task in Section 2.1: here we have the restriction $T + K \leq 12$. If $T = 11$, then we automatically need to have $K = 1$. For $T < 11$ and if the first neural network leads to $\mathbb{1}_{\{K(\boldsymbol{x})=1\}} = 0$, then the categorical conditional distribution for $K(\boldsymbol{x})$, given $K(\boldsymbol{x}) > 1$, can only take the values $k \in \{2, \ldots, 12 - T\}$. For this reason, instead of using the original conditional probabilities $\pi_2(\boldsymbol{x}), \ldots, \pi_{12}(\boldsymbol{x})$ resulting from the second neural network, we use in that case the modified conditional probabilities $\pi_k^*(\boldsymbol{x})$ given by

$$\pi_k^*(\boldsymbol{x}) \;=\; \frac{\pi_k(\boldsymbol{x})}{\sum_{l=2}^{12-T} \pi_l(\boldsymbol{x})}, \tag{2.11}$$

for all $k \in \{2, \ldots, 12 - T\}$, to simulate the number of payments $K(\boldsymbol{x})$ from $\{2, \ldots, 12 - T\}$.

2.4 Total individual claim size modeling

For the modeling of the total individual claim size, we add the number of payments K to the previous feature space and work with $\mathcal{X}_4$ given in (2.3). Let $n_4 = n_3$ and consider the same ordering of the claims as in Section 2.3. Then, we define the total individual claim size Y_i of claim i as

$$Y_i \;=\; \sum_{j=0}^{11} C_j^{(i)} \;>\; 0,$$

for all $i = 1, \ldots, n_4$. In particular, the total individual claim size Y_i is always to be understood net of recoveries. This leads us to the data

$$\mathcal{D}_4 \;=\; \{(\boldsymbol{x}_1, Y_1), \ldots, (\boldsymbol{x}_{n_4}, Y_{n_4})\}\,.$$

For a claim with feature $\boldsymbol{x} \in \mathcal{X}_4$ and payment indicator $Z = 1$, we model the total individual claim size $Y(\boldsymbol{x})$ with a log-normal distribution. We therefore choose a regression function

$$\mu : \mathcal{X}_4 \to \mathbb{R}, \qquad \boldsymbol{x} \mapsto \mu(\boldsymbol{x}) = \beta_0 + \sum_{j=1}^{q_2} \beta_j z_j^{(2)}(\boldsymbol{x}), \tag{2.12}$$

for a neural network with two hidden layers and network parameter $\boldsymbol{\alpha}$ given by

$$\boldsymbol{\alpha} = \left(w_{1,0}^{(1)}, \ldots, w_{q_2,q_1}^{(2)}, \beta_0, \ldots, \beta_{q_2}\right)' \in \mathbb{R}^{q_1(d+1)+q_2(q_1+1)+(q_2+1)}.$$

The regression function (2.12) is used to model the mean parameter of the total individual claim sizes, i.e. we make the model assumption

$$Y(\boldsymbol{x}) \,|\, Z = 1 \sim \mathrm{LN}\left(\mu(\boldsymbol{x}), \sigma_+^2\right), \tag{2.13}$$

for given variance parameter $\sigma_+^2 > 0$. This choice implies

$$\mathbb{E}\left[\log Y(\boldsymbol{x}) \,|\, Z = 1\right] = \mu(\boldsymbol{x}) \qquad \text{and} \qquad \mathrm{Var}\left(\log Y(\boldsymbol{x}) \,|\, Z = 1\right) = \sigma_+^2.$$

The density of $\log Y(\boldsymbol{x}) \,|\, Z = 1$ then motivates the choice of the square loss function (deviance statistics loss function)

$$\mathcal{L}(\boldsymbol{\alpha}) = \mathcal{L}_{\mathcal{D}_4}(\boldsymbol{\alpha}) = \sum_{i=1}^{n_4} \left(\log Y_i^+ - \mu(\boldsymbol{x}_i)\right)^2, \tag{2.14}$$

where we assume that the data $\mathcal{D}_4$ has been drawn independently from the joint distribution of $(\boldsymbol{x}, Y(\boldsymbol{x}) \,|\, Z = 1)$. Note that (2.14) does not depend on σ_+^2. The optimal model for the total individual claim size is then found by minimizing the loss function (2.14).

This calibrated model together with the input parameter $\sigma_+^2 > 0$ is then used to simulate the total individual claim size $Y(\boldsymbol{x})$ from (2.13). Note that the expected claim amount is increasing in σ_+^2, as we have

$$\mathbb{E}\left[Y(\boldsymbol{x}) \,|\, Z = 1\right] = \exp\left\{\mu(\boldsymbol{x}) + \frac{\sigma_+^2}{2}\right\}.$$

2.5 Number of recovery payments modeling

Now we focus on the recovery modeling. We use the $d = 9$ dimensional feature space $\mathcal{X}_5$ introduced in (2.4). Furthermore, we only consider claims i with $K_i > 1$, because recoveries may only happen if we have at least one positive payment. We define $n_5 \leq n_4$ to be the number of claims with more than one payment and order the claims appropriately in i such that $K_i > 1$ for all $i = 1, \ldots, n_5$. Then, we define the number of recovery payments K_i^- of claim i as

$$K_i^- = \min\left(\sum_{j=0}^{11} \mathbb{1}_{\left\{C_j^{(i)} < 0\right\}}, 2\right), \tag{2.15}$$

for all $i = 1, \ldots, n_5$. In particular, for all observed claims with more than two recovery payments, we set $K_i^- = 2$. This reduces combinatorial complexity in simulations (without much loss of accuracy) and provides us with the data

$$\mathcal{D}_5 = \left\{(\boldsymbol{x}_1, K_1^-), \ldots, (\boldsymbol{x}_{n_5}, K_{n_5}^-)\right\}.$$

For a claim with feature $\boldsymbol{x} \in \mathcal{X}_5$ and $K > 1$ payments, the corresponding number of recovery payments $K^-(\boldsymbol{x})$, conditioned on the event $\{K > 1\}$, is a categorical random variable taking values in $\{0, 1, 2\}$, i.e. we are in the same setup as in Section 2.1 with only three categorical classes. Thus, the calibration is done analogously.

This model then allows us to simulate the number of recovery payments $K^-(\boldsymbol{x})$. Note that also this simulation step needs additional care: if $K = 2$, then we can have at most one recovery payment and we scale the original probabilities $\pi_0(\boldsymbol{x}), \pi_1(\boldsymbol{x}), \pi_2(\boldsymbol{x})$ in that case to

$$\pi_0^*(\boldsymbol{x}) \;=\; \frac{\pi_0(\boldsymbol{x})}{\pi_0(\boldsymbol{x}) + \pi_1(\boldsymbol{x})} \qquad \text{and} \qquad \pi_1^*(\boldsymbol{x}) \;=\; \frac{\pi_1(\boldsymbol{x})}{\pi_0(\boldsymbol{x}) + \pi_1(\boldsymbol{x})}.$$

2.6 Total individual recovery size modeling

The modeling of the total individual recovery size is based on the feature space $\mathcal{X}_6 = \mathcal{X}_5$, given in (2.4), and we restrict to claims with $K_i^- > 0$. The number of these claims is denoted by $n_6 \leq n_5$. Appropriate ordering provides us with the total individual recovery Y_i^- of claim i as

$$Y_i^- \;=\; -\sum_{j=0}^{11} C_j^{(i)} \, \mathbb{1}_{\left\{C_j^{(i)} < 0\right\}},$$

for all $i = 1, \ldots, n_6$. This gives us the data

$$\mathcal{D}_6 \;=\; \left\{ (\boldsymbol{x}_1, Y_1^-), \ldots, (\boldsymbol{x}_{n_6}, Y_{n_6}^-) \right\}.$$

The remaining part is completely analogous to Section 2.4, we only need to replace the standard deviation parameter σ^+ by a given $\sigma^- > 0$.

2.7 Cash flow pattern modeling

The modeling of the cash flows is more involved, and we need to distinguish different cases. This distinction is done according to the total number of payments $K = 1, \ldots, 12$, the number of positive payments $K^+ = K - K^- = 1, \ldots, 12$ as well as the number of recovery payments $K^- = 0, 1, 2$.

2.7.1 Cash flow for single payments $K = 1$

The simplest case is the one of having exactly one payment $K = K^+ = 1$. In this case, we consider the payment delay after the reporting date. We define $n_{7a} \leq n_3$ to be the number of claims with exactly one payment and order the claims appropriately in i such that $K_i = 1$ for all $i = 1, \ldots, n_{7a}$. Then, we define the payment delay S_i of claim i as

$$S_i \;=\; \sum_{j=0}^{11} j \, \mathbb{1}_{\left\{C_j^{(i)} > 0\right\}} - T_i \;\geq\; 0,$$

for all $i = 1, \ldots, n_{7a}$. In other words, we simply subtract the reporting year from the year in which the unique payment occurs. This provides us with the data

$$\mathcal{D}_{7a} \;=\; \left\{ (\boldsymbol{x}_1, S_1), \ldots, (\boldsymbol{x}_{n_{7a}}, S_{n_{7a}}) \right\},$$

with $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_{7a}} \in \mathcal{X}_{7a}$ being the observed features, where we use

$$\mathcal{X}_{7a} \;=\; \{(\texttt{LoB}, \texttt{cc}, \texttt{AY}, \texttt{AQ}, \texttt{age}, \texttt{inj\_part}, T, Y)\} \tag{2.16}$$

as $d = 8$ dimensional feature space. For a claim with feature $\boldsymbol{x} \in \mathcal{X}_{7a}$ and $K = 1$ payment, the corresponding payment delay $S(\boldsymbol{x})$ is a categorical random variable assuming values in $\{0, \ldots, 11\}$. Similarly as for the number of payments, the claims with $S_i = 0$ are rather dominant. Therefore, we apply the same two-step modeling approach as in Section 2.3.
This calibrated model then allows us to simulate the payment delay $S(\boldsymbol{x})$. For given reporting delay T, we have the restriction $T + S \leq 11$, which is treated in the same way as in (2.11). Finally, the cash flow for a claim with $K = 1$ is given by $(C^{(j)}(\boldsymbol{x}))_{0 \leq j \leq 11}$ with

$$C^{(j)}(\boldsymbol{x}) \;=\; \begin{cases} Y(\boldsymbol{x}), & \text{if } j = T + S(\boldsymbol{x}), \\ 0, & \text{else.} \end{cases}$$

2.7.2 Cash flow for two payments $K = 2$

Now we consider claims with exactly two payments. Here we distinguish further between the two cases: (1) both payments are positive, and (2) one payment is positive and the other one negative.

(a) Two positive payments

We first consider the case where both payments are positive, i.e. $K = K^+ = 2$ and $K^- = 0$. In this case, we have to model the time points of the two payments as well as the split of the total claim amount to the two payments. For both models, we use the $d = 8$ dimensional feature space $\mathcal{X}_{7b} = \mathcal{X}_{7a}$, see (2.16). We define $n_{7b} \leq n_3$ to be the number of claims with exactly two positive payments and no recovery and order them appropriately in i such that $K_i = 2$ and $K_i^- = 0$ for all $i = 1, \ldots, n_{7b}$. The time points $R_i^{(1)}$ and $R_i^{(2)}$ of the two payments are given by

$$R_i^{(1)} \;=\; \min\left\{0 \leq j \leq 11 \,\middle|\, C_j^{(i)} \neq 0\right\} \qquad \text{and} \qquad R_i^{(2)} \;=\; \max\left\{0 \leq j \leq 11 \,\middle|\, C_j^{(i)} \neq 0\right\},$$

for all $i = 1, \ldots, n_{7b}$. Then, we modify the two-dimensional vector $(R_i^{(1)}, R_i^{(2)})$ to a one-dimensional categorical variable R_i by setting

$$R_i \;=\; \begin{cases} R_i^{(2)}, & \text{if } R_i^{(1)} = 0, \\ R_i^{(2)} + \sum_{k=12-R_i^{(1)}}^{11} k, & \text{else,} \end{cases} \tag{2.17}$$

for all $i = 1, \ldots, n_{7b}$. This leads us to the data

$$\mathcal{D}_{7b} \;=\; \{(\boldsymbol{x}_1, R_1), \ldots, (\boldsymbol{x}_{n_{7b}}, R_{n_{7b}})\}\,.$$

Note that R_i is categorical with $\binom{12}{2} = 66$ possible values. That is, we are in the same setup as in Section 2.1 but with 66 different classes. Once again, the calibration is done in an analogous fashion as above.

