Collective Reserving using Individual Claims Data

Łukasz Delong* Mathias Lindholm† Mario V. Wüthrich‡

Version of April 22, 2020

Abstract

The aim of this paper is to operationalize claims reserving based on individual claims data. We design a modeling architecture that is based on six different neural networks. Each network is a separate module that serves a certain modeling purpose. We apply our architecture to individual claims data and predict their settlement processes on a monthly time grid. A proof of concept is provided by benchmarking the resulting claims reserves with the ones received from the classical chain-ladder method which uses much coarser (aggregated) data.

Keywords. Claims reserving, individual claims data, micro-level reserving, neural networks, IBNR claims, RBNS claims, chain-ladder method, over-dispersed Poisson model.

1 Introduction

The aim of this paper is to operationalize claims reserving based on individual claims data. Stochastic models for individual claims reserving were introduced by Arjas [2] and Norberg [21, 22]. These authors use marked point processes to describe individual claims, but they provide little guidance on the use of these models in practical applications. Recently, more statistically oriented (semi-)parametric models have been studied by Larsen [15], Taylor et al. [27], Zhao et al. [34], Jessen et al. [12], Pigeon et al. [24], Antonio–Plat [1], Badescu et al. [3], Maciak et al. [18] and Okine et al. [23]. Still some of these proposals are not very operational, yet, because the assumed modeling frameworks are often too restrictive from an applied perspective. Individual claims reserving has only started to flourish with the emergence of machine learning methods in insurance: some papers are based on regression trees and gradient boosting, see Wüthrich [32], Lopez et al. [17], De Felice–Moriconi [7], and Duval–Pigeon [9], others are based on non-parametric and kernel methods, see Rosenlund [25], Bischofberger et al. [5], Baudry–Robert [4] or on neural networks, see Gabrielli [11], Kuo [14], Ticconi [28] and Delong–Wüthrich [8].
Surprisingly, many of these approaches pay little attention to the data itself, but they describe in much detail the methods used. This is surprising because the general aim here is to solve a data driven real world problem and, thus, the data should play the most important and central role in this modeling problem. We take a different approach here, namely, we start from the data using exploratory data analysis to build an appropriate model around the data. Exploratory data analysis serves in statistics at formulating appropriate hypotheses, see Tukey [29], which are then

*SGH Warsaw School of Economics, Institute of Econometrics, lukasz.delong@sgh.waw.pl
†Department of Mathematics, Stockholm University, lindholm@math.su.se
‡RiskLab, Department of Mathematics, ETH Zurich, mario.wuethrich@math.ethz.ch

verified (or rejected) using extended statistical testing, this often results in causal relationship conclusions. Whereas in machine learning one usually only infers correlations from the given data, which is sufficient to make good predictions, see Breiman [6]. This seems questionable from a statistical point of view and a good compromise between these two modeling cultures and communities still needs to be found. We will use exploratory tools to build reasonable models around the data, and we discuss assumptions often made in the claims reserving literature.
This paper is related to our previous work [8] which uses neural networks for individual claims reserving. We provide two essential new parts here. The first new contribution is an estimate of claims reserves for both reported *and* incurred but not reported claims. The latter reserves are missing in [8], and we propose a conceptually new approach for dealing with incurred but not reported claims. The second contribution is that we come up with a carefully designed simpler model compared to [8]. This simplification is crucial in improving run times, it allows us to consider finer time scales, and we verify that this simpler model provides accurate reserves by benchmarking it with the well-known chain-ladder method. Finer time scales are crucial if one wants to capture seasonal patterns and, for instance, sensitivities in insurance product features such as waiting periods in accident and health insurance, we refer to Figure 17 below.

Overview of our modeling architecture.
In Section 2 we introduce the claims reserving problem and state the basic model assumptions. Our claims reserving architecture is based on six neural networks which can be viewed as separate modules serving a specific modeling purpose. The 1st network is introduced in Section 3, this network serves at predicting incurred but not reported (IBNR) claim counts. In Section 4 we introduce the 2nd network that serves at modeling the two dimensional process of payment dates and closings of reported but not settled (RBNS) claims. The 3rd and 4th networks in Section 5 model claim payments as well as recovery payments of RBNS claims: the 3rd network indicates whether we have a positive payment or a negative recovery payment, and the 4th network determines its size. In Section 6 we introduce the last two networks that serve at modeling amounts of IBNR claims: these claim amounts are modeled by a compound distribution with the 5th network indicating whether we have a zero IBNR claim or not, and the 6th network determining the amount of a non-zero IBNR claim. Our modeling is done on two different time scales, we are using a weekly time grid for IBNR claims and a monthly time grid for RBNS claims. These choices are mainly determined by the reporting frequency of our data, but also by computational and statistical constraints. Finally, in Section 7 we conclude and give an outlook.

2 Notation and basic model assumptions

2.1 Individual claims and basic model assumptions

We start by describing a variant of the collective reserving model (CRM) of Wahl et al. [30] and Lindholm–Verrall [16]. This model can be seen as a discrete time and aggregated version of the individual claims model of Norberg [21, 22]. The original CRM is defined on a yearly time grid, for the moment we consider a daily time grid. A daily time grid is usually the smallest time scale available in real insurance data. An individual claim is described by a vector $\mathcal{C} = (T, U, W, (Y_s)_{s\geq 0}, (O_s)_{s\geq 0}; \vartheta)$:

variable	
T	accident date (in daily units as yyyy/mm/dd)
U	reporting date (in daily units as yyyy/mm/dd)
$W = U - T \geq 0$	non-negative reporting delay (in daily units)
$s \geq 0$	settlement delay after reporting date $U = T + W$ (in daily units)
$(Y_s)_{s\geq 0}$	claim payment process with payments Y_s at times $U + s$ (in daily time units)
$\mathcal{Y}_s = \mathbb{1}_{\{Y_s \neq 0\}}$	indicator for claim payments at times $U + s$ (in daily time units)
$(O_s)_{s\geq 0}$	$\{1, 0\}$-valued (daily) claim status process for open/closed claim at times $U + s$
ϑ	static claim features such as claim type

Our claim $\mathcal{C}$ corresponds to the accident date and the mark process described in (2.3) of Larsen [15]. Roughly speaking this process consists of two components: (1) a claim occurrence process describing accident dates T of claims, and (2) at every accident date a second (mark) process is launched describing the remaining components $(U, W, (Y_s)_{s\geq 0}, (O_s)_{s\geq 0}; \vartheta)$ of a given claim.

Assumptions 2.1 *Starting from a sufficiently rich probability space $(\Omega, \mathcal{F}, \mathbb{P})$, we assume that all considered claims $\mathcal{C}$ live on $(\Omega, \mathcal{F}, \mathbb{P})$ and are i.i.d. w.r.t. $\mathbb{P}$. Moreover, we assume that there is a maximal reporting delay $m > 0$ such that all claims $\mathcal{C}$ satisfy $0 \leq W < m$, $\mathbb{P}$-a.s.*

These are Assumptions (A1)-(A2) of Lindholm–Verrall [16]. Note that the i.i.d. assumption still allows us to model, e.g., seasonal patterns because the i.i.d. property (only) reflects that we select at random a claim from the pool of all claims over the entire observation period.

2.2 Time scales

Above, we have introduced claims $\mathcal{C} = (T, U, W, (Y_s)_{s\geq 0}, (O_s)_{s\geq 0}; \vartheta)$ on a daily time scale. Usually accident dates T and reporting dates U are available on a daily time scale. However, other variables like payments $(Y_s)_{s\geq 0}$ or the claim status process $(O_s)_{s\geq 0}$ may only be available on a coarser time scale due to memory, data warehousing and business process constraints.
In our approach, we are going to use problem and data adapted time scales. In particular, we will choose a *weekly* time scale to study accident dates T, reporting dates U and reporting delays W. The choice of a weekly time scale is justified by the fact that our claim occurrences have a weekly pattern (weekends behave differently from working days), and by selecting a weekly time scale we capture exactly one weekly cycle in each period. This will be done in Sections 3 and 6, where we treat incurred but not reported (IBNR) claims.
For the claim payment and claim status processes $(Y_k)_{k\geq 0}$ and $(O_k)_{k\geq 0}$ we will move to a *monthly* time scale because data is only available on this time scale (end-of-month data). To highlight this time change we use notation k for time (instead of s), thus, $k \geq 0$ will denote months for payment process and claim status modeling. This will be done in Sections 4 and 5, where we treat reported but not settled (RBNS) claims.
Whenever we discuss aggregated methods like the CRM or the chain-ladder model we will use an *annual* time grid, as it is typically done in industry. Aggregated methods cannot capture individual claim features which especially applies to seasonal factors. Therefore, aggregated methods typically act on a yearly time scale in order to capture an entire annual cycle.

2.3 The claims reserving problem

The typical situation in industry is that we have a cutoff date τ which reflects today's date. Throughout, we assume that the cutoff date is set to an end of a calendar year, i.e. a 31st of

December.[1] All variables that occur before τ are observable at this cutoff date, and the main task in claims reserving is to predict all variables that only occur after τ. We consider the run-off situation meaning that the exposures (for claim occurrence) are capped at τ, thus, all accident dates must fulfill $T \leq \tau$. The claims with reporting dates $U = T + W \leq \tau$ are reported at time τ and the claims with reporting dates $U = T + W > \tau \geq T$ are *incurred but not reported* (IBNR) claims. The reported claims can further be split in claims that are *closed* at time τ and claims that are *reported but not settled* (RBNS) at time τ. This is characterized by the claim status variable $O_{\tau-U} \in \{0, 1\}$ for closed and RBNS claims at time τ, respectively. The claims reserving exercise aims at predicting all payments Y_s after time τ, i.e. $s > \tau - U$, for claims with accident dates $T \leq \tau$. This comprises IBNR and RBNS claims, but it also contains closed claims at time τ which may be re-opened at a later time point (due to unexpected relapses).

2.4 Aggregated claims and the time ceiling function

To receive the classical chain-ladder model or the CRM, we need to aggregate individual claims $(\mathcal{C}_\ell)_\ell$ appropriately, where $\ell \geq 1$ labels all claims that have occurred before the cutoff date, i.e. with $T_\ell \leq \tau$. We first define a function that extracts the given period from a calendar date. In this paper, this period is either a calendar year, a calendar month or a calendar week, see Section 2.2. We denote this function by $\lceil \cdot \rceil$, and we call it the *time ceiling function*. Its explicit definition will depend on the time scale chosen (in all three cases we use the same notation but indicate the explicit choice in the surrounding text). We initialize our starting point by $\tau_0 = \text{yyyy}_0/\text{mm}_0/\text{dd}_0$, assuming that $\text{mm}_0/\text{dd}_0 = 01/01$ is a 1st of January.[2] Depending on the time scale chosen, the time ceiling function $\lceil \cdot \rceil$ considers the time difference between the dates yyyy/mm/dd and τ_0 in the chosen unit, i.e.

$$\text{yyyy/mm/dd} \;\mapsto\; \lceil \text{yyyy/mm/dd} \rceil = \begin{cases} \text{yyyy} - \text{yyyy}_0 & \text{yearly units,} \\ 12(\text{yyyy} - \text{yyyy}_0) + \text{mm} - \text{mm}_0 & \text{monthly units,} \\ \text{yyyy/mm/dd} - \tau_0 & \text{as weekly difference,} \end{cases} \tag{2.1}$$

we did not explicitly define the weekly case because the definition is a bit cumbersome, but we simply label consecutively the calendar weeks defining τ_0 to be week 0. Note that in all three cases we have initialization $\lceil \tau_0 \rceil = 0$, and $\lceil \tau \rceil$ denotes the cutoff date in the corresponding unit. Choose a fixed time scale (either years, months or weeks) and call the resulting time intervals *periods*. For occurrence periods $0 \leq i \leq \lceil \tau \rceil$ and reporting delays $j \geq 0$ we define the *number of reported claims* in accident period i with reporting delay j by

$$N_{i,j} = \sum_\ell \mathbb{1}_{\{\lceil T_\ell \rceil = i,\ \lceil U_\ell \rceil - \lceil T_\ell \rceil = j\}}. \tag{2.2}$$

In Wahl et al. [30] this quantity is studied on an annual scale and it is assumed to be over-dispersed Poisson (ODP) distributed; we come back to this assumption in Section 3.2, below. The reported claims at the cutoff date τ give the so-called *upper triangle* generating information

$$\mathcal{N}_\tau = \sigma\left\{N_{i,j};\ 0 \leq i \leq \lceil \tau \rceil,\ j \geq 0,\ i + j \leq \lceil \tau \rceil\right\}. \tag{2.3}$$

[1] Setting the cutoff date to a calendar year end makes some notation simpler, however, this choice is w.l.o.g.

[2] Setting the starting point to a calendar year beginning simplifies our notation, but this choice is w.l.o.g.

The classical chain-ladder prediction for claim counts is based on this information $\mathcal{N}_\tau$. The next quantity considered in Lindholm–Verrall [16] is the number of payments for given accident period i, reporting delay j and settlement delay $k \geq 0$ (after reporting $\lceil U \rceil$)

$$N^{\text{paid}}_{i,j,k} = \sum_\ell \left(\mathbb{1}_{\{\lceil T_\ell \rceil = i,\ \lceil U_\ell \rceil - \lceil T_\ell \rceil = j\}}\ \min\left\{ 1, \sum_{s \geq 0} \mathcal{Y}_{\ell,s} \mathbb{1}_{\{\lceil U_\ell + s \rceil - \lceil U_\ell \rceil = k\}} \right\} \right). \tag{2.4}$$

Wahl et al. [30] make a conditional Poisson assumption for $N^{\text{paid}}_{i,j,k}$ and Lindholm–Verrall [16] choose a distribution-free approach, see their Assumption (A3).
Similar to the number of payments $N^{\text{paid}}_{i,j,k}$, one defines the payments of a given accident period i, reporting delay j and settlement delay $k \geq 0$ (after reporting)

$$X_{i,j,k} = \sum_\ell \left(\mathbb{1}_{\{\lceil T_\ell \rceil = i,\ \lceil U_\ell \rceil - \lceil T_\ell \rceil = j\}}\ \sum_{s \geq 0} Y_{\ell,s} \mathbb{1}_{\{\lceil U_\ell + s \rceil - \lceil U_\ell \rceil = k\}} \right).$$

Finally, the payments for accident period i and payment delay j are given by

$$X_{i,j} = \sum_{0 \leq k \leq j} X_{i,j-k,k}.$$

This provides the following information at cutoff date τ, which is often used in claims reserving,

$$\begin{aligned} \mathcal{X}_\tau &= \sigma\left\{ X_{i,j};\ 0 \leq i \leq \lceil \tau \rceil,\ j \geq 0,\ i + j \leq \lceil \tau \rceil \right\}, \qquad \text{and} \\ \mathcal{F}_\tau &= \mathcal{N}_\tau \vee \mathcal{X}_\tau. \end{aligned}$$

The classical chain-ladder method for (cumulative) payments as well as many other claims reserving methods use information $\mathcal{X}_\tau$. Information $\mathcal{F}_\tau$ is used, for instance, in the double chain ladder model of Martínez-Miranda et al. [19, 20]. Typically, these classical methods are applied on an annual scale (to capture an entire annual seasonal cycle in an observation).

3 IBNR claim counts modeling

We start by discussing the modeling of claim counts $N_{i,j}$ which involves the modeling of accident dates T_ℓ, reporting dates U_ℓ and static claim features ϑ_ℓ of individual claims $\mathcal{C}_\ell$, $\ell \geq 1$, the lower index ℓ always indicates that we consider a specific individual claim.

3.1 Descriptive data analysis of claim counts

The available data comes from two different lines of business (LoBs).[3] Each claim $\mathcal{C}_\ell$ is established with an accident date T_ℓ, a reporting date U_ℓ, and a claim code illustrating the type of claim (having 9 different levels). Moreover, for each reported claim $\mathcal{C}_\ell$ we know the settlement processes of payments $(Y_{\ell,s})_s$ and of the claim status $(O_{\ell,s})_s$ up to the cutoff date τ, i.e. for $0 \leq s \leq \tau - U_\ell$. At the cutoff date $\tau = 2017/12/31$ we have observed $n = 34'260$ reported claims with accident dates after $\tau_0 = 2009/01/01$, i.e. with $\tau_0 \leq T_\ell \leq U_\ell \leq \tau$.

[3] For anonymization reasons we only illustrate aggregated data (over both LoBs), and all claim payments have been scaled with a constant.

Listing 1: Available static individual claims information of reported claims.

```
1 'data.frame':   34260 obs. of  5 variables:
2  $ Id       : int  2 4 5 6 8 9 13 14 16 17 ...
3  $ LoB      : Factor w/ 2 levels "LoB1","LoB2": 2 1 2 1 1 1 2 2 1 2 ...
4  $ code     : Factor w/ 9 levels "1","2","3","4",..: 7 4 7 4 1 1 7 7 4 7 ...
5  $ acc_date: num  20090102 20090104 20090105 20090102 20090105 ...
6  $ rep_date: num  20090107 20090107 20090108 20090108 20090108 ...
```

An excerpt of these reported claims (without their settlement processes) is illustrated in Listing 1. This data is sufficient to model claim occurrences and their reporting delays: We have accident dates T_ℓ on line 5 (in daily units) and reporting dates U_ℓ on line 6 (in daily units); their differences provide the reporting delays $W_\ell \geq 0$. Moreover, LoB_ℓ on line 3 and $(\text{claim code})_\ell$ on line 4 provide for our data set the static feature information $\vartheta_\ell = (\text{LoB}_\ell, (\text{claim code})_\ell)$ of reported claims $\ell = 1, \ldots, n$ at the cutoff date τ. We assume here, that ϑ_ℓ is fully known at the reporting date U_ℓ. This is the reason for calling it a static feature.

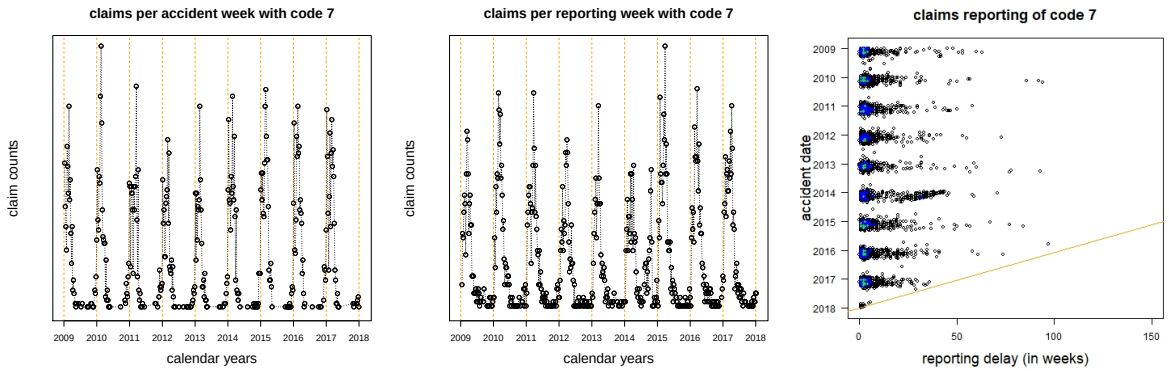

Figure 1: Observed claim counts per accident week (lhs), per reporting week (middle) and reported triangle (rhs) for claim code 7.

We analyze this reporting data for different claim codes (see line 4 of Listing 1). Figure 1 (lhs) illustrates the claim counts of claim code 7 aggregated over the accident dates T_ℓ on a weekly time scale (and reported up to the cutoff date τ). In this graph IBNR claims (with $U_\ell > \tau$) are missing, therefore the values in Figure 1 (lhs) may still increase, in particular, close to the cutoff date. The crucial observation is that we have a strong seasonal pattern for claims with code 7 (with mid year lows). Figure 1 (middle) shows exactly the same claims, but the x-axis has been changed to reporting dates U_ℓ on a weekly time scale. In general, the claims move slightly to the right compared to the accident date plot because of positive reporting delays $W_\ell = U_\ell - T_\ell \geq 0$. However, the main structure of the seasonal pattern remains. Finally, Figure 1 (rhs) gives the resulting scatter plot with accident dates T_ℓ on the y-axis and reporting delays W_ℓ on the x-axis (in weekly units). The orange line shows the cutoff date τ, which separates the so-called upper triangle (observations) from the lower triangle (to be predicted). We observe the same seasonal pattern as in the other two plots, and we also observe that all reported claims have a reporting delay W_ℓ of less than 100 weeks. Figure 21 in the appendix shows the corresponding scatter plots of all other claim codes. Some of them have a stronger seasonal pattern, but none of them

shows reporting delays bigger than 100 weeks. This justifies considering a maximal reporting delay in Assumptions 2.1.

3.2 Predicting the number of IBNR claims

Our first goal is to predict the number of claims which will be reported after the cutoff date τ, that is, satisfying $\tau_0 \leq T_\ell \leq \tau < U_\ell$. We choose a weekly time grid for this prediction problem, with $0 \leq i \leq I$ denoting accident weeks and with $0 \leq j \leq J$ denoting reporting delay weeks, we also refer to Section 2.2. The reason for choosing a weekly time grid is that this provides a good balance between computational efficiency and (seasonal) accuracy. Choosing a finer grid size is computationally more demanding without any benefits in better accuracy. Going over to a monthly time grid is not advisable at this stage because claim occurrences are not sufficiently homogeneous across an entire calendar month; note that there are sharp seasonal increases and decreases in claim counts, see Figure 1, which may lead to too much heterogeneity on a monthly time grid. Moreover, a monthly time grid is not equidistant because different calendar months have different numbers of days. Equidistance is not crucial, but it is still a nice property to have since it implies equal time exposures.
In our data we have $I = 470$ accident weeks from $\tau_0 = 2009/01/01$ to $\tau = 2017/12/31$, and the maximal reporting delay is $J = 100$ weeks, the latter is justified by Figure 21.
In the remainder of this section we choose the time ceiling function $\lceil \cdot \rceil$ in (2.1) on the weekly scale. In view of (2.2) this allows us to define the claim counts on that weekly scale

$$N_{i,j|\vartheta} = \sum_\ell \mathbb{1}_{\{\lceil T_\ell \rceil = i,\ \lceil U_\ell \rceil - \lceil T_\ell \rceil = j\}} \mathbb{1}_{\{\vartheta_\ell = \vartheta\}},$$

i.e. we aggregate on a weekly time scale all claims that have the same static feature $\vartheta = (\text{LoB}, \text{claim code})$. The resulting observations at the cutoff date τ are denoted by $\mathcal{N}_{\tau|\vartheta}$, see also (2.3). The upper trapezoid in Figure 1 (rhs) illustrates the observations at the cutoff date τ (above the orange line) for claim code 7.

Assumptions 3.1 *The claim arrival process $(T_\ell, W_\ell)_{\ell \geq 1}$ of each static feature ϑ is described by an independent two-dimensional inhomogeneous Poisson point process on $[\tau_0, \tau] \times [0, 840]$ (in daily units).*

We describe the two-dimensional claim arrival processes by independent (for different ϑ's) inhomogeneous Poisson point processes with accident dates $T_\ell \in [\tau_0, \tau]$ and reporting delays $W_\ell \in [0, 840]$, thus, we assume a maximal reporting delay of 120 weeks ($m = 7 \cdot 120 = 840$ days, see Assumptions 2.1). Assumption 3.1 is in line with Norberg [21, 22] assuming that the claims can be described by marked Poisson point processes with independent marks.
We assume that the inhomogeneous Poisson point processes of Assumption 3.1 can be described by a (measurable and bounded) intensity function

$$\lambda : [\tau_0, \tau] \times [0, 840] \times [0, 1] \times \{\vartheta\} \to \mathbb{R}_+, \qquad (t, w, v(t), \vartheta) \mapsto \lambda(t, w, v(t), \vartheta) > 0,$$

where $t \mapsto v = v(t) \in [0, 1]$ models the occurrence week *within* the calendar year using a linear function with initialization $v(\text{week } 1) = 0$ and $v(\text{week } 53) = 1$. This variable $v(t)$ will allow us to model annual seasonal patterns which are motivated by Figure 1.

Remark.

- We could drop component $v(t)$ in the notation of definition λ, because this component is fully determined by the accident date component t. We prefer the notation above because it explicitly highlights that we model an annual seasonal feature. This hint is also going to be given to the network regression function used below.