Next, we model the split of the total claim amount $Y_i = C^{(i)}_{R_i^{(1)}} + C^{(i)}_{R_i^{(2)}}$ for claims with $K = K^+ = 2$. Let $n_{7c} = n_{7b}$, $\mathcal{X}_{7c} = \mathcal{X}_{7b}$, see (2.16), and define the proportion of the first payment by

$$P_i \;=\; \frac{C^{(i)}_{R_i^{(1)}}}{Y_i},$$

for all $i \in 1, \ldots, n_{7c}$. This gives us the data

$$\mathcal{D}_{7c} \;=\; \left\{(\boldsymbol{x}_1, P_1), \ldots, (\boldsymbol{x}_{n_{7c}}, P_{n_{7c}})\right\}.$$

For a claim with feature $\boldsymbol{x} \in \mathcal{X}_{7c}$ and $K = K^+ = 2$, the corresponding proportion of its total individual claim size Y that is paid in the first payment is for simplicity modeled by a deterministic function $P(\boldsymbol{x})$. Note that one could easily randomize $P(\boldsymbol{x})$ using a Dirichlet distribution. However, at this modeling stage, the resulting differences would be of smaller magnitude. Hence, we directly fit the proportion function

$$P: \; \mathcal{X}_{7c} \;\to\; [0,1], \qquad \boldsymbol{x} \;\mapsto\; P(\boldsymbol{x}).$$

Similarly to the calibration in Section 2.2, we assume a regression function $\mu : \mathcal{X}_{7c} \to \mathbb{R}$ of type (2.12) for a neural network with two hidden layers. Then, for the output layer, we use

$$P(\boldsymbol{x}) \;=\; \frac{1}{1 + \exp\left\{-\mu(\boldsymbol{x})\right\}} \tag{2.18}$$

and as loss function the cross entropy function, see also (2.10),

$$\mathcal{L}(\boldsymbol{\alpha}) \;=\; \mathcal{L}_{\mathcal{D}_{7c}}(\boldsymbol{\alpha}) \;=\; -2\sum_{i=1}^{n_{7c}} P_i \log P(\boldsymbol{x}_i) + (1 - P_i)\log(1 - P(\boldsymbol{x}_i)), \tag{2.19}$$

where $\boldsymbol{\alpha}$ is the network parameter containing all the weights of the neural network.
From this model, we can then simulate the cash flow for a claim with $K = K^+ = 2$. First, we simulate $R(\boldsymbol{x})$. If $R(\boldsymbol{x}) \in \{1, \ldots, 11\}$, we have $R^{(1)}(\boldsymbol{x}) = 0$ and $R^{(2)}(\boldsymbol{x}) = R(\boldsymbol{x})$. If $R(\boldsymbol{x}) > 11$, we set

$$R^{(1)}(\boldsymbol{x}) \;=\; \max\left\{1 \le k \le 10 \;\middle|\; R(\boldsymbol{x}) > \sum_{u=12-k}^{11} u\right\} \quad \text{and} \quad R^{(2)}(\boldsymbol{x}) \;=\; R(\boldsymbol{x}) - \sum_{k=12-R^{(1)}(\boldsymbol{x})}^{11} k.$$

The cash flow is given by $(C^{(j)}(\boldsymbol{x}))_{0 \le j \le 11}$ with

$$C^{(j)}(\boldsymbol{x}) \;=\; \begin{cases} P(\boldsymbol{x})\,Y, & \text{if } j = R^{(1)}(\boldsymbol{x}), \\ (1 - P(\boldsymbol{x}))\,Y, & \text{if } j = R^{(2)}(\boldsymbol{x}), \\ 0, & \text{else.} \end{cases}$$

(b) One positive payment, one recovery payment

Now we focus on the case where we have $K^+ = 1$ positive and $K^- = 1$ negative payment. Here we only have to model the time points of the two payments, since we know the total individual claim size as well as the total individual recovery and we assume that the positive payment precedes the recovery payment. The modeling of the time points of the two payments is done as above, except that this time we use the $d = 9$ dimensional feature space

$$\mathcal{X}_{7d} = \left\{ \left(\texttt{LoB}, \texttt{cc}, \texttt{AY}, \texttt{AQ}, \texttt{age}, \texttt{inj\_part}, T, Y, Y^- \right) \right\},$$

where we include the information of the total individual recovery. Moreover, we define $n_{7d} \leq n_3$ to be the number of claims with exactly one positive payment and one recovery payment and order the claims appropriately in i such that $K_i = 2$ and $K_i^- = 1$ for all $i = 1, \ldots, n_{7d}$. This provides us with the data

$$\mathcal{D}_{7d} = \{(\boldsymbol{x}_1, R_1), \ldots, (\boldsymbol{x}_{n_{7d}}, R_{n_{7d}})\},$$

with R_i defined as in (2.17). The rest is done as above. We obtain the cash flow $(C^{(j)}(\boldsymbol{x}))_{0 \leq j \leq 11}$ with

$$C^{(j)}(\boldsymbol{x}) = \begin{cases} Y + Y^-, & \text{if } j = R^{(1)}(\boldsymbol{x}), \\ -Y^-, & \text{if } j = R^{(2)}(\boldsymbol{x}), \\ 0, & \text{else.} \end{cases}$$

Remark that we again have combinatorial complexity of $\binom{12}{2} = 66$ for the time points of the two payments. Since data is sparse, for this calibration we restrict to the 35 most frequent distribution patterns. More details on this restriction are provided in the next section.

2.7.3 Cash flow for more than two payments $K = 3, \ldots, 12$

On the one hand, the models for the cash flows in the case of more than two payments depend on the exact number of payments K. On the other hand, they also depend on the respective numbers of positive payments K^+ and negative payments K^-. If we have zero or one recovery payment ($K^- = 0, 1$), then we need to model (a) the time points where the K payments occur and (b) the proportions of the total gross claim amount $Y + Y^-$ paid in the K^+ positive payments. If $K^- = 0$, then there are no recovery payments and, thus, $Y^- = 0$. If $K^- = 1$, the recovery payment is always set at the end. In the case of $K^- = 2$ recovery payments, in addition to (a) and (b), we use another neural network to model (c) the proportions of the total individual recovery Y^- paid in the two recovery payments. The time point of the first recovery payment is for simplicity assumed to be uniformly distributed on the set of time points of the 2nd up to the $(K-1)$-st payment. The second recovery payment is always set at the end. The time point of the first payment is excluded for recovery in our model since we first require a positive payment before a recovery is possible. The three neural networks considered in this modeling part are outlined below in (a)-(c). Afterwards, we can model the cash flows for claims with $K = 3, \ldots, 12$ payments, see item (d) below.

(a) Distribution of the K payments

If we have $K = 12$ payments, then the distribution of these payments to the 12 development years is trivial, as we have a payment in every development year. Since the model is pretty much the same in all other cases $K \in \{3, \ldots, 11\}$, we present here the case $K = 6$ as illustration.

For the modeling of the distribution of the payments to the development years, we slightly simplify our feature space by dropping the categorical feature components `cc` and `inj_part`. Moreover, we simplify the feature `LoB` with its four categorical classes: since the lines of business 1 and 4 as well as the lines of business 2 and 3 behave very similarly w.r.t. the cash flow patterns, we merge these lines of business in order to get more volume (and less complexity). We denote this simplified lines of business by `LoB`*. Thus, we work with the $d = 8$ dimensional feature space

$$\mathcal{X}_{7e} \;=\; \left\{ \left(\texttt{LoB}^*, \texttt{AY}, \texttt{AQ}, \texttt{age}, T, Y, K^-, Y^-\right) \right\}.$$

Let $n_{7e} \leq n_3$ be the number of claims with exactly 6 payments and order the claims appropriately in i such that $K_i = 6$ for all $i = 1, \ldots, n_{7e}$. The time points $R_i^{(1)}, \ldots, R_i^{(6)}$ of the 6 payments are

$$R_i^{(1)} \;=\; \min\left\{ 0 \leq j \leq 11 \;\middle|\; C_j^{(i)} \neq 0 \right\} \qquad \text{and} \qquad R_i^{(k)} \;=\; \min\left\{ R_i^{(k-1)} < j \leq 11 \;\middle|\; C_j^{(i)} \neq 0 \right\},$$

for all $k = 2, \ldots, 6$ and $i = 1, \ldots, n_{7e}$. Then, we use the following binary representation

$$R_i \;=\; \sum_{k=1}^{6} 2^{R_i^{(k)}},$$

for all $i = 1, \ldots, n_{7e}$, for the time points of the 6 payments. This leads to the data

$$\mathcal{D}_{7e} \;=\; \left\{ (\boldsymbol{x}_1, R_1), \ldots, (\boldsymbol{x}_{n_{7e}}, R_{n_{7e}}) \right\},$$

where $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_{7e}} \in \mathcal{X}_{7e}$ and $R_1, \ldots, R_{n_{7e}} \in \mathcal{A}$ for some set $\mathcal{A} \subset \mathbb{N}$. Since there are

$$\binom{12}{6} \;=\; 924$$

possibilities to distribute the $K = 6$ payments to the 12 development years, we have $|\mathcal{A}| = 924$ distribution patterns. To reduce complexity (in view of sparse data), we decided to only allow for the most frequently observed distributions of the payments to the development years. For $K = 6$, we work with 21 different patterns, which cover 70% of all claims with $K = 6$. We denote the set containing these 21 patterns by $\widetilde{\mathcal{A}}$. See Table 1 for an overview, for each $K = 3, \ldots, 10$, of the number of possible different patterns, the number of allowed different patterns and the percentage of all claims covered with this choice of allowed distribution patterns.

number of payments K	3	4	5	6	7	8	9	10
number of possible patterns $\lvert\mathcal{A}\rvert$	220	495	792	924	792	495	220	66
number of allowed patterns $\lvert\widetilde{\mathcal{A}}\rvert$	15	18	17	21	17	20	26	24
percentage of claims covered	91%	83%	73%	70%	62%	61%	64%	76%

Table 1: Number of possible and allowed distribution patterns for $K = 3, \ldots, 10$ payments.

Note that for $K = 11$, we allowed for all the 12 possible distribution patterns. Going back to the case $K = 6$, we denote by $\widetilde{n}_{7e} \leq n_{7e}$ the number of claims with exactly $K = 6$ payments and with a distribution of these 6 payments to the 12 development years contained in the set $\widetilde{\mathcal{A}}$. Then, we modify the data $\mathcal{D}_{7e}$ accordingly to $\widetilde{\mathcal{D}}_{7e}$ by only considering the relevant observations in $\widetilde{\mathcal{A}}$. This provides us with a classification problem with $|\widetilde{\mathcal{A}}| = 21$ classes. This classification problem is treated analogously to the one in Section 2.1.

(b) Proportions of the $K^+ = K - K^-$ positive payments

If the number of positive payments K^+ is equal to one, then the amount paid in this unique positive payment is given by the total gross claim amount $Y + Y^-$. That is, we do not need to model the proportions of the positive payments. Since the model is basically the same in all other cases $K^+ \in \{2, \ldots, 12\}$, we present here the case $K^+ = 6$ as illustration.