- At first sight it would be more natural to model $t \mapsto v(t)$ by a sine function for an annual seasonal pattern. However, this would require that the modes of all ϑ's are in the same month/week. In our example this is not the case and, indeed, replacing $v(t)$ by a sine function has led to worse predictive performance on our data.

Assumption 3.1 implies that $(N_{i,j|\vartheta})_{i,j,\vartheta}$ are independent and Poisson distributed with expected numbers of claims

$$\Lambda(i,j,\vartheta) \stackrel{\text{def.}}{=} \mathbb{E}\left[N_{i,j|\vartheta}\right] = \int_{\{\lceil t\rceil=i,\lceil t+w\rceil-\lceil t\rceil=j\}} \lambda(t,w,v(t),\vartheta)\; dt\; dw. \tag{3.1}$$

We estimate regression function (3.1) with a 1st neural network based on features $(i,j,v(i),\vartheta)$, i.e. we give a hint to the neural network that it should search for an annual seasonal pattern by including $v(i)$ in the feature information. Thus, we model the following function by a neural network

$$(i,j,v(i),\vartheta) \;\mapsto\; \Lambda(i,j,\vartheta) = \Lambda(i,j,v(i),\vartheta). \tag{3.2}$$

To this end, we use the same approach and terminology as described in Wüthrich [33]. As neural network architecture we choose a network of depth 3 having $(20,15,10)$ hidden neurons in the hidden layers and with hyperbolic tangent activation function. The continuous feature components are pre-processed using the MinMaxScaler, the feature LoB is binary in our example and is therefore encoded as 0-1, and the categorical feature claim code is modeled by a two-dimensional embedding layer. The explicit network architecture used and further model specifications are given in the R code in Listing 2 in the appendix.
We use the nadam version of the stochastic gradient descent algorithm to fit this network to the data $\bigvee_{\vartheta} \mathcal{N}_{\tau|\vartheta}$ in the upper triangle. As objective function we choose the Poisson deviance loss given by

$$\mathcal{L}\Big(\bigvee_{\vartheta} \mathcal{N}_{\tau|\vartheta}\Big) = \sum_{\vartheta} \sum_{i+j\le\lceil\tau\rceil} N_{i,j|\vartheta} \left[\frac{\Lambda(i,j,v(i),\vartheta)}{N_{i,j|\vartheta}} - 1 - \log\left(\frac{\Lambda(i,j,v(i),\vartheta)}{N_{i,j|\vartheta}}\right)\right].$$

This will provide us with estimated network function $(i,j,v(i),\vartheta) \;\mapsto\; \widehat{\Lambda}(i,j,v(i),\vartheta)$.

Remark.

- If we assumed a cross-classified structure $\Lambda(i,j,v(i),\vartheta) = \alpha_i\beta_j$, we would obtain the Poisson version of the chain-ladder method. The latter is solved within generalized linear models (GLMs). This GLM may miss important interactions and seasonal patterns (because we only have multiplicative interactions in the cross-classified case). For this reason, we prefer the neural network approach.

- Similarly to the chain-ladder method we do not include any exposure measure. We do this because of lack of information. If we would have exposure information it could be included in the feature information, and it would lead to a Bornhuetter-Ferguson type of model.

- The network $(i, j, v(i), \vartheta) \mapsto \widehat{\Lambda}(i, j, v(i), \vartheta)$ is fitted on the upper triangle $i + j \leq \lceil \tau \rceil$, and its extrapolation to the lower triangle $i + j > \lceil \tau \rceil$ gives us the predicted numbers $\widehat{\Lambda}(i, j, v(i), \vartheta)$ of IBNR claims. Appropriateness of this extrapolation is difficult to judge without any additional information. For this reason we benchmark our approach with the classical chain-ladder method that is known to be robust in fairly general situations.

Our neural network is trained on the data $\bigvee_{\vartheta} \mathcal{N}_{\tau|\vartheta}$ in the upper triangle, using 20% of the data as validation data to track over-fitting. We illustrate the training performance of this

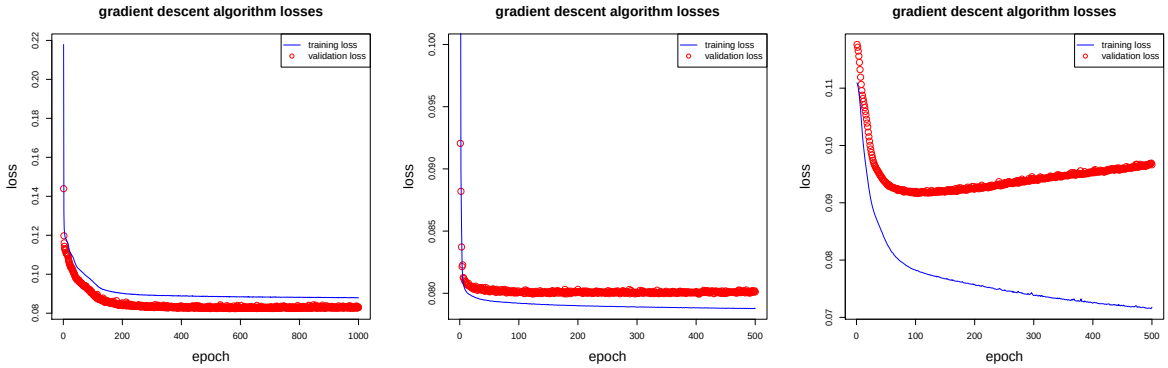

Figure 2: Training and validation losses of (lhs) the claim counts model described by (3.1), (middle) the payment indicator process network (4.2), and (rhs) the recovery indicator process.

fitting procedure in Figure 2 (lhs). Using a callback during training we retrieve the parameters with the smallest validation loss. To receive an unbiased model in the upper triangle we apply to this network the bias regularization step described in Section 3.4 of [33], the code for this regularization step is given in Listing 3 in the appendix. This provides in the upper triangle in average 0.077277 claims per accident week i and reporting delay week j with $i + j \leq \lceil \tau \rceil$, see Table 1, and an in-sample loss of 0.1510.[4]

	run time	# param.	in-sample loss	average per (i, j)	number of IBNR claims
neural network	427s	644	0.1510	7.7277%	402
ODP chain-ladder method	–	11	–	7.7277%	458

Table 1: Claim counts: run time of calibration, number of model parameters, in-sample loss, estimated number of claims on the weekly time grid in the upper triangle, and predicted number of IBNR claims.

This estimated model for (3.1)-(3.2) is then extrapolated to the lower triangle, and the results are illustrated in Figure 3 for claim code 7. The left-hand side and the middle plots show the estimated marginal counts for accident weeks and reporting weeks, the red dotted lines exclude IBNR claims (estimated model in upper triangle), and the cyan and green lines include IBNR claims (prediction in lower triangle). This is clearly visible at the right end of the plots

[4]Run times are measured on a personal laptop with CPU @ 2.50GHz (4 CPUs) with 16GB RAM (which is a comparably modest computational power).

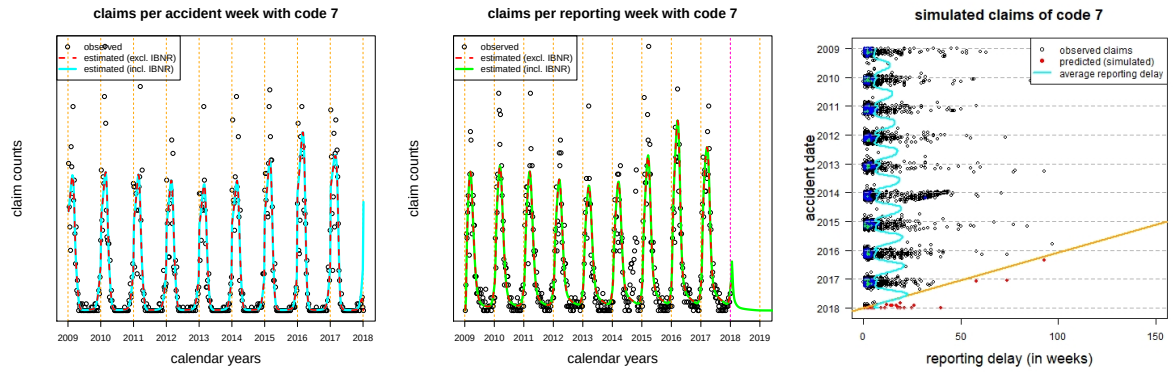

Figure 3: Observed and predicted claims per accident week (lhs), per reporting week (middle) and reported (black) and predicted (red) triangles (rhs) for claim code 7.

by a steeply increasing cyan curve for accident dates, and by the light-green extrapolation for reporting dates. Using this estimated model we run one simulation for the lower triangle (using the corresponding features) resulting in the red dots (claims) of Figure 3 (rhs). This extrapolates the structure in the upper triangle to the lower one. The cyan line in Figure 3 (rhs) gives the estimated average reporting delay per accident week. We observe a strong seasonal pattern in these average reporting delays, however, this pattern has to be considered with care because there are almost no claims during mid-years. These estimated average reporting delays can be interpreted in a similar way as the operational times considered in Bischofberger et al. [5].
In the appendix we show the resulting scatter plots for all claim codes in Figure 21. We note that most claim codes have a seasonal pattern. From Figure 3 we also observe that the Poisson model cannot fully explain extremes (which may be caused by external factors). This could be a motivation to study Cox processes, see Badescu et al. [3], here we refrain from doing so because IBNR claims play a marginal role in our reserves calculation (as we will see below).
Aggregating the predicted numbers $\widehat{\Lambda}(i,j,v(i),\vartheta)$ over the lower triangle $i+j>\lceil\tau\rceil$ we obtain a prediction of 402 IBNR claims in the lower triangle. We compare this prediction to the one obtained by the chain-ladder method based on yearly observations (2.3).[5] From the chain-ladder method we receive 458 IBNR claims which is bigger. Performing the over-dispersed Poisson (ODP) bootstrap described in England–Verrall [10] we receive a rooted mean square error of prediction (RMSEP) of 49 (this is based on an estimated over-dispersion of 3.89). Thus, the neural network and the chain-ladder predictions differ by roughly 1.1 RMSEPs which almost explains the differences between the two methods.

4 Payment indicator and claim status modeling of RBNS claims

In this and the next Sections 4 and 5 we model the settlement processes of reported claims. This includes the payment indicator and claim status process $(\mathcal{Y}_s, O_s)_{s\geq 0}$ of each individual claim (Section 4) as well as the payment process $(Y_s)_{s\geq 0}$ in case of payments (Section 5). The data for this modeling part is always available at the end of a calendar month (end-of-month

[5]We perform the chain-ladder method on all data simultaneously. If we had a bigger portfolio, we would partition it into homogeneous sub-portfolios.

data), thus, the granularity of these observed settlement processes is more coarse than the claim occurrence data. For this reason, we move to a monthly grid size in this and the next section. To highlight this we change the time notation from the daily $s \geq 0$ to a monthly $k \in \mathbb{N}_0$. The time ceiling function $\lceil \cdot \rceil$ is understood on the monthly time scale in this and the next section. The claim status process $(O_k)_{k \in \mathbb{N}_0}$ always denotes end-of-month statuses, and the monthly payments and payment indicators are for $k \in \mathbb{N}_0$ defined as follows

$$Y_k = \sum_{s \geq 0} Y_s \mathbb{1}_{\{\lceil U+s \rceil - \lceil U \rceil = k\}} \qquad \text{and} \qquad \mathcal{Y}_k = \mathbb{1}_{\{Y_k \neq 0\}}. \tag{4.1}$$

This is just a complicated way of saying that Y_k are the total individual claim payments within the k-th settlement month for a given claim; and k is called the settlement delay (after reporting).

4.1 Description of our RBNS claims modeling approach

Kuo [14] performs a simultaneous multi-period forecast after the cutoff date τ for the entire run-off $(Y_k)_{k > \lceil \tau \rceil - \lceil U \rceil}$ of individual reported claims. This multi-period forecast is based on a long short-term memory (LSTM) network model. On a yearly time grid this is computationally feasible, on our finer monthly time grid we prefer a model that performs successive one-period-ahead forecasts. This allows us to keep tractability, interpretability as well as the possibility to intervene the process during different modeling steps of the settlement process.
Successive one-period-ahead forecasts means that we recursively extrapolate the settlement process into the lower triangle. This seems simple, but it requires special attention. The observations $(Y_k, \mathcal{Y}_k, O_k)_{k \leq \lceil \tau \rceil - \lceil U \rceil}$ in the *upper triangle* are on a random variable level, and predictions in the *lower triangle* typically are on the level of estimated expected values $(\widehat{Y}_k, \widehat{\mathcal{Y}}_k, \widehat{O}_k) = (\widehat{\mathbb{E}}[Y_k], \widehat{\mathbb{E}}[\mathcal{Y}_k], \widehat{\mathbb{E}}[O_k])$, for $k > \lceil \tau \rceil - \lceil U \rceil$. Thus, typically, upper and lower triangles are based on variables of a different quality, for instance, for the claim status we have $O_k \in \{0, 1\}$ and $\widehat{O}_k = \widehat{\mathbb{E}}[O_k] \in (0, 1)$. This poses some challenges in path-wise one-step-ahead predictions because it requires that we simulate observations into the lower triangle which serve as covariates for the next step of the one-step-ahead forecast. Most papers in the literature do not make this clear distinction, Gabrielli [11] being an exception making a careful distinction between random variables and expected values internally in his network architecture.
Monte Carlo extrapolation of the indicator process $(\mathcal{Y}_k, O_k)_{k \in \mathbb{N}_0}$ is not difficult because it involves a stochastic process on a (finite) categorical state space. Monte Carlo extrapolation of the claim payment process $(Y_k)_{k \in \mathbb{N}_0}$ poses quite a challenge because there is no simple off-the-shelf model that is sufficiently accurate in simulating claim payments, i.e. real world payments are by far more complex than can be captured by classical models (also having reasonable run times). For this reason, we divide this modeling into two parts: The first part, described here in Section 4, concerns modeling of the indicator process $(\mathcal{Y}_k, O_k)_{k \in \mathbb{N}_0}$. This modeling part will not rely on any claim payment size information and can, therefore, easily be extrapolated to the lower triangle by Monte Carlo simulation. The second part, described in Section 5 below, concerns modeling of the claim payment process $(Y_k)_{k \in \mathbb{N}_0}$, conditionally on knowing the indicator process. Since claims reserves are *expected values* we are mainly interested in appropriate *expected claim payments*. Therefore, this modeling step is designed such that we do not need any simulation, but we try to directly assess expected future payments as good as possible from a gamma distribution model.

Disclosure. Obviously, the gamma distribution is not sufficiently accurate for payment size simulation. Nevertheless, we use the gamma model, the reason being that it allows for direct (not involving hyperparameters) maximum likelihood estimation (MLE) of a (single) canonical parameter that directly provides the mean of the model. The latter property makes the gamma model superior to many other models like the log-normal model if we are only interested in expected values. Typically, the gamma distribution is too light-tailed in real applications, for this reason, the gamma model will slightly over-fit to large observations. However, we can live with this deficiency as there is no better alternative that is computationally sufficiently tractable.

4.2 Stylized facts about payments and closings

There has been a debate in the reserving literature about a one-payment assumption for general insurance claims, see for instance Assumption (D2) in [19]. We empirically test this assumption by plotting the pattern of the observed relative number of payments on a monthly scale, set

$$\widehat{\pi}_k = \frac{\sum_{\ell=1}^n \mathcal{Y}_{\ell,k} \mathbb{1}_{\{\lceil U_\ell \rceil + k \le \lceil \tau \rceil\}}}{\sum_{\ell=1}^n \mathbb{1}_{\{\lceil U_\ell \rceil + k \le \lceil \tau \rceil\}}} \in [0,1], \qquad \text{for } k \ge 0,$$

where $\mathcal{Y}_{\ell,k} \in \{0,1\}$ is the monthly payment indicator of claim ℓ in settlement month k, see (4.1). Figure 4 gives the cumulative empirical payment pattern $(\widehat{\Pi}_k = \sum_{j=0}^k \widehat{\pi}_j)_{k\ge 0}$ for the different claim codes 1 to 9. We observe that claim codes 3,4,6,9 have on average less than one monthly payment, and claim codes 1,2,5,7,8 have on average significantly more than one monthly payment, claim code 7 having on average close to 4 monthly payments per claim. Hence, we can clearly reject the one-payment assumption for our data.

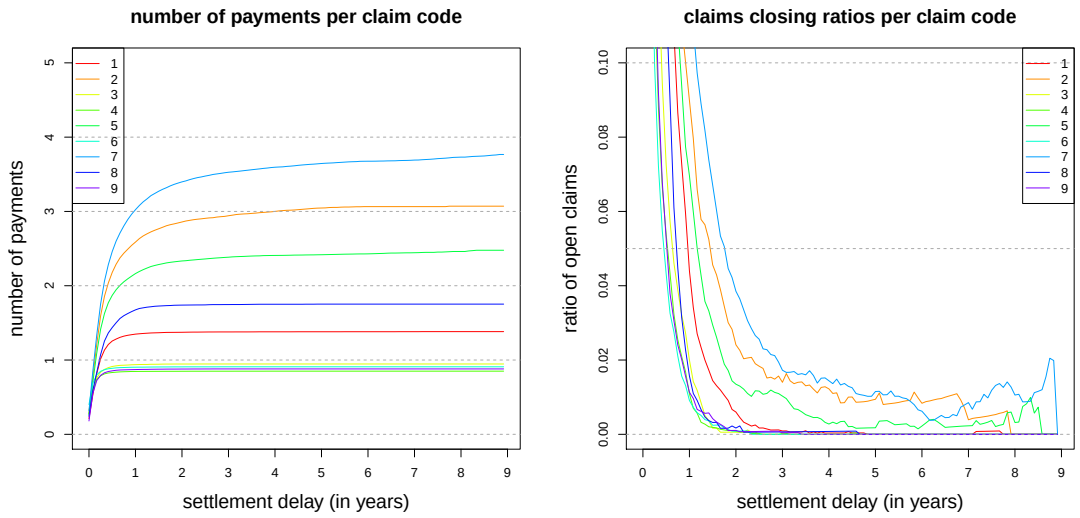

Figure 4: (lhs) Cumulative empirical payment pattern $(\widehat{\Pi}_k)_{k\ge 0}$, and (rhs) claim status ratios $(\widehat{\varpi}_k)_{k\ge 0}$ for the different claim codes.

Next, we plot the empirical ratio of open claims for settlement month $k \ge 0$ given by

$$\widehat{\varpi}_k = \frac{\sum_{\ell=1}^n O_{\ell,k} \mathbb{1}_{\{\lceil U_\ell \rceil + k \le \lceil \tau \rceil\}}}{\sum_{\ell=1}^n \mathbb{1}_{\{\lceil U_\ell \rceil + k \le \lceil \tau \rceil\}}} \in [0,1].$$

We observe a fast settlement of the claims with claim codes 1,3,4,6,8,9, but we still expect open claims for claim codes 2,5,7 after 10 years of settlement delay, see Figure 4 (rhs). Henceforth, the settlement process is expected to be long-tailed for some claim types.

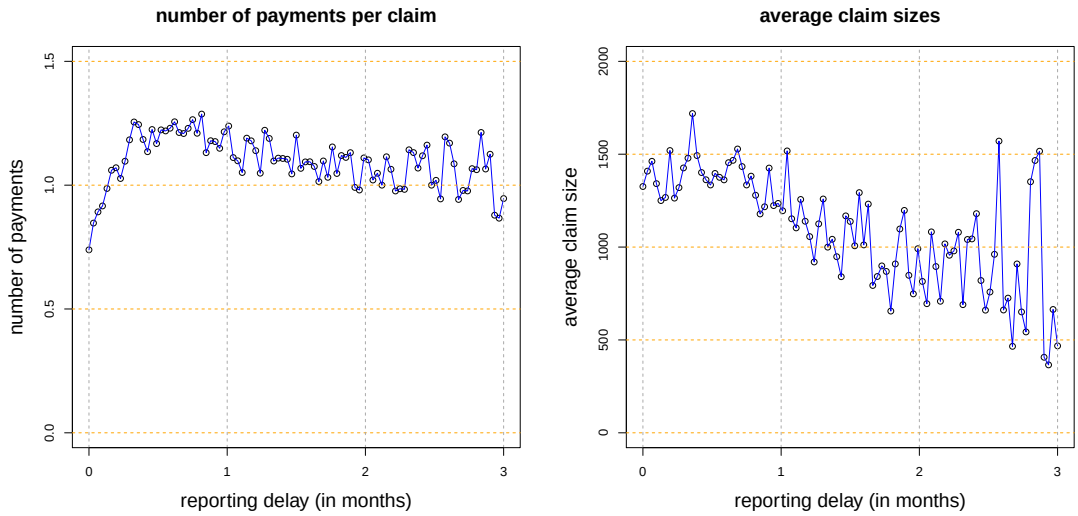

Figure 5: (lhs) Empirical number of payments per claim as a function of reporting delay (in months); (rhs) average claim sizes as a function of reporting delay over all claim codes.

In Figure 5 we plot the empirical number of payments and the average claim sizes as a function of reporting delay. Note that we calculate these numbers on a fully developed rectangle to receive consistent results. Therefore, the plots in Figure 5 do not reflect fully settled claims because they may still develop beyond these rectangles. However, these plots clearly illustrate that the payments depend on reporting delays, thus, an independence assumption between reporting delays and payments is clearly not fulfilled for our data.

4.3 Indicator process modeling

We are now on a monthly scale and model the indicator process $(\mathcal{Y}_k, O_k)_{k\geq 1}$ of reported claims.

4.3.1 Neural network model

First, we modify our two-dimensional Bernoulli process to a one-dimensional categorical process by defining for $k \geq 1$ the response variable

$$\mathcal{I}_k = \mathcal{Y}_k + 2 \cdot O_k \in \{0, 1, 2, 3\}.$$

We model $\mathcal{I}_k$ with a categorical distribution having (normalized) probability vector

$$\boldsymbol{x}_k \;\mapsto\; \boldsymbol{p}(\boldsymbol{x}_k) = (p_0(\boldsymbol{x}_k), p_1(\boldsymbol{x}_k), p_2(\boldsymbol{x}_k), p_3(\boldsymbol{x}_k))^\top \;\in \mathbb{R}^4_+, \tag{4.2}$$

for given feature information $\boldsymbol{x}_k$. We first describe the choice of the feature information $\boldsymbol{x}_k$. Similarly to (3.1), it contains the reporting delay $W = U - T \geq 0$, the week within the calendar year $v(T) \in [0, 1]$ of claim occurrence (for seasonal pattern modeling) and the static feature

information $\vartheta = (\text{LoB}, \text{claim code})$, see Section 2.1. We do not include the accident date T information for stability reasons. It has turned out in experiments that the inclusion of T may lead to an undesired extrapolation of trends, which might be problematic if it affects the extrapolation of the upper right triangle (over a long time period).[6]

Additionally, we are going to include past trajectory $(\mathcal{Y}_l, O_l)_{0\le l\le k-1}$ information in the feature. Since claim re-openings are scarce (this is verified in Figure 7, below) we are only going to include the last claim status O_{k-1}, this does not cause a major loss of information and it reduces the dimension of the problem substantially; we emphasize that re-openings are still modeled but we make a Markov assumption in this variable. Next, we include one year of past monthly payment indicator observations $(\mathcal{Y}_l)_{k-12\le l\le k-1}$; again to reduce the dimension of the problem we do not provide the whole past payment indicator process. To still keep track of multiple monthly payments over the entire past we add a variable $\mathfrak{y}_{k-1} = \sum_{l=0}^{k-1} \mathcal{Y}_l$. Finally, we would like to use regression function (4.2) simultaneously for all settlement periods $k \ge 1$, therefore, we have to mask the undefined part $(\mathcal{Y}_l)_{l<0}$ in $(\mathcal{Y}_l)_{k-12\le l\le k-1}$ for small k; we do this by setting this value identically equal to -1, and we add k to the feature information $\boldsymbol{x}_k$ to highlight to the network which settlement month we are considering (this is beneficial for model calibration because we guide the network in the right direction for finding model structure). Summarizing, we choose feature information, note that $\vartheta = (\text{LoB}, \text{claim code}) \in \{0,1\}\times\{\text{claim code}\} \subset \mathbb{R}\times\{\text{claim code}\}$,

$$\boldsymbol{x}_k = (W, v(T), (\mathcal{Y}_l)_{k-12\le l\le k-1}, \mathfrak{y}_{k-1}, O_{k-1}, k, \vartheta)^\top \in \mathbb{R}^{18} \times \{\text{claim code}\}. \tag{4.3}$$

We model the probability vector $\boldsymbol{x}_k \mapsto \boldsymbol{p}(\boldsymbol{x}_k)$ given in (4.2) by a 2nd neural network of depth 3 having $(20, 15, 10)$ hidden neurons in the hidden layers. The only things that we are changing compared to Listing 2 are (1) the input dimension on line 4 from 4 to 18 because we have 18 real-valued feature components in $\boldsymbol{x}_k$, (2) the output function on line 14 has now 4 units and we change the output activation to the softmax function because we have categorical responses with 4 classes, and (3) the loss function on line 21 is changed to categorical cross-entropy.