As in the previous part, we use the $d = 8$ dimensional feature space $\mathcal{X}_{7f} = \mathcal{X}_{7e}$. Let $n_{7f} \leq n_3$ be the number of claims with exactly 6 positive payments and order the claims appropriately in i such that $K_i^+ = K_i - K_i^- = 6$ for all $i = 1, \ldots, n_{7f}$. Define $R_i^{+(1)}, \ldots, R_i^{+(6)}$ to be the time points of the 6 positive payments, i.e.

$$R_i^{+(1)} = \min\left\{0 \leq j \leq 11 \,\middle|\, C_j^{(i)} > 0\right\} \qquad \text{and} \qquad R_i^{+(k)} = \min\left\{R_i^{+(k-1)} < j \leq 11 \,\middle|\, C_j^{(i)} > 0\right\},$$

for all $k = 2, \ldots, 6$ and $i = 1, \ldots, n_{7f}$. Then, we can split the total gross claim amount $Y_i + Y_i^-$ to the annual payments $C^{(i)}_{R_i^{+(1)}}, \ldots, C^{(i)}_{R_i^{+(6)}}$, and define the proportion $P_i^{(k)}$ paid in the k-th positive annual payment as

$$P_i^{(k)} = \frac{C^{(i)}_{R_i^{+(k)}}}{Y_i + Y_i^-},$$

for all $k = 1, \ldots, 6$ and $i = 1, \ldots, n_{7f}$. This equips us with the data

$$\mathcal{D}_{7f} = \left\{\left(\boldsymbol{x}_1, P_1^{(1)}, \ldots, P_1^{(6)}\right), \ldots, \left(\boldsymbol{x}_{n_{7f}}, P_{n_{7f}}^{(1)}, \ldots, P_{n_{7f}}^{(6)}\right)\right\}.$$

For a claim with feature $\boldsymbol{x} \in \mathcal{X}_{7f}$ and $K^+ = K - K^- = 6$ positive payments, the corresponding proportions $P^{(1)}(\boldsymbol{x}), \ldots, P^{(6)}(\boldsymbol{x})$ of the total gross claim amount $Y + Y^-$ that are paid in the individual payments are for simplicity assumed to be deterministic. Note that we could randomize the proportions by simulating from a Dirichlet distribution, but, as in Section 2.7.2, we refrain from doing so. Hence, we consider the proportion functions

$$P^{(k)} : \mathcal{X}_{7f} \to [0,1], \qquad \boldsymbol{x} \mapsto P^{(k)}(\boldsymbol{x}),$$

for all $k = 1, \ldots, 6$, with normalization $\sum_{k=1}^{6} P^{(k)}(\boldsymbol{x}) = 1$, for all $\boldsymbol{x} \in \mathcal{X}_{7f}$. We use the same model assumptions as in (2.7) by setting for $k = 1, \ldots, 6$

$$P^{(k)}(\boldsymbol{x}) = \frac{\exp\{\mu_k(\boldsymbol{x})\}}{\sum_{l=1}^{6} \exp\{\mu_l(\boldsymbol{x})\}}, \tag{2.20}$$

for appropriate regression functions $\mu_k : \mathcal{X}_{7f} \to \mathbb{R}$ resulting as output layer from a neural network with two hidden layers. As in (2.19), we consider the cross entropy loss function

$$\mathcal{L}(\boldsymbol{\alpha}) = \mathcal{L}_{\mathcal{D}_{7f}}(\boldsymbol{\alpha}) = -2 \sum_{i=1}^{n_{7f}} \sum_{k=1}^{6} P_i^{(k)} \log P^{(k)}(\boldsymbol{x}_i),$$

where $\boldsymbol{\alpha}$ is the corresponding network parameter. This model is calibrated as described in Section 2.1. Remark that if $K^+ = 2$, the model (2.20) simplifies to the binomial case, see (2.18).

(c) Proportions of the recovery payments if $K^- = 2$

In the case of $K^- = 2$ recovery payments, we need to model the proportion of the total individual recovery Y^- paid in the first recovery payment. For this, we work with the $d = 10$ dimensional feature space

$$\mathcal{X}_{7g} \;=\; \left\{ \left(\texttt{LoB}, \texttt{cc}, \texttt{AY}, \texttt{AQ}, \texttt{age}, \texttt{inj\_part}, T, K, Y, Y^- \right) \right\}.$$

We denote by $n_{7g} \leq n_6$ the number of claims with exactly two recovery payments and order the claims appropriately in i such that $K_i^- = 2$ for all $i = 1, \ldots, n_{7g}$. Recall that we set $K_i^- = 2$ for all claims i with two or more recovery payments, see (2.15). Moreover, we add all the amounts of the recovery payments done after the second recovery payment to the second one. Let

$$R_i^- \;=\; \min \left\{ 0 \leq j \leq 11 \,\middle|\, C_j^{(i)} < 0 \right\}$$

denote the time point of the first recovery payment, for $i \in \{1, \ldots, n_{7g}\}$. Then, the proportion P_i^- of the total individual recovery that is paid in the first recovery payment is given by

$$P_i^- \;=\; \frac{-C_{R_i^-}^{(i)}}{Y_i^-},$$

for all $i = 1, \ldots, n_{7g}$. This provides us with the data

$$\mathcal{D}_{7g} \;=\; \left\{ \Big(\boldsymbol{x}_1, P_1^-), \ldots, (\boldsymbol{x}_{n_{7g}}, P_{n_{7g}}^- \Big) \right\}.$$

The remaining modeling part is then done completely analogously to the second part of the two positive payments case (a) in Section 2.7.2.

(d) Cash flow modeling

Finally, using the three neural network models outlined above, we can simulate the cash flow for a claim with more than two payments and with feature $\boldsymbol{x} \in \mathcal{X}_7$, see (2.5). We illustrate the case $K = 6$. Note that we only allow for cash flow patterns in $\widetilde{\mathcal{A}}$ that are compatible with the reporting delay T. We start by describing the case $T = 0$. In this case, there is no difficulty and we directly simulate the cash flow pattern $R(\boldsymbol{x}) \in \widetilde{\mathcal{A}}$. This provides us 6 payments in the time points

$$\begin{aligned} R^{(6)}(\boldsymbol{x}) \;&=\; \max \left\{ 0 \leq j \leq 11 \,\middle|\, 2^{j+1} \leq R(\boldsymbol{x}) \right\} \qquad \text{and} \\ R^{(k)}(\boldsymbol{x}) \;&=\; \max \left\{ 0 \leq j < R^{(k+1)}(\boldsymbol{x}) \,\middle|\, 2^{j+1} \leq R(\boldsymbol{x}) - \sum_{l=k+1}^{6} R^{(l)}(\boldsymbol{x}) \right\}, \quad \text{for } k = 1, \ldots, 5. \end{aligned}$$

For reporting delay $T = 1$, the set $\widetilde{\mathcal{A}}$ of potential cash flow patterns becomes smaller because some of them have to be dropped to remain compatible with $T = 1$. For this reason, we simulate with probability $\frac{1}{2}$ a pattern from $\widetilde{\mathcal{A}}$, and with probability $\frac{1}{2}$ the 6 time points $R^{(1)}(\boldsymbol{x}), \ldots, R^{(6)}(\boldsymbol{x})$ are drawn in a uniform manner from the remaining possible time points in

$\{T = 1, \ldots, 11\}$. For $T > 1$, the potential subset of patterns in $\widetilde{\mathcal{A}}$ becomes (almost) empty. For this reason, we simply simulate uniformly from the compatible configurations in $\{T, \ldots, 11\}$. Having the 6 time points for the payments, we distinguish the 3 different cases $K^- \in \{0, 1, 2\}$:
Case $K^- = 0$: we calculate the proportions $P^{(1)}(\boldsymbol{x}), \ldots, P^{(6)}(\boldsymbol{x})$ according to point (b) above and we receive the cash flow $(C^{(j)}(\boldsymbol{x}))_{0 \le j \le 11}$ with

$$C^{(j)}(\boldsymbol{x}) \;=\; \begin{cases} P^{(l)}(\boldsymbol{x})\, Y, & \text{if } j = R^{(l)}(\boldsymbol{x}) \text{ for some } 1 \le l \le 6, \\ 0, & \text{else.} \end{cases}$$

Case $K^- = 1$: we have 5 positive payments with proportions $P^{(1)}(\boldsymbol{x}), \ldots, P^{(5)}(\boldsymbol{x})$ modeled according to point (b) above. This provides the cash flow $(C^{(j)}(\boldsymbol{x}))_{0 \le j \le 11}$ with

$$C^{(j)}(\boldsymbol{x}) \;=\; \begin{cases} P^{(l)}(\boldsymbol{x})\ (Y + Y^-)\,, & \text{if } j = R^{(l)}(\boldsymbol{x}) \text{ for some } 1 \le l \le 5, \\ -Y^-, & \text{if } j = R^{(6)}(\boldsymbol{x}), \\ 0, & \text{else.} \end{cases}$$

Case $K^- = 2$*:* we have 4 positive payments with proportions $P^{(1)}(\boldsymbol{x}), \ldots, P^{(4)}(\boldsymbol{x})$ according to point (b) above and 2 negative payments with proportions $P^-(\boldsymbol{x})$ and $1 - P^-(\boldsymbol{x})$ according to point (c) above. The time point of the first recovery $P^-(\boldsymbol{x})$ is simulated uniformly from the set of time points

$$\left\{R^{(2)}(\boldsymbol{x}), \ldots, R^{(5)}(\boldsymbol{x})\right\}.$$

The time point $R^{(1)}(\boldsymbol{x})$ is reserved for a first positive payment and the time point $R^{(6)}(\boldsymbol{x})$ for the second recovery payment. Summarizing, we get the cash flow $(C^{(j)}(\boldsymbol{x}))_{0 \le j \le 11}$ with

$$C^{(j)}(\boldsymbol{x}) \;=\; \begin{cases} P^{(l)}(\boldsymbol{x})\ (Y + Y^-)\,, & \text{if } j = R^{(l)}(\boldsymbol{x}) \text{ for some } 1 \le l \le 5 \text{ and } j \ne R^-(\boldsymbol{x}), \\ -P^-(\boldsymbol{x})\, Y^-, & \text{if } j = R^-(\boldsymbol{x}), \\ -(1 - P^-(\boldsymbol{x}))Y^-, & \text{if } j = R^{(6)}(\boldsymbol{x}), \\ 0, & \text{else.} \end{cases}$$

Of course, if $K = 3$ and $K^- = 2$, we do not need to simulate the proportions of the positive payments, as there is only one positive payment, which occurs in the beginning. Similarly, if $K = 12$, we do not need to simulate the time points of the payments, since there is a payment in every development year.

2.8 Claim status modeling

Finally, we design the model for the claim status process which indicates whether a claim is open or closed at the end of each accounting year. This process modeling will also allow for re-opening. Similarly to the payments, we do not model the status of a claim or its changes within an accounting year, but only focus on its status at the end of each accounting year. The modeling procedure of the claim status uses two neural networks, which are described below.
We remark that the closing date information was of lower quality in our data set compared to all other information. For instance, some of the dates have been modified retrospectively which, of course, destroys the time series aspect. For this reason, we have decided to model this process in a more crude form, however, still capturing predictive power.

2.8.1 Re-opening indicator

We start by modeling the occurrence of a re-opening, i.e. whether a claim gets re-opened after having been closed at an earlier date. We use the $d = 15$ dimensional feature space

$$\mathcal{X}_{8a} \;=\; \left\{ \left(\texttt{LoB}, \texttt{AQ}, T, (\widetilde{C}^{(j)})_{0\le j\le 11} \right) \right\},$$

where we do not consider the exact payment amounts, but the simplified version

$$\widetilde{C}^{(j)} \;=\; \begin{cases} -\frac{1}{2}, & \text{if } C^{(j)} = 0, \\ 0, & \text{if } C^{(j)} \neq 0 \text{ and } C^{(j)} \le 1'000, \\ \frac{1}{2}, & \text{if } C^{(j)} \ge 1'000, \end{cases} \tag{2.21}$$

for all $j = 0, \ldots, 11$. Let $n_{8a} \le n$ denote the number of claims i for which we have the full information $(I_i^{(j)})_{T\le j\le 11}$. For the ease of data processing, we set $I_i^{(j)} = 1$ for all development years before claims reporting T. Then, we can define the re-opening indicator V_i as

$$V_i \;=\; \begin{cases} 1, & \text{if } \sum_{j=1}^{11} \mathbb{1}_{\left\{I_i^{(j)} - I_i^{(j-1)} = 1\right\}} \ge 1, \\ 0, & \text{else}, \end{cases}$$

for all $i = 1, \ldots, n_{8a}$. In particular, if $V_i = 1$, then claim i has at least one re-opening and if $V_i = 0$, then claim i has not been re-opened. This leads us to the data

$$\mathcal{D}_{8a} \;=\; \{(\boldsymbol{x}_1, V_1), \ldots, (\boldsymbol{x}_{n_{8a}}, V_{n_{8a}})\},$$

where $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_{8a}} \in \mathcal{X}_{8a}$. For a given feature $\boldsymbol{x} \in \mathcal{X}_{8a}$, the corresponding re-opening indicator $V(\boldsymbol{x})$ is a Bernoulli random variable. Thus, model calibration is done analogously to Section 2.2 with, however, a neural network with only one hidden layer.

2.8.2 Closing delay indicator (claims without a re-opening)

For claims without a re-opening, we model the closing delay indicator determining whether the closing occurs in the same year as the last payment or if the closing occurs later. In case of no payments ($Z_i = 0$), we replace the year of the last payment by the reporting year. We use the same $d = 15$ dimensional feature space as for the re-opening indicator and set $\mathcal{X}_{8b} = \mathcal{X}_{8a}$. Let $n_{8b} \le n_{8a}$ be the number of claims without a re-opening and order them appropriately in i such that $V_i = 0$ for all $i = 1, \ldots, n_{8b}$. Then, we define the closing delay indicator W_i as

$$W_i \;=\; \begin{cases} 1, & \text{if } Z_i = 1 \text{ and } \max\left\{0 \le j \le 11 \,\middle|\, I_i^{(j)} = 1\right\} \ge \max\left\{0 \le j \le 11 \,\middle|\, C_i^{(j)} \neq 0\right\}, \\ 1, & \text{if } Z_i = 0 \text{ and } \max\left\{0 \le j \le 11 \,\middle|\, I_i^{(j)} = 1\right\} \ge T_i, \\ 0, & \text{else}, \end{cases}$$

for all $i = 1, \ldots, n_{8b}$. Hence, we have $W_i = 1$ if the closing occurs in a later year compared to the year of the last payment (or in a later year compared to the claims reporting year in case there is no payment) and $W_i = 0$ otherwise. This leads us to the data

$$\mathcal{D}_{8b} \;=\; \{(\boldsymbol{x}_1, W_1), \ldots, (\boldsymbol{x}_{n_{8b}}, W_{n_{8b}})\}.$$

For a claim with feature $\boldsymbol{x} \in \mathcal{X}_{8b}$, the corresponding closing delay indicator $W(\boldsymbol{x})$ is again a Bernoulli random variable. Therefore, model calibration is done analogously to Section 2.2. Similarly as for the re-opening indicator, we use a neural network with only one hidden layer.

2.8.3 Simulation of the claim status

Based on the feature space $\mathcal{X}_{8a}$, we first simulate the re-opening indicator $V(\boldsymbol{x})$ leading to the two cases (i) and (ii) described below. Note that before a claim is reported, for ease of data processing, we simply set its status to open (this has no further relevance).
(i) Case $V(\boldsymbol{x}) = 0$ (without re-opening): for the given feature $\boldsymbol{x} \in \mathcal{X}_{8a}$, we calculate the closing delay probability $\pi(\boldsymbol{x}) = \mathbb{P}[W(\boldsymbol{x}) = 1]$ using the neural network of Section 2.8.2. The closing delay $B(\boldsymbol{x})$ is then sampled from a categorical distribution on $\{0, \ldots, 12\}$ with probabilities

$$\begin{aligned} \mathbb{P}[B(\boldsymbol{x}) = 0] &= 1 - \pi(\boldsymbol{x}), \\ \mathbb{P}[B(\boldsymbol{x}) = 1] &= \frac{9}{10}\,\pi(\boldsymbol{x}), \\ \mathbb{P}[B(\boldsymbol{x}) = k] &= \frac{1}{10}\,\frac{1}{11}\,\pi(\boldsymbol{x}), \qquad \text{for } k = 2, \ldots, 12. \end{aligned}$$

The resulting closing delay $B(\boldsymbol{x}) \in \{0, \ldots, 12\}$ is added to the year of the last payment (or to the reporting year if there is no payment). If this sum exceeds the value 11, the claim is still open at the end of the last modeled development year. This provides the claim status process $(I^{(j)}(\boldsymbol{x}))_{0 \le j \le 11}$ with

$$I^{(j)}(\boldsymbol{x}) = \begin{cases} 1, & \text{if } Z = 1 \text{ and } j < B(\boldsymbol{x}) + \max\left\{0 \le j \le 11 \,\middle|\, C^{(j)} \neq 0\right\}, \\ 1, & \text{if } Z = 0 \text{ and } j < B(\boldsymbol{x}) + T, \\ 0, & \text{else.} \end{cases}$$

(ii) Case $V(\boldsymbol{x}) = 1$ (with re-opening): if we have at least one payment for the considered claim ($Z = 1$), then the first settlement time $B_1(\boldsymbol{x})$ is simulated from a uniform distribution on the set

$$\left\{T, \ldots, \max\left\{0 \le j \le 11 \,\middle|\, C^{(j)} \neq 0\right\}\right\}.$$

The second settlement time $B_2(\boldsymbol{x})$ is simulated from a uniform distribution on the set

$$\left\{\max\left\{0 \le j \le 11 \,\middle|\, C^{(j)} \neq 0\right\} + 2, \ldots, 13\right\}.$$

In particular, the first settlement arrives between the reporting year and the year of the last payment. Then, the claim gets re-opened in the year following the first settlement. The second settlement, if there is one, arrives between two years after the year of the last payment and the last modeled development year. In case the second settlement arrives after the last modeled development year, we simply cannot observe it and the claim is still open at the end of the last modeled development year. In case the first settlement happens at the last modeled development year, we do not even observe the re-opening.
If the claim does not have any payment ($Z = 0$), then we set $B_1(\boldsymbol{x}) = T$ for the first settlement time. In particular, the claim gets closed for the first time in the same year as it is reported. The second settlement time $B_2(\boldsymbol{x})$ is simulated from a uniform distribution on the set $\{T + 2, \ldots, 13\}$.

This leads to the claim status process $(I^{(j)}(\boldsymbol{x}))_{0\leq j\leq 11}$ with

$$I^{(j)}(\boldsymbol{x}) \;=\; \begin{cases} 1, & \text{if } j < B_1(\boldsymbol{x}) \text{ or } B_1(\boldsymbol{x}) < j < B_2(\boldsymbol{x}), \\ 0, & \text{if } j = B_1(\boldsymbol{x}) \text{ or } j \geq B_2(\boldsymbol{x}). \end{cases}$$

3 Model calibration using momentum-based gradient descent

In Section 2 we have introduced several neural networks that need to be calibrated to the data. This calibration involves the choice of the numbers of hidden neurons q_1 and q_2 as well as the choice of the corresponding network parameter $\boldsymbol{\alpha}$. We first focus on the network parameter $\boldsymbol{\alpha}$ for given q_1 and q_2.

3.1 Gradient descent methods

State-of-the-art for finding the optimal network parameter $\boldsymbol{\alpha}$ w.r.t. a given differentiable loss function $\boldsymbol{\alpha} \mapsto \mathcal{L}(\boldsymbol{\alpha})$ is the gradient descent method (GDM). The GDM locally improves the loss in an iterative way. Consider the Taylor approximation of $\mathcal{L}$ around $\boldsymbol{\alpha}$, then

$$\mathcal{L}(\widetilde{\boldsymbol{\alpha}}) \;=\; \mathcal{L}(\boldsymbol{\alpha}) + \left(\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\alpha})\right)'\left(\widetilde{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\right) + o\left(\|\widetilde{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\|\right),$$

as $\|\widetilde{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\| \to 0$. The locally optimal move points into the direction of the negative gradient $-\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\alpha})$. If we choose a learning rate $\varrho > 0$ into that direction, we obtain a local loss decrease

$$\mathcal{L}\left(\boldsymbol{\alpha} - \varrho\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\alpha})\right) \;\approx\; \mathcal{L}(\boldsymbol{\alpha}) - \varrho\left\|\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\alpha})\right\|^2, \tag{3.1}$$

for ϱ small. Iterative application of these locally optimal moves (with tempered learning rates) will converge ideally to the (local) minimum of the loss function. Note that (a) it is possible to end up in saddle points; (b) different starting points of this algorithm should be explored to see whether we converge to different (local) minima resp. saddle points and (c) the speed of convergence should be fine-tuned. An improved version of the GDM is the so-called momentum-based GDM introduced in [9]. Consider a velocity vector $\mathbf{v}$ with the same dimensions as $\boldsymbol{\alpha}$ and initialize $\mathbf{v} = \mathbf{0}$, corresponding to zero velocity in the beginning. Then, in every iteration step of the GDM, we are building up velocity to achieve a faster convergence. In formulas, this provides

$$\begin{aligned} \mathbf{v} &\;\leftarrow\; \mu\mathbf{v} - \varrho\,\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\alpha}), \\ \boldsymbol{\alpha} &\;\leftarrow\; \boldsymbol{\alpha} + \mathbf{v}, \end{aligned}$$

where $\mu \in [0,1]$ is the momentum coefficient controlling how fast velocity is built up. By choosing $\mu = 0$, we get the original GDM without a velocity vector, see (3.1). Fine-tuning $0 < \mu \leq 1$ may lead to faster convergence. We refer to the relevant literature for more on this topic.

3.2 Gradients of the loss functions involved

In Section 2 we have met three different model types of neural networks:

- categorical case with more than two categorical classes;

- Bernoulli case with exactly two categorical classes;
- log-normal case.

In order to apply the momentum-based GDMs, we need to calculate the gradients of the corresponding loss functions of these three model types. As illustrations, we choose the reporting delay T for the categorical case, the payment indicator Z for the Bernoulli case, and the total individual claim size Y for the log-normal case.

3.2.1 Categorical case (with more than two categorical classes)

The loss function $\boldsymbol{\alpha} \mapsto \mathcal{L}(\boldsymbol{\alpha})$ for the modeling of the reporting delay T is given in (2.8). The gradient $\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\alpha})$ can be calculated as

$$\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\alpha}) \;=\; -2\sum_{i=1}^{n_1}\sum_{t\in\mathcal{T}}\mathbb{1}_{\{T_i=t\}}\nabla_{\boldsymbol{\alpha}}\log\pi_t(\boldsymbol{x}_i) \;=\; -2\sum_{i=1}^{n_1}\sum_{t\in\mathcal{T}}\mathbb{1}_{\{T_i=t\}}\frac{1}{\pi_t(\boldsymbol{x}_i)}\nabla_{\boldsymbol{\alpha}}\pi_t(\boldsymbol{x}_i).$$

We have for the last gradients

$$\nabla_{\boldsymbol{\alpha}}\pi_t(\boldsymbol{x}_i) \;=\; \nabla_{\boldsymbol{\alpha}}\frac{\exp\{\mu_t(\boldsymbol{x}_i)\}}{\sum_{s\in\mathcal{T}}\exp\{\mu_s(\boldsymbol{x}_i)\}} \;=\; \pi_t(\boldsymbol{x}_i)\left(\nabla_{\boldsymbol{\alpha}}\mu_t(\boldsymbol{x}_i) - \sum_{s\in\mathcal{T}}\pi_s(\boldsymbol{x}_i)\nabla_{\boldsymbol{\alpha}}\mu_s(\boldsymbol{x}_i)\right),$$

for all $t \in \mathcal{T}$ and $i = 1, \ldots, n_1$. Collecting all terms, we conclude

$$\begin{aligned}\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\alpha}) \;&=\; -2\sum_{i=1}^{n_1}\sum_{t\in\mathcal{T}}\mathbb{1}_{\{T_i=t\}}\left(\nabla_{\boldsymbol{\alpha}}\mu_t(\boldsymbol{x}_i) - \sum_{s\in\mathcal{T}}\pi_s(\boldsymbol{x}_i)\nabla_{\boldsymbol{\alpha}}\mu_s(\boldsymbol{x}_i)\right)\\ &=\; -2\sum_{i=1}^{n_1}\sum_{t\in\mathcal{T}}\mathbb{1}_{\{T_i=t\}}\nabla_{\boldsymbol{\alpha}}\mu_t(\boldsymbol{x}_i) + 2\sum_{i=1}^{n_1}\sum_{s\in\mathcal{T}}\pi_s(\boldsymbol{x}_i)\nabla_{\boldsymbol{\alpha}}\mu_s(\boldsymbol{x}_i)\\ &=\; -2\sum_{i=1}^{n_1}\sum_{t\in\mathcal{T}}\left(\mathbb{1}_{\{T_i=t\}} - \pi_t(\boldsymbol{x}_i)\right)\nabla_{\boldsymbol{\alpha}}\mu_t(\boldsymbol{x}_i).\end{aligned}$$

There remains to calculate the gradients $\nabla_{\boldsymbol{\alpha}}\mu_t(\boldsymbol{x}_i)$, for all $t \in \mathcal{T}$ and $i = 1, \ldots, n_1$. This is done using the backpropagation algorithm, which in today's form goes back to [12].