We pre-process our data so that we obtain the observations $(\mathcal{I}_{\ell,k}, \boldsymbol{x}_{\ell,k})$ of all claims ℓ for which we have observed at least two settlement periods up to the cutoff date. This results in 1'753'173 observations $(\mathcal{I}_{\ell,k}, \boldsymbol{x}_{\ell,k})$ that can be used to train the 2nd network. We perform the same covariate pre-processing and training strategy as above, using 20% of the data as validation data to track over-fitting. The resulting network with the smallest validation loss is chosen, and we apply the bias regularization step using the code in Listing 4.

Figure 2 (middle) shows the training process of this indicator process network, and the corresponding results after regularization are provided in Table 2. We benchmark the results with a classical categorical GLM, with appropriately pre-processed features. We observe from Table 2 that the GLM comes close the network, but it is not quite as good as the network (note that the network has been early stopped to prevent from over-fitting). This justifies the use of the (more complex) neural network approach.

[6] This undesired extrapolation of trends may best be illustrated by giving an explicit example. If we consider settlement month $k = 1$, inclusion of T in feature $\boldsymbol{x}_1$ may model inflation of payments from 2009 to 2017. Inclusion of T in feature $\boldsymbol{x}_k$ for settlement month $k = 9 \cdot 12 = 108$ would then extrapolate this inflation to years 2018-2026. It is clear that this might be problematic, in particular, because claim payments in early settlement periods (small k) may be of a different nature than payments for later settlement periods (large k), since the former rather relate to simple claims and the latter to complex claims.

	run time	# param.	in-sample loss	estimated average probabilities $\widehat{\boldsymbol{p}}(\cdot)$
homogeneous model	–	3	0.5838	$(93.85\%, 1.39\%, 3.44\%, 1.32\%)$
categorical GLM	590s	111	0.1660	$(93.85\%, 1.39\%, 3.44\%, 1.32\%)$
neural network	1493s	957	0.1576	$(93.85\%, 1.39\%, 3.44\%, 1.32\%)$

Table 2: Indicator process: run time of calibration (including the bias regularization step), number of model parameters, in-sample losses, estimated average probabilities.

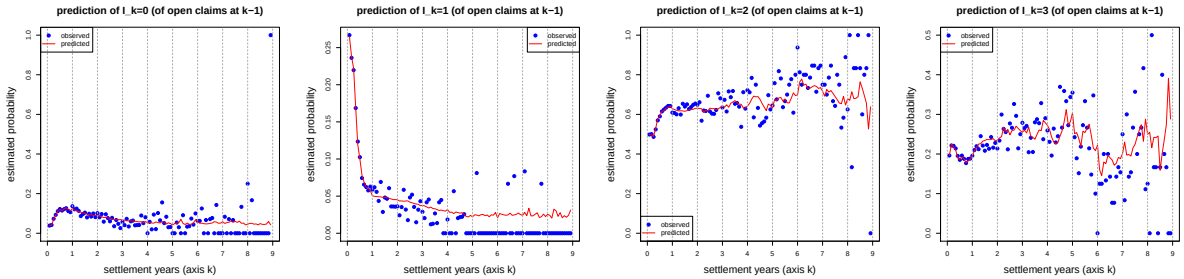

Figure 6: In-sample observed (blue) and predicted (red) one-period-ahead variables $\mathcal{I}_k = \{0, 1, 2, 3\}$ for open claims $O_{k-1} = 1$ one month before $k \geq 1$.

We start by analyzing the in-sample figures in the upper triangle. Figure 6 gives the observed and predicted one-period-ahead variables $\mathcal{I}_k$, $k \geq 1$, averaged over all open claims $O_{k-1} = 1$ at the end of month $k-1$: the 1st graph shows the average probability of an open claim being closed in settlement month k without payment (average 6%), the 2nd graph gives the average probability of a closing with a payment (average 5%), the 3rd graphs shows the probability that the claim remains open without a payment (average 67%), and the 4th graph gives the probability of remaining open with a payment in settlement month k (average 22%). The latter graph is quite wiggly for higher settlement months k, this is because we only have very few open claims in later periods (as underlying volume), and the estimated probabilities also depend on the other feature information in $\boldsymbol{x}_k$. Finally, Figure 7 gives the predicted re-openings $\mathcal{I}_k = \{1, 2, 3\}$ for closed claims $O_{k-1} = 0$ one month before. This number is less than 4% and even much smaller for late periods, thus, this confirms our scarcity statement about re-openings above.

4.3.2 Extrapolation of reported claims to the lower triangle

We are now in the situation where we can extrapolate the indicator process $(\mathcal{I}_k)_k$ of reported claims to the lower triangle. As described in Section 4.1, we do this by successive one-period-ahead Monte Carlo simulations (based on the estimated categorical probabilities (4.2)). Monte Carlo simulations are necessary to carry forward the right feature information $\boldsymbol{x}_{k-1}$ for $k > \lceil \tau \rceil - \lceil U \rceil$.

Figure 8 summarizes this successive one-period-ahead Monte Carlo extrapolation to the lower triangle averaged over all reported claims. The lines in this figure provide exactly the same graphs as in Figure 4, these are complemented by dots that reflect the Monte Carlo extrapolations to the lower triangle (out-of-sample), based on the estimated claim indicator process described by (4.2). With regards to payments, only claim code 2 shows some differences (Figure

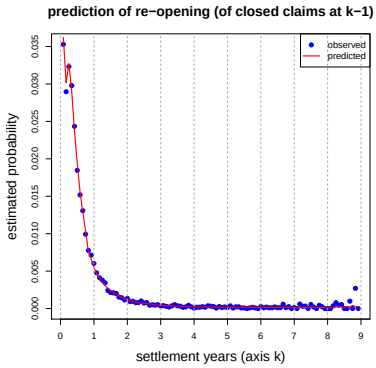

Figure 7: In-sample observed (blue) and predicted (red) re-openings $\mathcal{I}_k = \{1, 2, 3\}$ for closed claims $O_{k-1} = 0$ one month before $k \geq 1$.

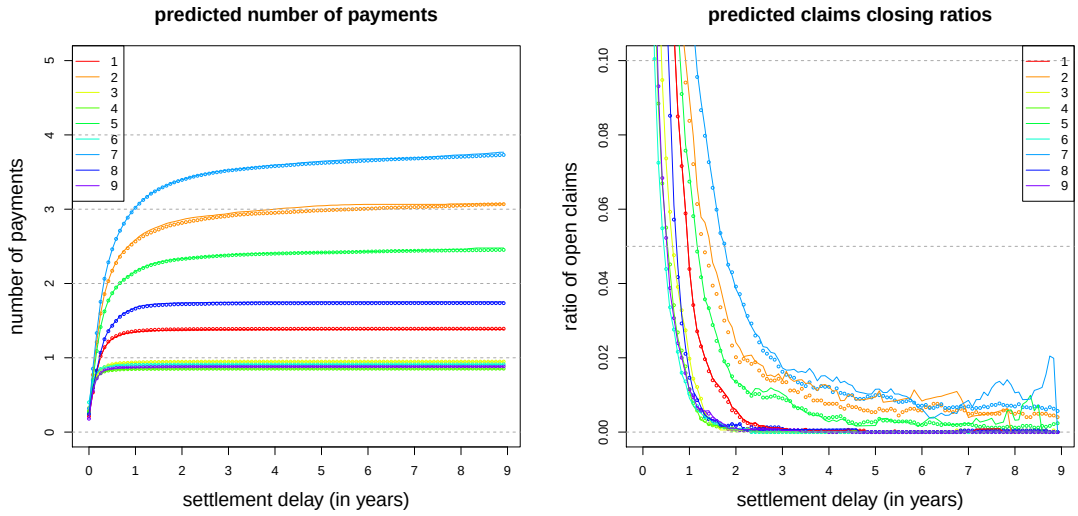

Figure 8: (lhs) Cumulative payment pattern $(\widehat{\Pi}_k)_{k\geq 0}$, and (rhs) claim status ratios $(\widehat{\varpi}_k)_{k\geq 0}$ for the different claim codes: the line gives the empirical values in the upper triangle (identical to Figure 4) and the dots the predicted values extrapolated to the lower triangle (out-of-sample).

4, lhs), which emphasizes a small trend in the process. The right-hand side of Figure 4 shows smoothed versions of the claim closings. The claims with codes 2, 5 and 7 have still 0.40%, 0.24% and 0.57%, respectively, of open claims after 10 years of settlement. We conclude that the estimated model extrapolates the indicator processes $(\mathcal{Y}_k, O_k)_{k\in\mathbb{N}_0}$ of reported claims from the upper to the lower triangle such that the basic payment and closing patterns per claim code are preserved, and it does not add an accounting year effect in our case. We emphasize that the statements made in this section are based on one single Monte Carlo extrapolation to the whole lower triangle. In Section 4.3.4 we will average over multiple such extrapolations to diminish Monte Carlo simulation uncertainty.

4.3.3 Analysis of the extrapolation of reported claims

Using the above Monte Carlo extrapolation for every reported claim $\ell = 1, \ldots, n = 34'260$ we now have a full trajectory $(\mathcal{Y}_{\ell,k}, O_{\ell,k})_{k=0,\ldots,d}$ over the maximal settlement delay of $d = 107$ months for all claims reported within $\tau_0 = 2009/01/01$ and $\tau = 2017/12/31$. The Monte Carlo simulation provides us with a total of 3'437 RBNS monthly claim payments in the lower triangle; this corresponds to the summation of the variables (2.4) over the lower triangle (and restricted to reported claims at the cutoff date τ).

# of payments	0	1	2	3	4	5	6	7	8	9	≥ 10
probability	9.9%	58.9%	16.1%	6.2%	2.6%	1.6%	1.1%	0.8%	0.6%	0.5%	1.7%

Table 3: Estimated probabilities of number of monthly payments per reported claim over $d+1 = 109$ settlement months, with an average of 1.7597 monthly payments per claim (based on one Monte Carlo extrapolation).

Using this extrapolation of the payment indicator process for reported claims we can estimate the number of monthly payments per reported claim. There is one claim that has predicted 54 monthly payments within the $d + 1 = 108$ settlement months, and in average we have 1.7597 monthly payments per claim. The summary statistics over all $n = 34'260$ reported claims is given in Table 3. Remarkable is that 9.9% of the claims do not suffer any payments, 58.9% of the claims have one monthly payment and 31.2% of the claims have more than one monthly payment, clearly contradicting the one-payment assumption, see Section 4.2.

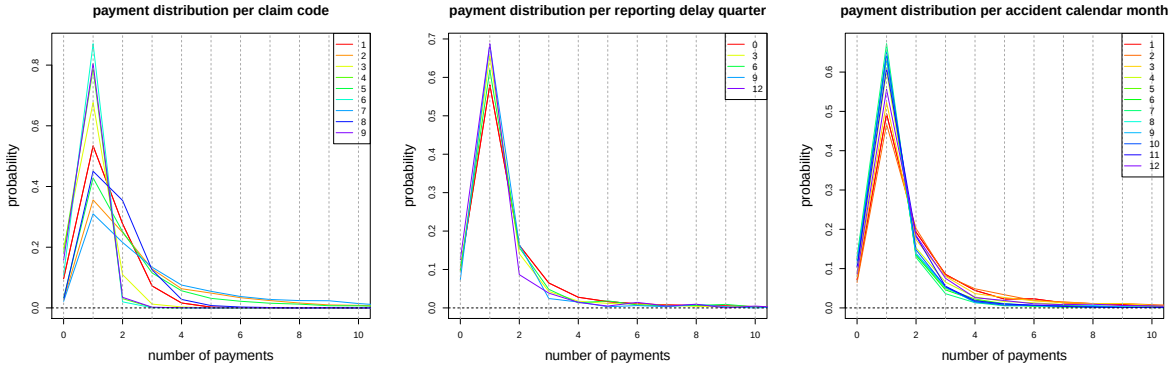

Figure 9: Estimated distributions of number of monthly payments (lhs) for different claim codes, (middle) for different reporting delay quarters, (rhs) for different calendar months of claim occurrences; discrete distributions are for better visualization illustrated by line plots.

In Figure 9 we give the distributions of the number of monthly payments per claim as a function of different feature components (note that these distributions are discrete but for better visualization we use line plots). The left-hand side shows the distributions for different claim codes, these distributions reflect Figure 8 (lhs), and they illustrate that the claim code feature results in important differences in the number of payments. Figure 9 gives the distributions as a function of reporting delay quarters. These distributions reflect the empirical plot in Figure 5 (lhs) which says that the reporting delay information is an important feature for the number of payments,

i.e. any corresponding independence assumption should be abolished. Also the calendar month of claim occurrence seems to be important feature information as can be seen from Figure 9 (rhs). However, here the explanation might be more difficult because this feature component strongly interacts with the claim code feature. Nevertheless, this provides justification that the feature information in (4.3) has been chosen reasonably.

4.3.4 Comparison to the chain-ladder method

We compare our predicted number of monthly payments to the one we receive from the chain-ladder method. For the chain-ladder method we can consider a maximal reporting *plus* settlement delay of $m = 107$ months in (2.4), since the data does not go beyond this delay for the chain-ladder method (because we do not use the information of reporting delays in the payment process). For this reason, we also need to cap accordingly the number of payments received from our neural network prediction, because the latter allows us to predict payments up to a maximal delay of $\lceil U \rceil - \lceil T \rceil + d = \lceil U \rceil - \lceil T \rceil + 107 \geq 107$ months. The capping to the chain-ladder domain provides us with a predicted number of 3'365 RBNS payments (compared to the 3'437 RBNS payments stated in Section 4.3.3).
This number of 3'365 RBNS payments comes from one single Monte Carlo extrapolation on the lower triangle. To diminish Monte Carlo simulation uncertainty we repeat this procedure 50 times and average the results. This provides us with an average of 3'466 RBNS payments, and if we truncate this estimate to the chain-ladder domain we receive in average 3'393 RBNS payments. This average has an estimated Monte Carlo simulation error (standard deviation) estimate of 19 RBNS payments.
Secondly, we need to estimate the number of payments for the 402 IBNR claims in Table 1. These IBNR claims are established with feature information like claim type, accident date, etc., and we could design a network to estimate the number of payments for these IBNR claims. We are going to do this in detail for payment amounts, below. For the moment, we use a simple homogeneous (crude) estimate which is obtained by multiplying the number of IBNR claims with the average number of payments per claim, which is given by 1.7606 (this corresponds to the number in the caption of Table 3 but averaged over the 50 Monte Carlo extrapolations). This results in 708 payments for IBNR claims, which altogether provides a predicted number of 4'101 monthly payments for claims having occurred before the cutoff date τ.

	# RBNS payments	# IBNR payments	total number of payments
neural network	3'393(19)	708	4'101
ODP chain-ladder method	NA	NA	4'064

Table 4: Number of outstanding monthly payments after the cutoff date restricted to the domain of the chain-ladder method, the bracket shows one standard deviation Monte Carlo uncertainty.

We now perform the ODP chain-ladder method on the annualized number of monthly payments in the upper triangle, characterized by the (annual) indexes (i, j) and

$$N^{\text{paid}}_{i,j} = \sum_{0 \leq k \leq j} N^{\text{paid}}_{i,j-k,k},$$

we refer to (2.4) for the definition of the latter variables. The chain-ladder prediction provides us with an estimate of 4'064 monthly payments in the lower triangle, thus, slightly less than the 4'100 monthly payments obtained from the neural network approach. But, the difference is (surprisingly) small. In fact, the difference is very small which provides a first proof of concept for our approach. Applying the ODP bootstrap method of England–Verrall [10] (with dispersion estimate 6.79) we receive an RMSEP of 279, i.e. of a bigger magnitude than the difference between the two methods in Table 4. Basically, this means that on the aggregate level we cannot distinguish the results. Of course, this is a very promising result for individual claims reserving because it supports that we meet the right level of total predictions, but beyond the ODP chain-ladder method we have an allocation to individual claim types, see e.g. Figure 9.
Our payment indicator model has been starting from the number of reported claims (being Poisson distributed), and then we have applied a Bernoulli thinning to that process to receive the number of monthly payments. Conditionally given the number of reported claims, this corresponds to a sum of independent Bernoulli random variables (which due to heterogeneity in $\boldsymbol{x}_k$ is not binomially distributed in our case). This is different from the assumptions of the CRM in Wahl et al. [30] who assume a Poisson random variable instead: our number of monthly payments is bounded, $\mathbb{P}$-a.s., given the number of reported claims, whereas in Wahl et al. [30] this random variable is unbounded. Unconditionally, our number of payments is Poisson distributed (thinned number of reported claims), this assumption is also used in Wüthrich [31]. However, the latter reference assumes that these Poisson distributed numbers of payments are independent for different i and j, whereas this is clearly not the case here.

5 Payment size modeling of RBNS claims

We are on a monthly scale and we predict the payment process $(Y_k)_{k\geq 1}$ of reported claims, given the indicator process $(\mathcal{Y}_k, O_k)_{k\geq 1}$ (modeled in the previous section), see Section 4.1.

5.1 Recovery payments

In total we have observed 56'852 payments from $\tau_0 = 2009/01/01$ to $\tau = 2017/12/31$ on all reported claims. 1'122 of these payments have a negative sign, i.e. in these cases the insurance company can recover claim expenses. Thus, roughly 2% of the payments $Y_{\ell,k}$ are negative.

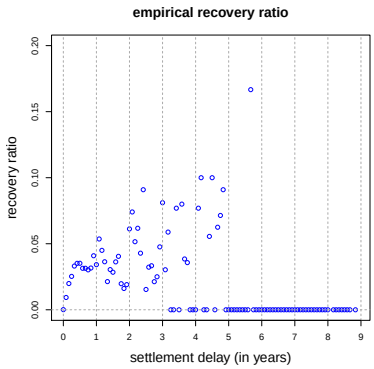

Figure 10: Ratio of the number of recovery payments as a function of settlement month $k \geq 0$.

Figure 10 illustrates the empirical ratio of recovery payments relative to all payments as a function of the settlement month $k \geq 0$. Not surprisingly, this ratio depends on the settlement month k, but it also depends on other feature information like claim code, etc. We do not show these other graphs because scarcity of recovery payments does not provide nice explanatory graphs with obvious structure.

We choose a next 3rd neural network that models the recovery indicator. We are doing this in a simple and straightforward way by assuming that the random variables $\mathcal{R}_k = \mathbb{1}_{\{Y_k<0\}}$ are Bernoulli distributed, conditionally on having a payment $\mathcal{Y}_k = \mathbb{1}_{\{Y_k\neq 0\}} = 1$ in settlement months $k \geq 1$, with recovery probabilities

$$\boldsymbol{x}_k \;\mapsto\; p(\boldsymbol{x}_k) \;\in\; (0,1),$$

for feature information

$$\boldsymbol{x}_k = (W, v(T), (\mathcal{Y}_l)_{k-12\leq l\leq k-1}, \mathfrak{y}_{k-1}, O_{k-1}, O_k, k, \vartheta)^\top \;\in\; \mathbb{R}^{19} \times \{\text{claim code}\}. \tag{5.1}$$

This feature information differs from (4.3), because to model recovery indicators we additionally know from the previous modeling step whether the claim has been closed at the end of settlement month k, i.e. we know O_k.

We set up the neural network for recovery indicators completely analogously to the one used in Section 4.3.1, the only differences are that the input is changed according to the new features (5.1) and the output is changed to a logistic regression function. The bias regularization, which is particularly important in class imbalanced problems, is performed replacing the Poisson family by the binomial one in Listing 3. Figure 2 (rhs) shows the gradient descent training performance, and Table 5 provides the calibration statistics. Again we benchmark with a GLM, the (early stopped) neural network again shows superior performance over the GLM, and we continue with the network regression model.

	run time	# param.	in-sample loss	estimated average probability $\widehat{p}(\cdot)$
homogeneous model	–	1	0.2235	2.36%
logistic GLM	1s	37	0.1900	2.36%
neural network	30s	944	0.1580	2.36%

Table 5: Recovery indicator process: run time of calibration, number of model parameters, loss, estimated average recovery probability.

Figure 11 compares the estimated neural network model to the observations. We show the sensitivities of the recovery rates as a function of settlement month $k \geq 1$ (lhs) and of reporting delay W (rhs). The regression function is less sensitive in other variables such as the (seasonal) calendar week $v(T)$ of the accident date T. Remark that the recovery process can easily be extrapolated to the lower triangle because all necessary information is readily available from the extrapolation given in Section 4.3.3.

5.2 Descriptive data analysis of claim payment sizes

All previous models have been considering point processes (Poisson, categorical and Bernoulli). Claim sizes are typically modeled with absolutely continuous distribution functions. Before doing so we try to understand claim payment sizes empirically.

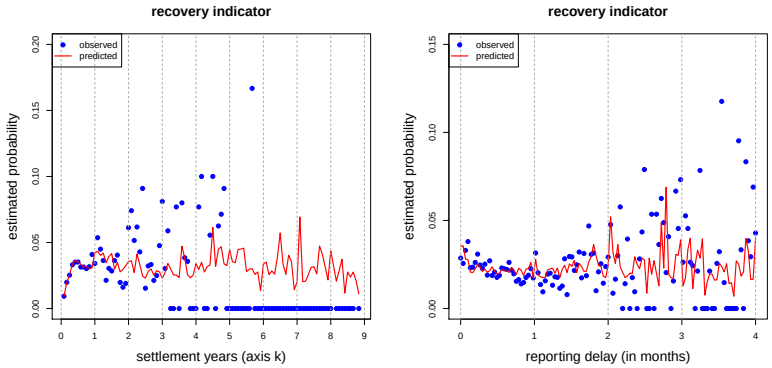

Figure 11: Observed and estimated recovery payment indicators as a function of (lhs) settlement month and (rhs) reporting delay.

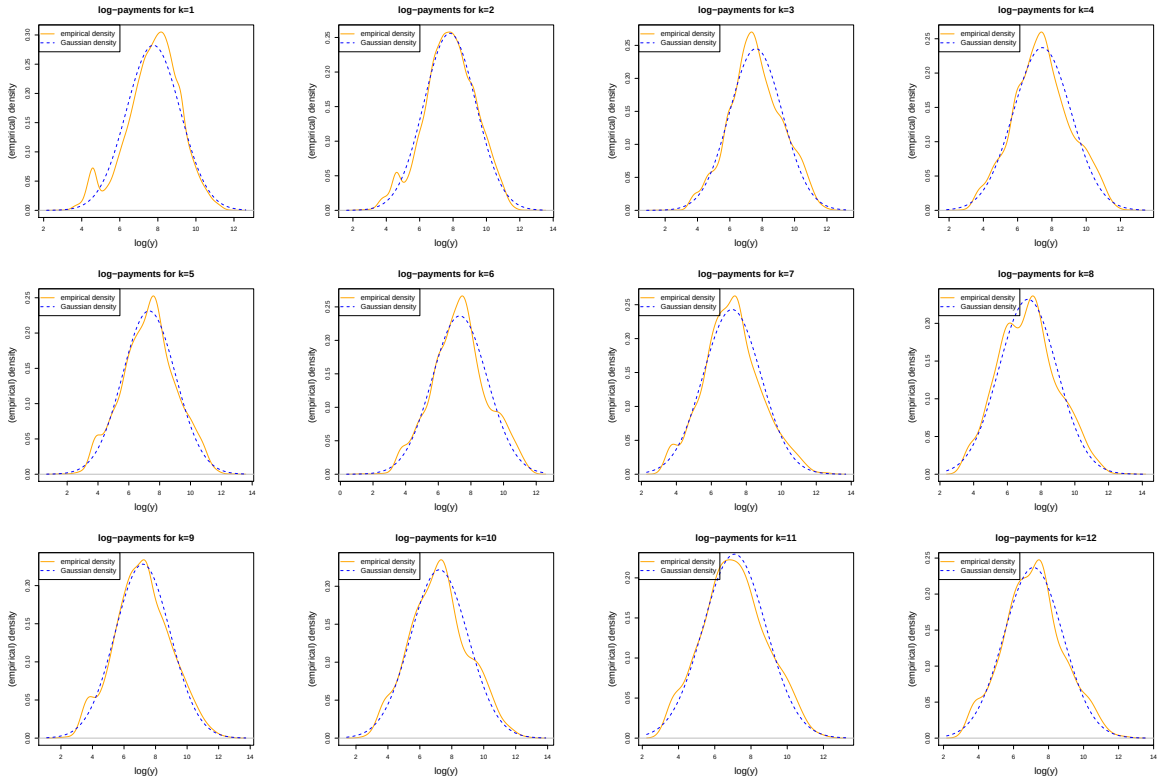

Figure 12: Logarithmized absolute payments $\log|Y_k|$ in settlement months $k = 1, \ldots, 12$.