3.2.2 Bernoulli case (two categorical classes)

We calculate the gradient $\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\alpha})$ for the modeling of the payment indicator Z with corresponding loss function $\mathcal{L}(\boldsymbol{\alpha})$ given in (2.10). We get as in the categorical case above

$$\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\alpha}) \;=\; -2\sum_{i=1}^{n_2}\left(\frac{Z_i}{\pi(\boldsymbol{x}_i)} - \frac{1-Z_i}{1-\pi(\boldsymbol{x}_i)}\right)\nabla_{\boldsymbol{\alpha}}\pi(\boldsymbol{x}_i),$$

with gradient

$$\nabla_{\boldsymbol{\alpha}}\pi(\boldsymbol{x}_i) \;=\; \frac{\exp\{-\mu(\boldsymbol{x})\}}{\left(1+\exp\{-\mu(\boldsymbol{x})\}\right)^2}\nabla_{\boldsymbol{\alpha}}\mu(\boldsymbol{x}_i) \;=\; \pi(\boldsymbol{x}_i)\left(1-\pi(\boldsymbol{x}_i)\right)\nabla_{\boldsymbol{\alpha}}\mu(\boldsymbol{x}_i),$$

for all $i = 1, \ldots, n_2$. Collecting all terms, we obtain

$$\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\alpha}) \;=\; -2\sum_{i=1}^{n_2}\left(Z_i - \pi(\boldsymbol{x}_i)\right)\nabla_{\boldsymbol{\alpha}}\mu(\boldsymbol{x}_i).$$

We again apply backpropagation to calculate the gradient $\nabla_{\boldsymbol{\alpha}}\mu(\boldsymbol{x}_i)$, for all $i = 1, \ldots, n_2$.

3.2.3 Log-normal case

Finally, the loss function $\mathcal{L}(\boldsymbol{\alpha})$ for the modeling of the total individual claim size Y is given in (2.14). Hence, for the gradient $\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\alpha})$, we have

$$\nabla_{\boldsymbol{\alpha}}\mathcal{L}(\boldsymbol{\alpha}) \;=\; \sum_{i=1}^{n_4} \nabla_{\boldsymbol{\alpha}} \left(\log Y_i^+ - \mu(\boldsymbol{x}_i)\right)^2 \;=\; -2\sum_{i=1}^{n_4} \left(\log Y_i^+ - \mu(\boldsymbol{x}_i)\right) \nabla_{\boldsymbol{\alpha}}\mu(\boldsymbol{x}_i),$$

where the last gradient $\nabla_{\boldsymbol{\alpha}}\mu(\boldsymbol{x}_i)$, for all $i = 1, \ldots, n_4$, is again calculated using backpropagation.

3.3 Choice of the numbers of hidden neurons

For each modeling step of our simulation machine, we still need to determine the optimal neural network in terms of the numbers q_1 and q_2 of hidden neurons. These hyperparameters are determined by splitting the original data set into a training set and a validation set, where for each calibration we choose at random 90% of the data for the training set. The training set is then used to fit the models for the different choices of hyperparameters q_1 and q_2 by minimizing the corresponding (training) in-sample losses of the functions $\boldsymbol{\alpha} \mapsto \mathcal{L}(\boldsymbol{\alpha})$. This is done as described in the previous sections (for given q_1 and q_2). The hyperparameter choices q_1 and q_2 (and model choices, respectively) are then done by choosing the model with the smallest (validation) out-of-sample loss on the validation set.

4 Chain-ladder analysis

In this section we use the calibrated stochastic simulation machine to perform a small claims reserving analysis. We generate data from the simulation machine and compare it to the real data. For both data sets, we analyze the resulting claims reporting patterns and the corresponding claims cash flow patterns. For claims reportings, we separate the individual claims $i = 1, \ldots, n$ by accident year $\texttt{AY} \in \{1994, \ldots, 2005\}$ and reporting delays $T \in \{0, \ldots, 11\}$. For claims cash flows, we separate the individual claims $i = 1, \ldots, n$ again by accident year $\texttt{AY} \in \{1994, \ldots, 2005\}$ and aggregate the corresponding payments over the development delays $j = 0, \ldots, 11$. The reported claims and the claims payments that are available by the end of accounting year 2005 then provide the so-called upper claims reserving triangles. These triangles of reported claims of real and simulated data are shown in Tables 2 and 3, the triangles of cumulative claims payments of real and simulated data are given in Tables 5 and 6. At a first glance, these triangles show that the simulated data looks very similar to the real data, with a slightly bigger similarity for claims reportings than for claims cash flows.

These data sets can be used to perform a chain-ladder claims reserving analysis. We therefore use Mack's chain-ladder model, for details see [6]. We calculate the chain-ladder reserves for both the real and the simulated data, and we also calculate Mack's square-rooted conditional mean square error of prediction $\sqrt{\text{msep}}$. We start the analysis on the claims reportings. Using the chain-ladder method, we can predict the number of incurred but not yet reported (IBNYR) claims. These are the predicted numbers of late reported claims in the lower triangles in Tables

accident	reporting delay T											
year AY	0	1	2	3	4	5	6	7	8	9	10	11
1994	861’899	59’056	1’540	460	230	154	84	56	50	28	32	12
1995	850’297	64’733	1’568	562	216	124	94	62	44	34	32	
1996	781’875	61’465	1’742	414	252	153	76	62	38	22		
1997	756’147	59’269	1’466	496	210	147	54	48	40			
1998	753’552	60’249	1’660	530	248	136	98	44				
1999	754’992	59’690	1’625	468	208	100	68					
2000	766’684	61’120	1’274	320	136	88						
2001	758’443	61’449	1’024	286	90							
2002	745’125	55’246	876	200								
2003	757’843	53’272	956									
2004	733’785	51’742										
2005	730’978											

Table 2: Triangle of reported claims of the real data.

accident	reporting delay T											
year AY	0	1	2	3	4	5	6	7	8	9	10	11
1994	860’337	60’143	1’837	553	256	155	101	70	57	35	36	21
1995	851’877	62’924	1’776	499	263	164	80	58	52	29	36	
1996	783’006	60’511	1’557	477	182	122	87	49	53	39		
1997	756’015	59’556	1’434	399	169	129	77	42	42			
1998	752’454	61’913	1’422	380	164	100	65	37				
1999	753’635	61’552	1’279	362	153	116	49					
2000	768’180	59’636	1’215	338	150	71						
2001	759’501	60’131	1’124	304	132							
2002	744’478	55’577	1’012	270								
2003	757’635	53’352	937									
2004	732’884	52’586										
2005	731’357											

Table 3: Triangle of reported claims of the simulated data.

2 and 3. The resulting predictions are provided in the 2nd and 5th columns of Table 4. We observe a high similarity between the results on the real and the simulated data. This similarity carries over to the prediction uncertainty analysis illustrated by the columns $\sqrt{\text{msep}}$ in Table 4. From this, we conclude that, at least from a chain-ladder reserving point of view, our stochastic simulation machine provides very reasonable claims reporting patterns.

accident year AY	CL predicted IBNYR claims	$\sqrt{\text{msep}}$	in %	CL predicted IBNYR claims	$\sqrt{\text{msep}}$	in %
	real data			simulated data		
1994	0			0		
1995	12	0	0.0 %	21	0	0.0 %
1996	40	0	0.4 %	52	0	0.4 %
1997	65	5	8.4 %	82	7	8.6 %
1998	105	7	6.2 %	129	9	6.6 %
1999	156	10	6.1 %	178	14	7.7 %
2000	235	19	8.0 %	255	21	8.3 %
2001	357	32	9.0 %	370	35	9.4 %
2002	536	65	12.2 %	535	53	9.9 %
2003	944	135	14.3 %	925	95	10.2 %
2004	2’201	330	15.0 %	2’170	265	12.2 %
2005	57’734	3’542	6.1 %	57’789	3’410	5.9 %
total	62’385	3’565	5.7 %	62’506	3’425	5.5 %

Table 4: Chain-ladder predicted numbers of IBNYR claims and Mack’s $\sqrt{\text{msep}}$ for the real and the simulated data.

accident	development delay j											
year AY	0	1	2	3	4	5	6	7	8	9	10	11
1994	78'433	120'396	130'167	134'749	137'143	138'798	139'994	141'052	142'106	142'913	143'652	144'247
1995	79'372	124'532	135'488	140'338	143'145	144'859	146'408	147'624	148'735	149'686	150'578	
1996	71'398	113'335	123'336	128'052	130'779	132'692	134'175	135'359	136'381	137'311		
1997	68'600	107'716	117'556	122'331	125'304	127'477	129'037	130'171	131'216			
1998	68'055	109'906	120'706	126'443	129'727	131'998	133'745	135'176				
1999	71'989	114'344	127'311	134'174	138'075	140'614	142'530					
2000	72'225	118'418	131'615	138'216	142'024	144'450						
2001	74'891	126'244	141'008	147'923	151'917							
2002	78'167	129'105	143'731	150'560								
2003	82'668	134'010	148'161									
2004	80'630	130'390										
2005	82'015											

Table 5: Triangle of cumulative claims payments (in 10'000 CHF) of the real data.

accident	development delay j											
year AY	0	1	2	3	4	5	6	7	8	9	10	11
1994	80'491	117'807	129'673	135'331	138'591	140'745	142'268	143'394	144'344	145'090	145'694	146'076
1995	79'170	116'943	129'313	135'330	138'785	141'079	142'743	144'045	145'115	146'002	146'760	
1996	71'675	107'228	119'578	125'407	128'771	131'009	132'566	133'718	134'657	135'472		
1997	68'857	104'291	116'406	122'177	125'495	127'662	129'194	130'388	131'270			
1998	67'418	103'415	116'194	122'262	125'673	127'877	129'452	130'588				
1999	69'308	107'587	121'381	127'930	131'534	133'924	135'502					
2000	73'359	113'266	127'878	134'804	138'806	141'431						
2001	73'338	115'626	131'197	138'481	142'591							
2002	74'887	117'602	133'850	141'539								
2003	81'921	127'461	144'444									
2004	81'394	128'864										
2005	86'837											

Table 6: Triangle of cumulative claims payments (in 10'000 CHF) of the simulated data.

Finally, Table 7 shows the results of the chain-ladder analysis for claims payments. Columns 2 and 5 of that table provide the chain-ladder (CL) reserves. These are the payment predictions for the cash flows paid after accounting year 2005 and complete the lower triangles in Tables 5 and 6. Also here we see high similarities between the real data and the simulated data analysis. The slightly shorter cash flow patterns in the simulated data carries over to the prediction uncertainties illustrated by the columns $\sqrt{\text{msep}}$ in Table 7. We remark that fine-tuning the two

accident year AY	CL reserves	$\sqrt{\text{msep}}$	in %	CL reserves	$\sqrt{\text{msep}}$	in %
	real data			simulated data		
1994	0			0		
1995	624	117	18.8 %	385	109	28.2 %
1996	1'337	146	10.9 %	991	161	16.2 %
1997	2'112	168	8.0 %	1'723	179	10.4 %
1998	3'224	177	5.5 %	2'636	187	7.1 %
1999	4'686	259	5.5 %	3'943	209	5.3 %
2000	6'476	394	6.1 %	5'826	230	3.9 %
2001	9'275	599	6.5 %	8'846	290	3.4 %
2002	13'049	889	6.8 %	12'489	421	3.4 %
2003	19'973	1'421	7.1 %	20'817	837	4.0 %
2004	32'532	2'394	7.4 %	36'647	2'050	5.6 %
2005	82'706	5'039	6.1 %	84'175	4'968	5.9 %
total	175'994	6'275	3.6 %	178'076	5'732	3.2 %

Table 7: Chain-ladder reserves for claims payments (in 10'000 CHF) and Mack's $\sqrt{\text{msep}}$ for the real and the simulated data.

variance parameters σ_+^2 for the total individual claim size and σ_-^2 for the total individual recovery simulations, see (2.13) and Section 2.6, or making them even feature $\boldsymbol{x}$ dependent, could further increase the similarity between the real and the simulated data. For the time-being, we refrain from doing so because we believe that the current calibration is sufficiently good for simulating reasonable data that may serve as a common ground for individual claims reserving research.

Acknowledgment

Our greatest thanks go to Suva, Dr. Peter Blum and Dr. Olivier Steiger for providing data, their insights and for their immense support.

References

[1] Antonio, K., Plat, R. (2014). Micro-level stochastic loss reserving for general insurance. *Scandinavian Actuarial Journal* **2014/7**, 649-669.