Figure 12 shows logarithmized absolute payments $\log|Y_{\ell,k}|$ for settlement months $k = 1, \ldots, 12$ of all claims ℓ with $\mathcal{Y}_{\ell,k} = 1$. The orange curve gives the empirical density and the blue dotted line is a Gaussian approximation (method of moments fit). We observe that the Gaussian distribution provides a surprisingly good fit to the logged absolute values $\log|Y|$ of non-zero payments. However, there are certain default values for payments, see for instance settlement month $k = 1$ (top-left). These default values may cause some difficulties in modeling because they rather come from a discrete than an absolutely continuous distribution function.

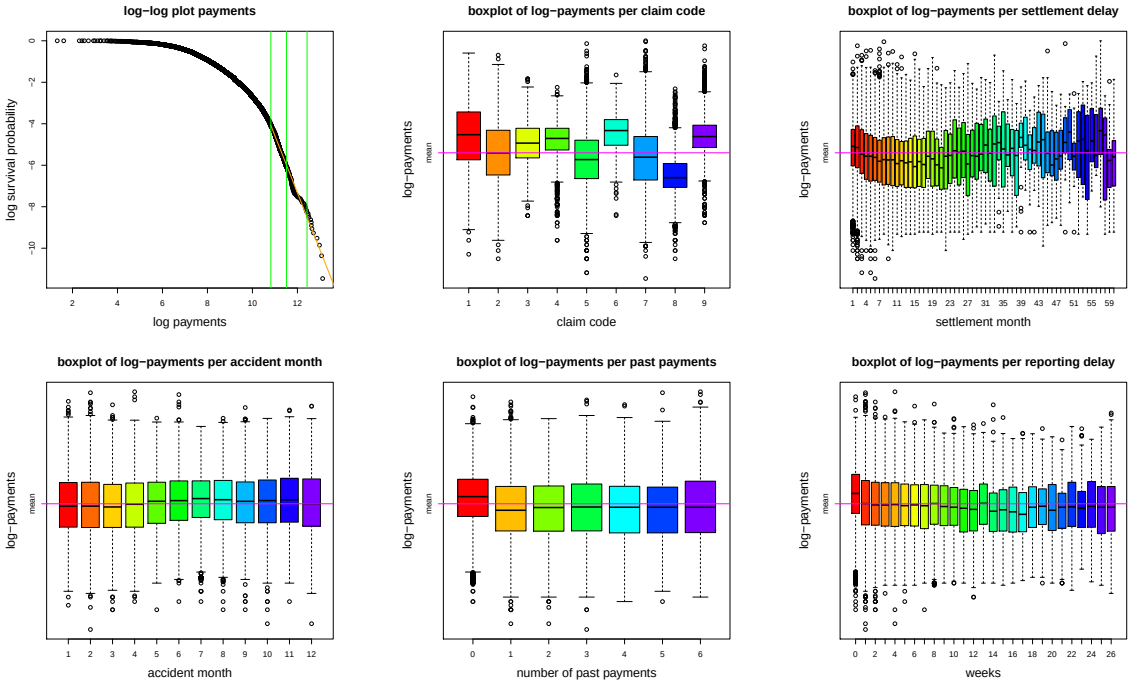

Figure 13: top row: (lhs) Log-log plot of absolute values of payments $\log|Y_k|$, $k \geq 1$, with Pareto tail approximation, (middle) boxplots of payments w.r.t. claim codes, (rhs) w.r.t. settlement month $k \geq 1$; bottom row boxplots of payments w.r.t. (lhs) accident months, (middle) the number of previous payments, and (rhs) reporting delay W (the magenta lines are the averages over all payments).

Figure 13 (1st row, lhs) gives the log-log plot of all absolute values of payments. The vertical green lines are at values $\log(50'000)$, $\log(100'000)$ and $\log(250'000)$, the orange straight line gives a Pareto distribution approximation to all payments above $50'000$; the MLE of the Pareto tail parameter α is $\widehat{\alpha} = 2.75$ which indicates (together with the plot) heavy-tailedness of large payments. Figure 13 (1st row, middle) shows the boxplots for different claim codes. This plot tells us that the claim code is important feature information for payments modeling. Figure 13 (1st row, rhs) gives the boxplots as a function of settlement months $k \geq 1$. This plot shows time dependence and, henceforth, rejects many models proposed in the actuarial literature. The 2nd row of Figure 13 shows payments as a function of accident months (related to $v(T)$), the number of previous monthly payments $\mathfrak{y}_{k-1} \in \mathbb{N}_0$ and of reporting delay W. From these plots we conclude that payments are less sensitive in these variables. However, the latter is misleading because some of these variables interact with others, for instance, reporting delay is an important feature for claim code 7 but not for claim code 6 as can be seen from Figure 14.

5.3 Payment size regression modeling

We set up a 4th neural network to model the payment sizes. We have observed 56'852 payments. This is a rather small number of observations, and to prevent from over-fitting, we reduce feature information compared to (5.1). In particular, the masking for early periods has caused

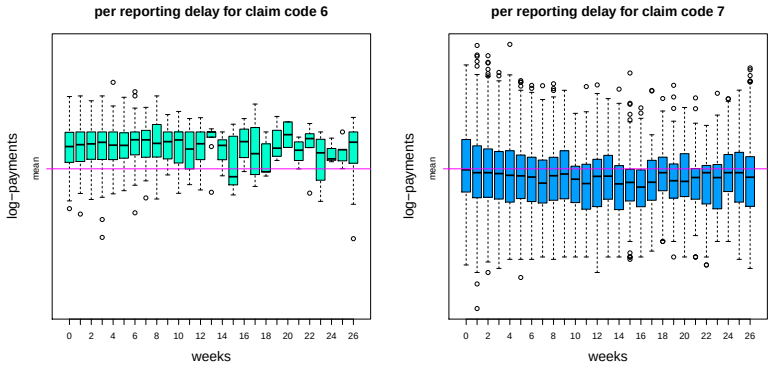

Figure 14: Boxplots of logged absolute values of payments w.r.t. reporting delay W for claim code 6 (lhs) and 7 (rhs).

some smoothness issues in our experiments on payment size modeling, therefore, we reduce the payment indicator history in the feature information. On the other hand, we include the recovery indicator $\mathcal{R}_k$, which is available due to the previous modeling step (presented in Section 5.1). We choose the following feature information for the modeling of Y_k, conditional on $\mathcal{Y}_k = 1$,

$$\boldsymbol{x}_k = (W, v(T), \mathcal{Y}_{k-1}, \mathfrak{y}_{k-1}, O_{k-1}, O_k, \mathcal{R}_k, k, \vartheta)^\top \in \mathbb{R}^9 \times \{\text{claim code}\}. \tag{5.2}$$

We emphasize that this feature information does not include past payments information $(Y_l)_{l<k}$, see Section 4.1, and therefore no Monte Carlo extrapolation of Y_k is needed.

	run time	# param.	in-sample loss	estimated average
(a0) homogeneous model (log-scale)	–	1	2.6221	7.5622
(a1) Gaussian neural network (log-scale)	89s	744	2.0352	7.5661
(a2) – bias corrected (original scale)			13'208	6'254
(b0) homogeneous model (original scale)	–	1	2.3574	6'254
(b1) gamma neural network (original scale)	91s	744	1.8137	6'315
(b2) – bias corrected (original scale)			12'851	6'254
(c1) gamma GLM benchmark (original scale)	1s	17	1.9352	6'357
(c2) – bias corrected (original scale)			13'186	6'254

Table 6: Claim size modeling: run time of calibration, number of model parameters, loss, estimated average for (a0)-(a1) Gaussian model on log-scale with square loss, (a2) bias corrected on original scale with rooted square loss, (b0)-(b1) gamma model on original scale with gamma deviance loss, (b2) bias corrected on original scale with rooted square loss, (c1)-(c2) gamma GLM benchmark.

We try two different neural networks for claim payments $|Y_k|$ modeling. The architectures of the two networks only differ in the output layer: (a) linear output function for log-payments $\log|Y_k|$ with square loss function, and (b) exponential output function for payments $|Y_k|$ with gamma deviance loss function. Besides this difference the two architectures are chosen identical, and they are identical to the ones chosen above with input dimension adapted to feature (5.2). To track over-fitting we again use 20% of the data as validation data. We choose this validation

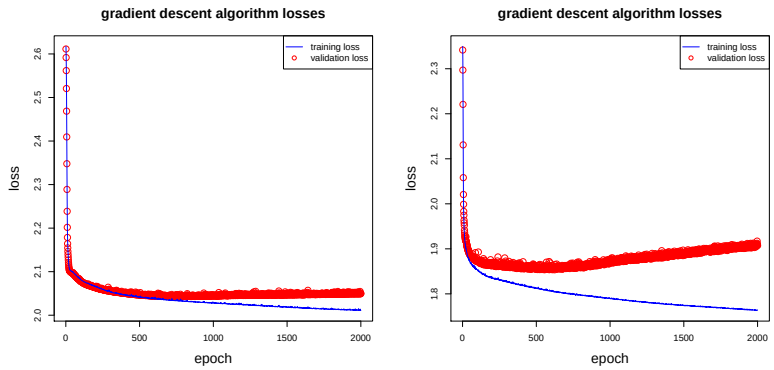

Figure 15: Training and validation losses of claim sizes fitting (lhs) log-normal model, (rhs) gamma model.

set in a stratified way w.r.t. payment sizes such that we have a well-balanced validation set (compared to the training data). The gradient descent performance is illustrated in Figure 15, and the corresponding losses are given in Table 6 on lines (a0)-(a1) for the Gaussian case on the log-scale, and on lines (b0)-(b1) for the gamma case on the original scale. Both networks result in a biased estimated model on the original scale because we work on the log-scale in the former and because we do not use the canonical link in the latter. Therefore, we correct for the biases by shifting the intercept parameter in the output layer accordingly. This provides the results on lines (a2) and (b2) on the original scale (note that the estimated averages of 6′254 are now in line with the empirical estimate). To compare the two models we calculate the rooted square losses, and we note that the gamma model should be preferred according to this figure.

Linking to Section 4.1, it is commonly the case that the gamma model has a better performance on the original scale than the log-normal one (the latter being estimated in the log-scale). Usually, the log-normal model is very accurate on the log-scale, but the back-transformation to the original scale would require that also the variance parameter is modeled as a regression function w.r.t. feature information to get sufficient accuracy on the original scale; thus, we have to estimate two parameters with accurate regression models in the log-normal model. On the other hand, the gamma model directly estimates the mean parameter on the original scale, which typically leads to better estimates for means compared to the log-normal model (even if we correct with a constant for the bias, accounting for the fact that we do not use the canonical link). For this reason, we focus on the gamma model in the sequel and we denote the resulting regression function by $\boldsymbol{x}_k \mapsto \mu(\boldsymbol{x}_k)$. This gamma neural network model is benchmarked with a gamma GLM in Table 6, the figures indicate that we should continue with the more complex neural network.[7]

Figure 16 (lhs) shows the QQ-plot of the shifted residuals $\text{Res}_k + 1 = (|Y_k| - \mu(\boldsymbol{x}_k))/\mu(\boldsymbol{x}_k) + 1$ which should have a gamma distribution with mean 1 and constant dispersion under our model assumptions. Figure 16 (lhs) shows that the observations are clearly more heavy tailed than the gamma model allows for. This is also supported by Figure 16 (rhs) which provides the Tukey–Anscombe plot. Additionally, we have explored a small-large payments split, and we have fitted

[7]Note that a gamma generalized additive model (GAM) has a similar performance as the gamma GLM. The corresponding gamma loss of the GAM is 1.9246 and the rooted mean square loss is 13'172.