[2] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)* **2/4**, 303-314.

[3] Hiabu M., Margraff, C., Martínez-Miranda, M.D., Nielsen, J.P. (2016). The link between classical reserving and granular reserving through double chain-ladder and its extensions. *British Actuarial Journal* **21/1**, 97-116.

[4] Hornik, K., Stinchcombe, M., White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks* **2/5**, 359-366.

[5] Jessen, A.H., Mikosch, T., Samorodnitsky, G. (2011). Prediction of outstanding payments in a Poisson cluster model. *Scandinavian Actuarial Journal* **2011/3**, 214-237.

[6] Mack, T. (1993). Distribution-free calculation of the standard error of chain ladder reserve estimates. *ASTIN Bulletin* **23/2**, 213-225.

[7] Martínez-Miranda, M.D., Nielsen, J.P., Verrall, R.J., Wüthrich, M.V. (2015). The link between classical reserving and granular reserving through double chain-ladder and its extensions. *Scandinavian Actuarial Journal* **2015/5**, 383-405.

[8] Pigeon, M., Antonio, K., Denuit, M. (2013). Individual loss reserving with the multivariate skew normal framework. *ASTIN Bulletin* **43/3**, 399-428.

[9] Rumelhart, D.E., Hinton, G.E., Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature* **323/6088**, 533-536.

[10] Taylor, G., McGuire, G., Sullivan, J. (2008). Individual claim loss reserving conditioned by case estimates. *Annals of Actuarial Science* **3/1-2**, 215-256.

[11] Verrall, R.J., Wüthrich, M.V. (2016). Understanding reporting delay in general insurance. *Risks* **4/3**, 25.

[12] Werbos, P. (1982). Applications of advances in nonlinear sensitivity analysis. *System modeling and optimization* **1982**, 762-770.

[13] Wüthrich (2018). Machine learning in individual claims reserving. *To appear in Scandinavian Actuarial Journal.*

A Descriptive statistics of the chosen data set

In this appendix we provide descriptive statistics of the data used to calibrate the individual claims history simulation machine. For confidentiality reasons, we can only show aggregate statistics of the claims portfolio.

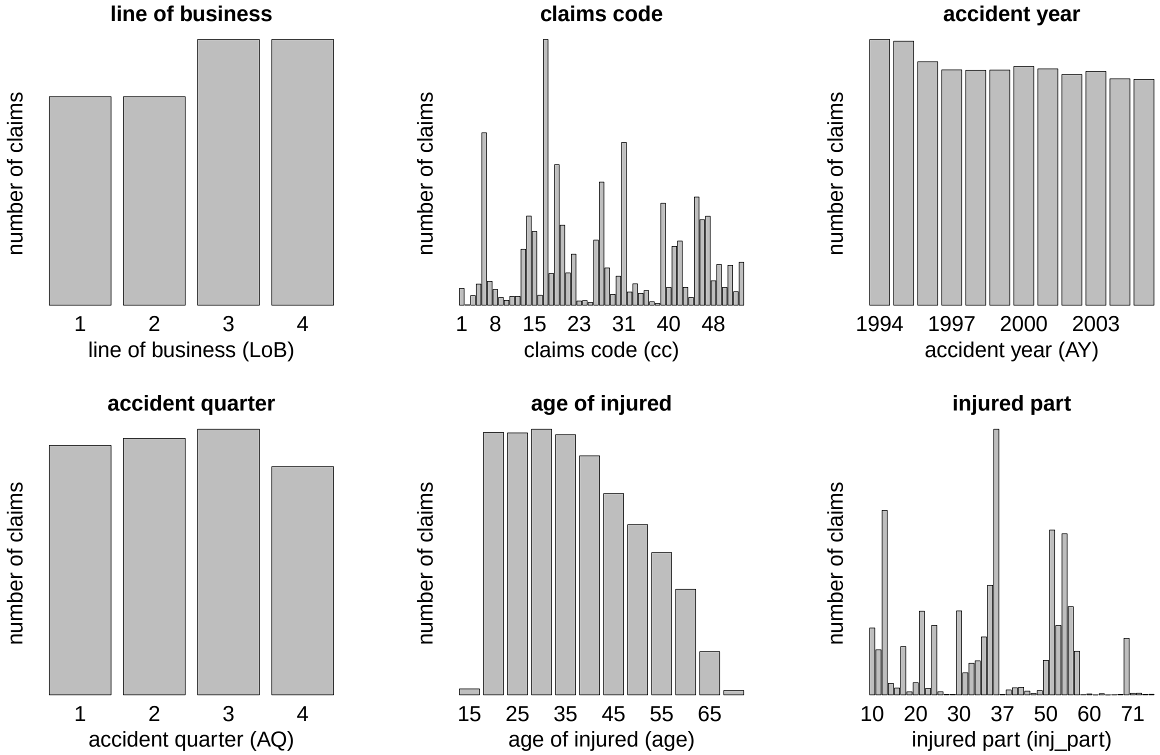

Figure 2: Portfolio distributions w.r.t. the features `LoB` (top left), `cc` (top center), `AY` (top right), `AQ` (bottom left), `age` (bottom center) and `inj_part` (bottom right).

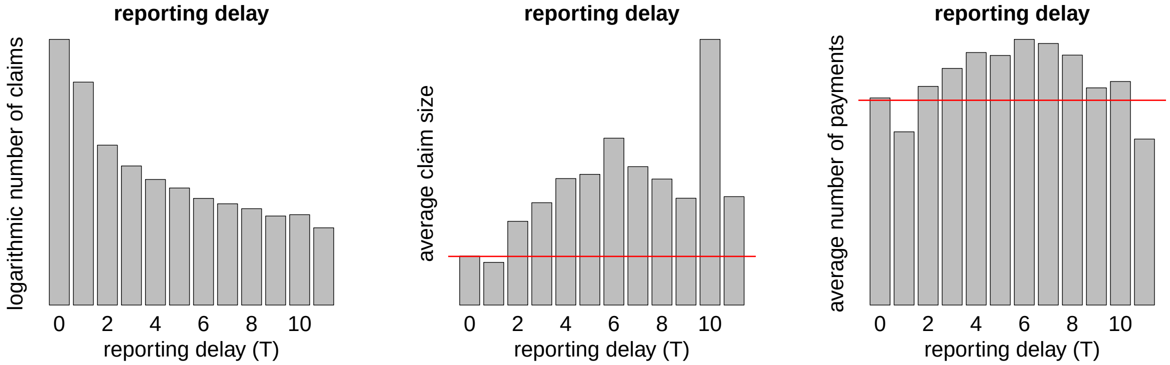

Figure 3: Logarithmic number of claims (left), average claim size (center) and average number of payments (right) w.r.t. the reporting delay T; the red lines show the averages.

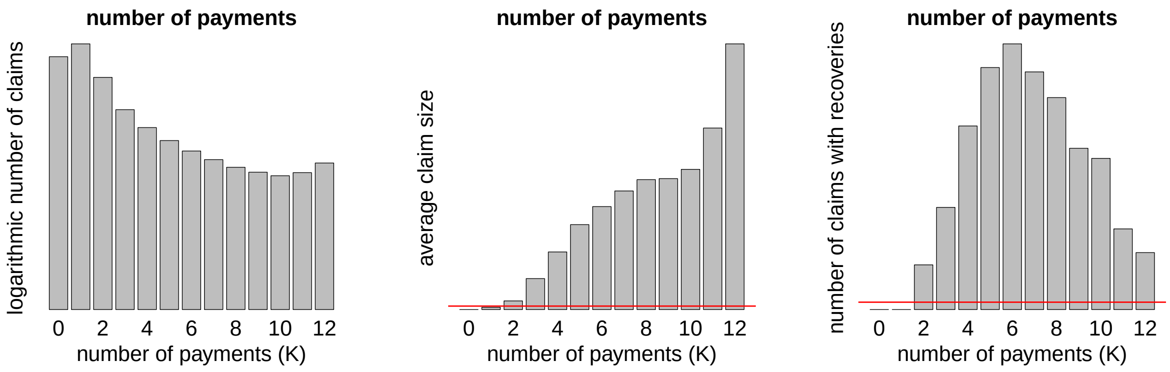

Figure 4: Logarithmic number of claims (left), average claim size (center) and number of claims with recoveries (right) w.r.t. the number of payments K; the red lines show the averages.

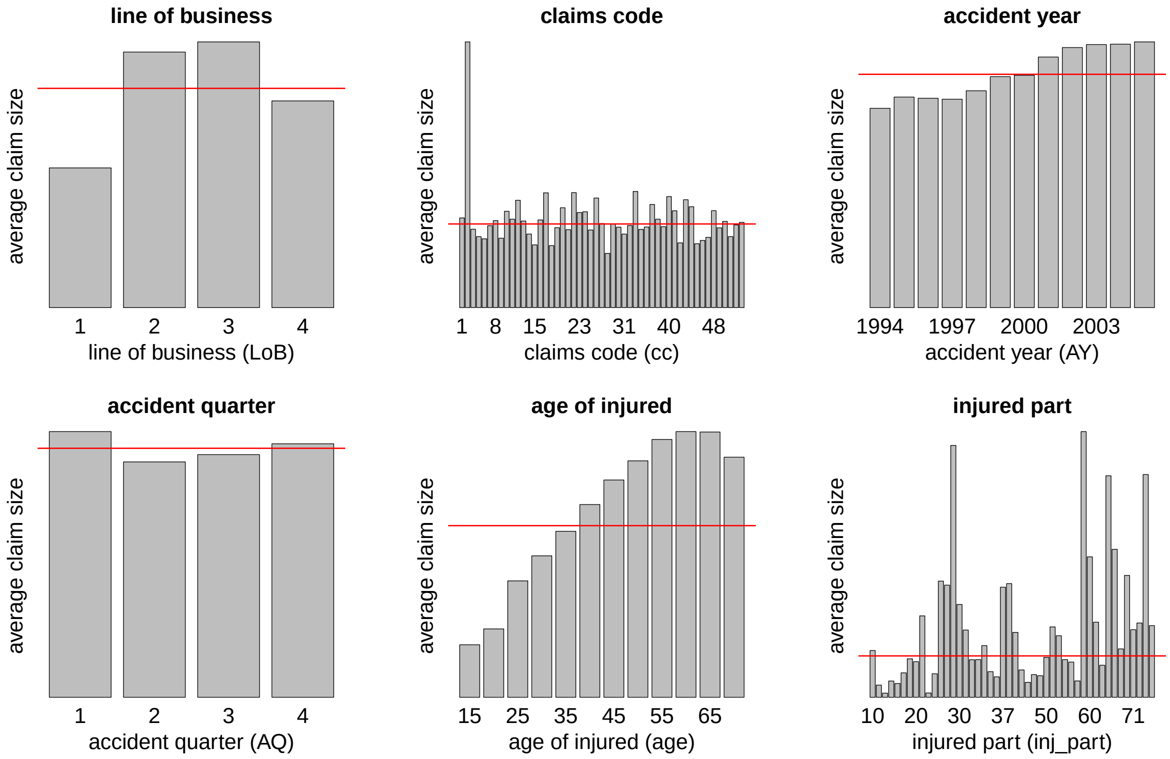

Figure 5: Average claim size w.r.t. the features `LoB` (top left), `cc` (top center), `AY` (top right), `AQ` (bottom left), `age` (bottom center) and `inj_part` (bottom right); the red lines show the averages.

B Procedure of generating a synthetic portfolio

In order to use the stochastic simulation machine derived above, we require a portfolio of features $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}_1$, see (2.1). Therefore, we need an additional scenario generator that simulates reasonable synthetic portfolios. In this appendix we describe the design of our portfolio scenario generator which provides portfolios similar in structure to the original portfolio.

Our algorithm of synthetic portfolio generation uses the following input parameters:

- V = totally expected number of claims;
- $(p_l)_{1 \le l \le 4}$ = categorical distribution for the allocation of the claims to the 4 lines of business;
- $(r_l)_{1 \le l \le 4}$ = growth parameters for the numbers of claims in the 12 accident years for each of the 4 lines of business.

In a first step, we use these parameters to simulate the total number of claims and allocate them to the lines of business `LoB` and the accident years `AY`. We start by simulating $(V_l)_{1 \le l \le 4}$ according to

$$(V_l)_{1 \le l \le 4} \ \sim \ \text{Multinomial}\left(V, (p_l)_{1 \le l \le 4}\right).$$

To determine the distribution of the claims among the 12 accident years within each line of business $l = 1, \ldots, 4$, we simulate $(X_j^{(l)})_{1 \le l \le 4, 2 \le j \le 12}$ according to

$$X_j^{(l)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(r_l, r_l^2).$$

Then, we define the weights $W_1^{(l)} = 1$ and

$$W_j^{(l)} \ = \ W_{j-1}^{(l)} \exp\left\{X_j^{(l)}\right\},$$

for all $l = 1, \ldots, 4$ and $j = 2, \ldots, 12$. Finally, we set

$$V_{l,j} \ = \ V_l \, \frac{W_j^{(l)}}{\sum_{j'=1}^{12} W_{j'}^{(l)}},$$

for all $l = 1, \ldots, 4$ and $j = 1, \ldots, 12$, to be the expected number of claims in line of business l with accident year j. Conditionally given $\mathbf{V} = (V_{1,1}, V_{1,2}, \ldots, V_{4,12})$, we simulate the number of claims $N_{l,j}$ in line of business l with accident year j by

$$N_{l,j} \mid \mathbf{V} \overset{\text{ind.}}{\sim} \text{Poi}(V_{l,j}),$$

for all $l = 1, \ldots, 4$ and $j = 1, \ldots, 12$. Note that we have $\mathbb{E}[\sum_{j=1}^{12} N_{l,j}] = V p_l$, which justifies the above modeling choices.

After having simulated the number of claims $N_{l,j}$ for each line of business l and accident year j, we need to establish these claims with the remaining feature components `cc`, `AQ`, `age` and `inj_part`. This is achieved by choosing a multivariate distribution having a Gaussian copula and appropriate marginal densities. These densities and the covariance parameters of the Gaussian copula have been estimated from the real data. For the explicit parametrization, we refer to the R-function `Feature.Generation` in our simulation package.

C Sensitivities of selected neural networks

In this final appendix we consider 11 selected neural networks of our simulation machine and present the impact on the response variable of the respective most influential features. For each neural network considered, we use the corresponding calibration data set, fix a feature component (e.g. the accident quarter AQ) and vary its value over its entire domain (e.g. $\{1, \ldots, 4\}$ for the accident quarter AQ) to analyze the sensitivities in this feature component.

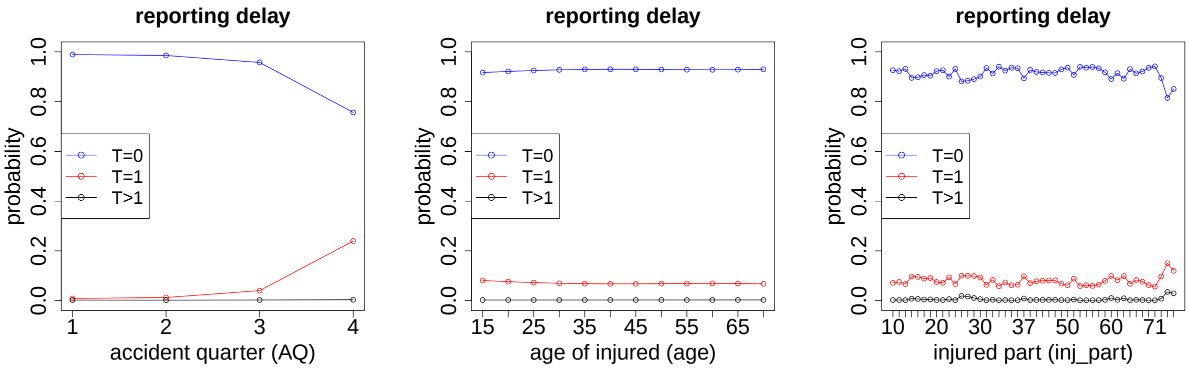

Figure 6: Reporting delay T w.r.t. the features AQ (left), age (center) and inj_part (right).

In Figure 6 we analyze the reporting delay T as a function of the features AQ, age and inj_part. Not surprisingly, the accident quarter has the biggest influence, because a claim occurring in December is likely to be reported only in the next accounting year.

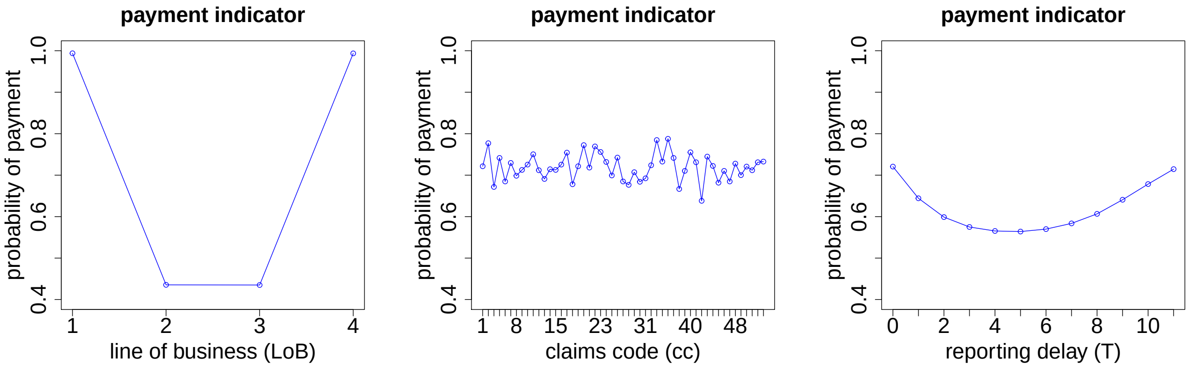

Figure 7: Payment indicator Z w.r.t. the features LoB (left), cc (center) and reporting delay T (right).

Figure 7 tells us that claims in line of business LoB $\in \{1, 4\}$ almost always have a payment. In contrast, we expect only roughly half of the claims in line of business LoB $\in \{2, 3\}$ to have a payment. Furthermore, the claims code cc causes some variation in the probability of having a payment, and claims with either a small or a large reporting delay T have a higher probability of having a payment than claims with a medium reporting delay.

Recall that in determining the number of payments K, we use two neural networks, where in the first one we model whether we have $K = 1$ or $K > 1$ payments. According to Figure 8, claims

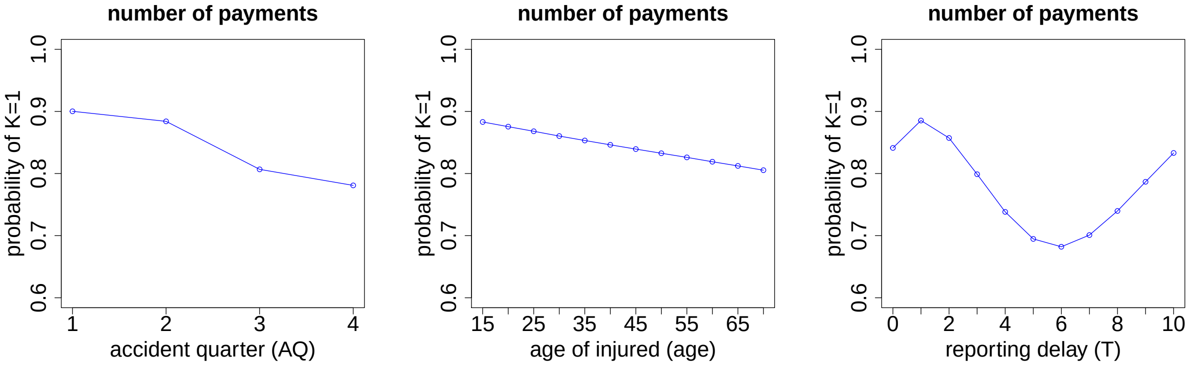

Figure 8: Indicator whether we have $K = 1$ or $K > 1$ payments w.r.t. the features `AQ` (left), `age` (center) and reporting delay T (right).

that occur later in a year tend to have a higher probability of having more than one payment. The same holds true with increasing age of the injured. In passing from reporting delay $T = 0$ to $T = 1$, the probability of having only one payment increases. But then we observe a sinus curve shape in that probability as a function of T.

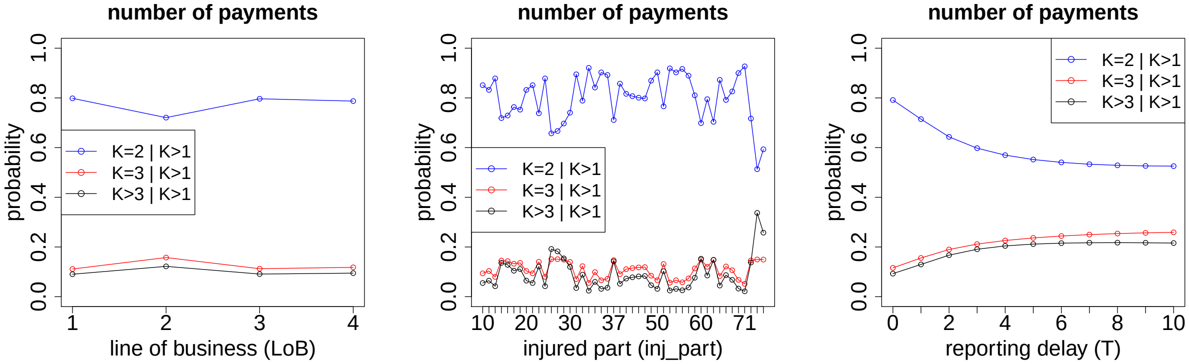

Figure 9: Conditional distribution of the number of payments K, given $K > 1$, w.r.t. the features `LoB` (left), `inj_part` (center) and reporting delay T (right).

The second neural network used to determine the number of payments K models the distribution of K, conditioned on $K > 1$. As we see in Figure 9, claims in line of business `LoB` $= 2$ tend to have more payments than claims in other lines of business, and both `inj_part` and reporting delay T heavily influence the number of payments.

In Figure 10 we present sensitivities for the expected total individual claim size Y on the log scale. The main drivers here are the line of business `LoB` and the number of payments K.

Figure 11 tells us that claims in line of business `LoB` $\in \{1, 4\}$ almost never have a recovery. Moreover, the probability of having at least one recovery payment first increases with the number of payments K but then slightly decreases again. Finally, up to 50% of the claims with a small total individual claim size Y (of less than 10 CHF) have a recovery. This also comprises claims whose recovery is almost equal to the total gross claim amount, leading to a small net claim size. In general, the higher the total individual claim size, the less likely are recovery payments.

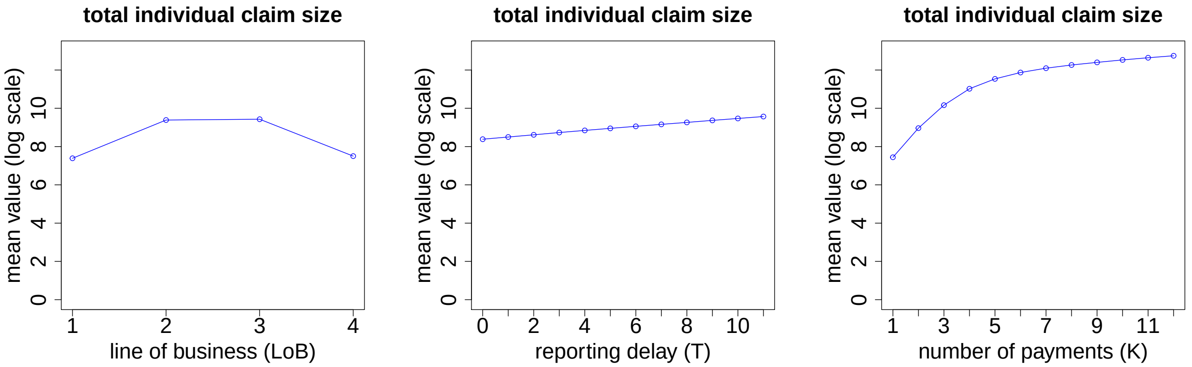

Figure 10: Total individual claim size Y (on log scale) w.r.t. the features LoB (left), reporting delay T (center) and number of payments K (right).

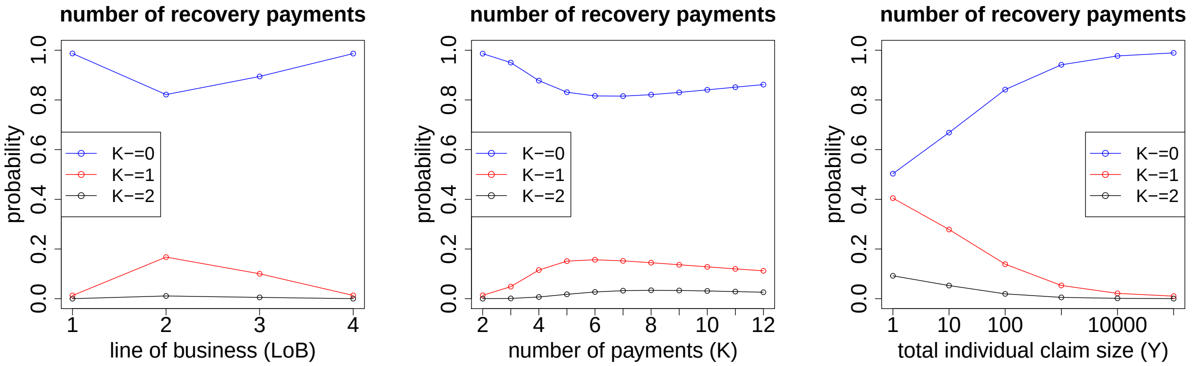

Figure 11: Number of recovery payments K^{-} w.r.t. the features LoB (left), number of payments K (center) and total individual claim size Y (right).

According to Figure 12, the total individual recovery Y^{-} is substantially higher for claims in line of business $\texttt{LoB} \in \{2, 3\}$, compared to claims in line of business $\texttt{LoB} \in \{1, 4\}$. Furthermore, if we have a recovery, then the higher the number of payments K and the total individual claim size Y, the higher also the recovery, where the increase w.r.t. the number of payments is decisively more pronounced.

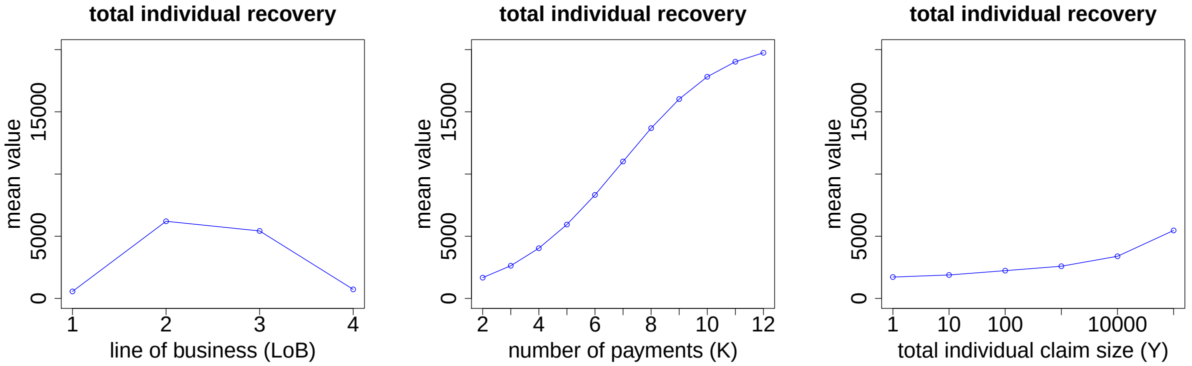

Figure 12: Total individual recovery Y^{-} w.r.t. the features LoB (left), number of payments K (center) and total individual claim size Y (right).

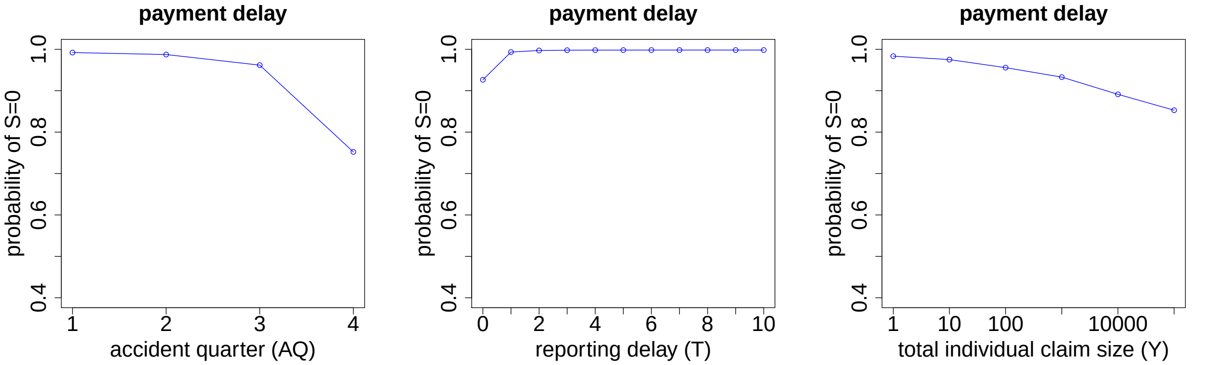

Figure 13: Indicator whether we have payment delay $S = 0$ or $S > 0$ in the case of $K = 1$ payment w.r.t. the features `AQ` (left), reporting delay T (center) and total individual claim size Y (right).

In determining the payment delay S for claims with exactly one payment, we use two neural networks. In the first one we model whether $S = 0$ or $S > 0$ and in the second one we consider the conditional distribution of S, given $S > 0$. Here we only present sensitivities for the first neural network. We observe, see Figure 13, that the probability of a payment delay equal to zero decreases with increasing accident quarter `AQ` and increasing total individual claim size Y. In particular, claims that occur in the last quarter of a year have a considerably higher probability of having a payment delay. This might be explained by claims for which the short time lag between the accident date and the end of the year only suffices for claims reporting but not for claims payments. Finally, claims with a reporting delay $T > 0$ almost never have an additional payment delay.

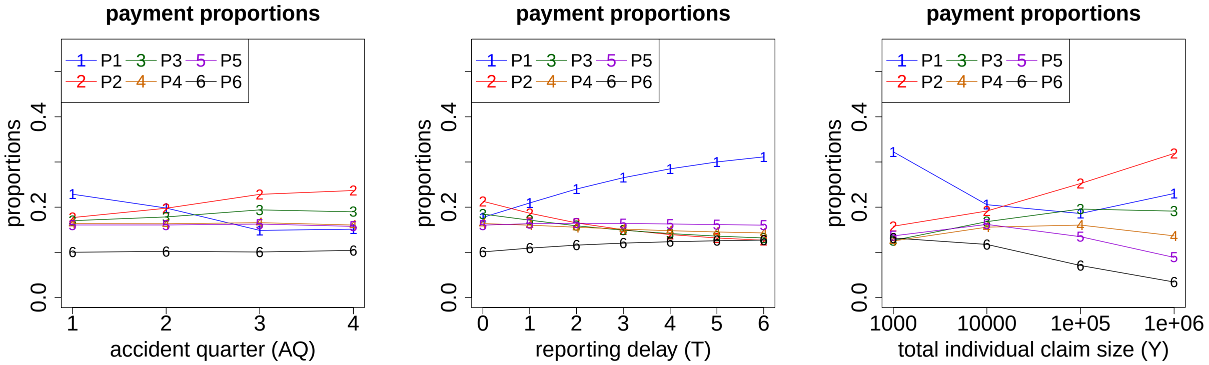

Figure 14: Proportions $P^{(1)}, \ldots, P^{(6)}$ of the total gross claim amount $Y + Y^-$ paid in the $K^+ = 6$ positive payments w.r.t. the features `AQ` (left), reporting delay T (center) and total individual claim size Y (right).

As a representative of the neural networks that calculate the proportions with which the total gross claim amount $Y + Y^-$ is distributed among the K^+ positive payments, we choose the one for $K^+ = 6$. According to Figure 14, we see some monotonicity, but apart from that these proportions do not vary considerably. For claims which occur early during a year or have a high reporting delay T or a comparably small total individual claim size Y, the biggest proportion of the total gross claim amount is paid in the first (positive) payment.

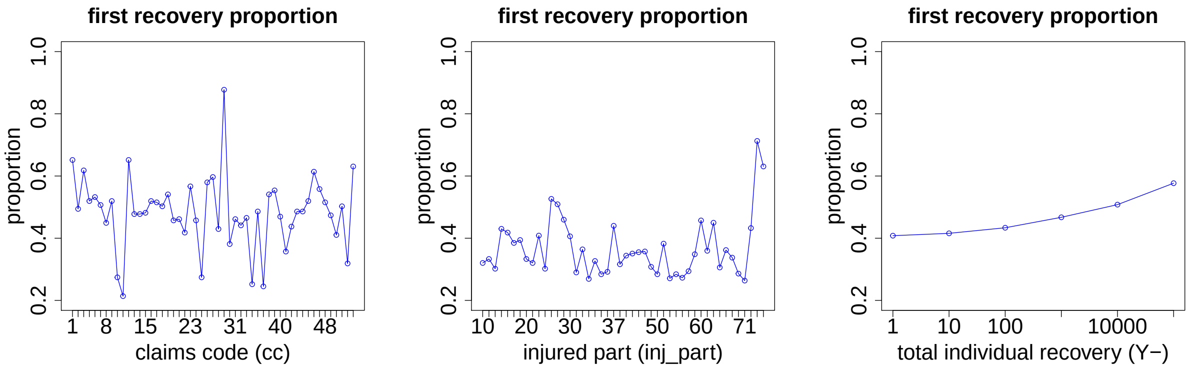

Figure 15: Proportion P^- of the total individual recovery Y^- paid in the first recovery payment in the case of $K^- = 2$ recovery payments w.r.t. the features `cc` (left), `inj_part` (center) and total individual recovery Y^- (right).

According to Figure 15, in the case of $K^- = 2$ recovery payments, the proportions paid in the first recovery payment vary substantially for the different values of the features `cc` and `inj_part`. We also observe that the higher the total individual recovery Y^-, the higher the proportion paid in the first recovery.

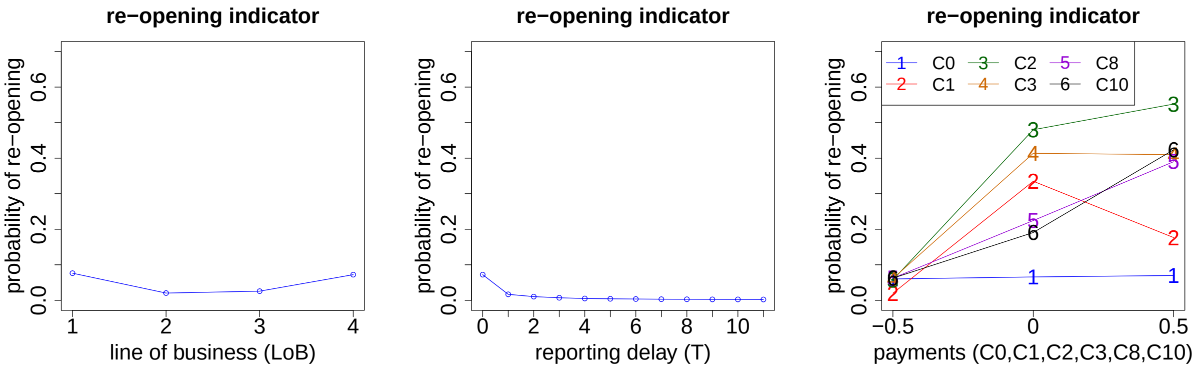

Figure 16: Re-opening indicator V w.r.t. the features `LoB` (left), reporting delay T (center) and yearly payments $C^{(0)}, C^{(1)}, C^{(2)}, C^{(3)}, C^{(8)}, C^{(10)}$ (right).

In Figure 16 we see that claims in line of business `LoB` $\in \{1, 4\}$ have a higher re-opening probability than claims in line of business `LoB` $\in \{2, 3\}$. Moreover, the higher the reporting delay T of a claim, the lower the rate of reopening. Finally, the probability of re-opening heavily depends on the cash flow. In order to not overload the plot, we only show sensitivities w.r.t. the payments $C^{(0)}, C^{(1)}, C^{(2)}, C^{(3)}, C^{(8)}, C^{(10)}$. Recall that for this neural network, the yearly payments are coded with the values $-\frac{1}{2}$, 0 and $\frac{1}{2}$, see (2.21). Summarizing, one can say that if we have a payment after the first development year, then the probability of re-opening is quite high.