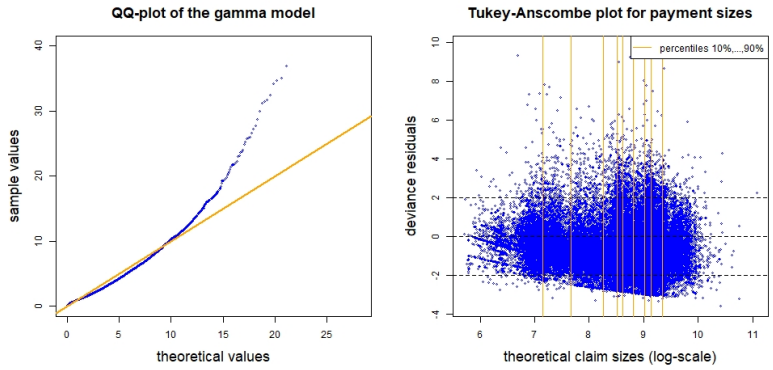

Figure 16: (lhs) QQ-plot of the estimated gamma model (b2) of Table 6, (rhs) Tukey–Anscombe plot of the residuals against the estimated means.

a gamma model to the small payments. The resulting Tukey–Anscombe plot looked very similar to the one in Figure 16 (rhs), moreover, the manual intervention of choosing a large payment threshold has clearly been visible in the plot (which is unsatisfactory).

Conclusion. We continue with gamma model (b2) of Table 6. This only requires one single network to be fitted and one single (mean) parameter to be estimated. Because we are *only* interested in having good mean estimates this is sufficient for our purposes, we also refer to the disclosure on page 12.

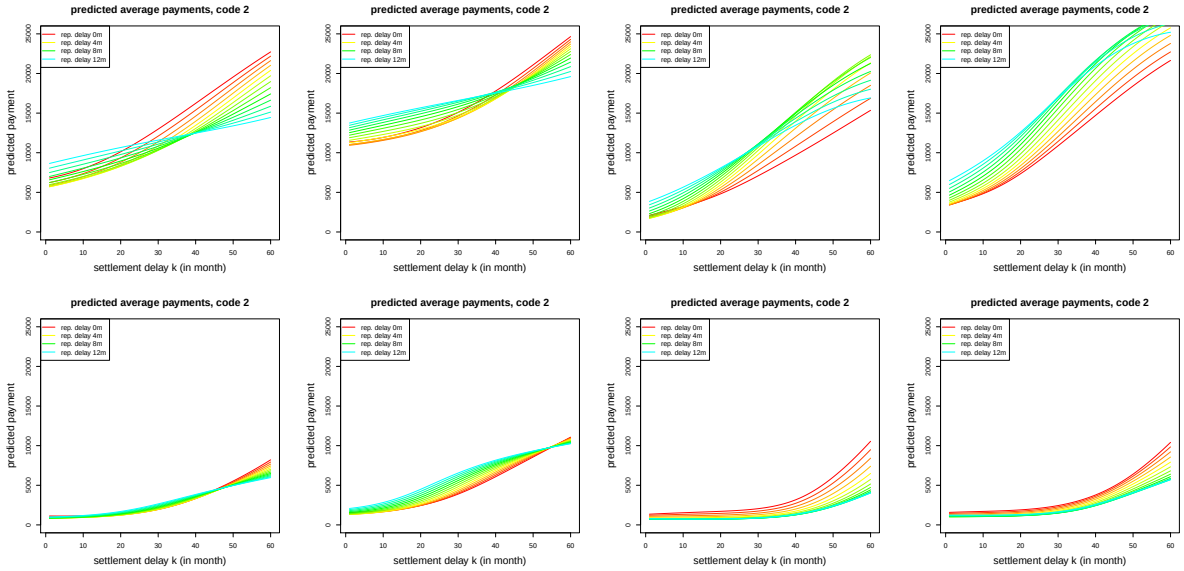

Figure 17: (1st row) predicted payments and (2nd row) predicted recovery payments for different features $\boldsymbol{x}$ and identical claim code 2: the x-axis illustrates different settlement months $k \in \{0, \ldots, 60\}$, the different colors illustrate different reporting delays from 0 days to 1 year, and the plots from left to right show $(\mathcal{I}_{k-1}, O_k) = (0,1), (1,1), (0,0), (1,0)$.

The fitted network allows us to study expected payments $\mu(\boldsymbol{x}_k)$ as a function of $\boldsymbol{x}_k$. Figure

17 illustrates such plots. We fix claim code 2, the x-axis illustrates different settlement months $k \in \{0, \ldots, 60\}$, and the different colors illustrate different reporting delays from 0 days to 1 year. The first row gives predicted claim payments and the second row predicted recovery payments (conditional on $\mathcal{R}_k = 1$). The first column gives the predictions for claims without payments in the previous period $\mathcal{Y}_{k-1} = 0$ and claims remaining open $O_k = 1$ after settlement period k, the second column for $\mathcal{Y}_{k-1} = 1$ and $O_k = 1$, the third column for $\mathcal{Y}_{k-1} = 0$ and $O_k = 0$, and the fourth column for $\mathcal{Y}_{k-1} = 1$ and $O_k = 0$. We observe that all this feature information influences predicted payments and recoveries. For these plots the undetermined feature components in (5.2) have been set to $v(T) = 1/4$ (March), $O_{k-1} = 1$ (open), $\mathfrak{y}_{k-1} = \mathcal{Y}_{k-1}$ (only a potential payment in the previous period) and the LoB has been chosen to be the more likely one for the given claim code 2.

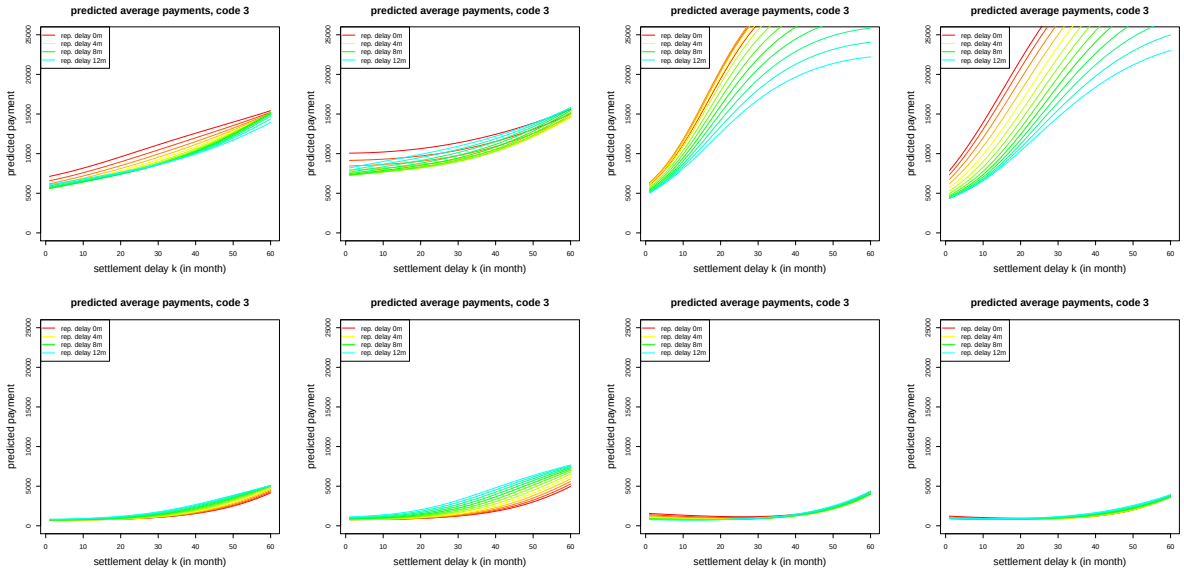

Figure 18: (1st row) predicted payments and (2nd row) predicted recovery payments for different features $\boldsymbol{x}$ and identical claim code 3: the x-axis illustrates different settlement months $k \in \{0, \ldots, 60\}$, the different colors illustrate different reporting delays from 0 days to 1 year, and the plots from left to right show $(\mathcal{I}_{k-1}, O_k) = (0,1), (1,1), (0,0), (1,0)$.

Figure 18 gives the same plots for claim code 3 (on the same y-scale). Comparing the top right plots of Figures 17 and 18, for instance, illustrates that the sizes of expected payments can be either increasing or decreasing as a function of reporting delay W (different colors), for claim code 2 it is increasing whereas for claim code 3 it is decreasing, i.e. later reported claims are less severe for claim code 3.

5.4 Extrapolation of payments to the lower triangle

We are now ready to expand the payment processes of RBNS claims (this encloses relapses of closed claims) to the lower triangle. We do this by successive one-period-ahead predictions using Monte Carlo simulations, see Section 4.1. This includes three steps for each settlement month k in the lower triangle. *1st step:* simulate the categorical payment and claim status indicator $\mathcal{I}_k = \mathcal{Y}_k + 2 \cdot O_k$ based on feature information (4.3); if $\mathcal{Y}_k = 1$ process the next two

steps, otherwise go to settlement month $k+1$ for this claim. *2nd step:* simulate the Bernoulli recovery indicator $\mathcal{R}_k$ based on feature information (5.1), this feature information includes the previously simulated claim status indicator O_k. *3rd step:* calculate the expected payment size $\mathbb{E}[Y_k] = \mu(\boldsymbol{x}_k)$ based on feature $\boldsymbol{x}_k$ given in (5.2), this includes the previously simulated claim status indicator O_k and the recovery indicator $\mathcal{R}_k$. Proceed to settlement month $k+1$. This procedure is iterated over the entire lower triangle.
To smooth simulation uncertainty, we average over 50 Monte Carlo extrapolations in the lower triangle. This provides us with an RBNS reserve of $23'832 \cdot 10^3$ for all reported claims at the cutoff date τ. If we again truncate these payments to the chain-ladder domain, see Section 4.3.4, we receive RBNS reserves of $23'160 \cdot 10^3$ (having a Monte Carlo simulation uncertainty (standard deviation) of $195 \cdot 10^3$). In the next section we compare these results to the classical chain-ladder reserves. We conclude this section by giving some plots of the resulting RBNS reserves.

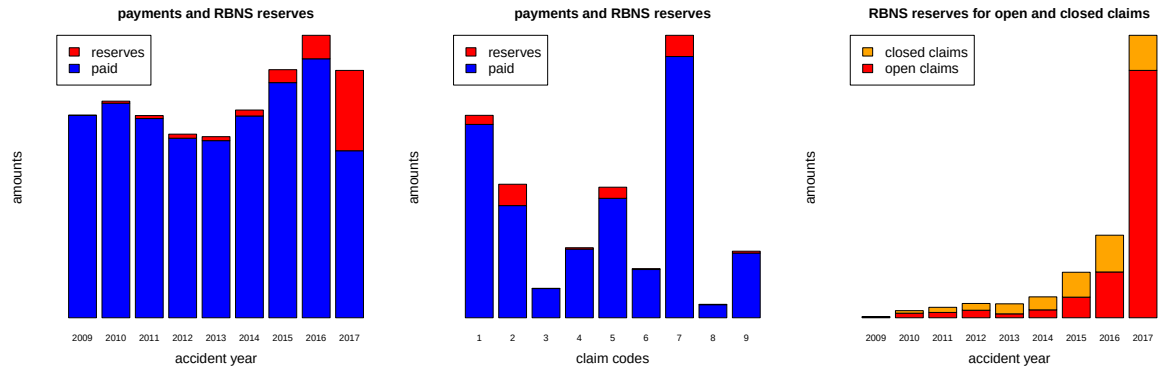

Figure 19: Payments and reserves for reported claims (lhs) as a function of accident year, (middle) split by claim codes, (rhs) reserves split by open and closed claims at the cutoff date (the latter being reserves for re-openings coming from relapses).

We are now able to illustrate the reserves for reported claims w.r.t. the available feature information. Figure 19 (lhs, middle) illustrate payments and reserves for reported claims at the cutoff date τ as a function of accident year and of claim codes, respectively. Figure 19 (rhs) shows the split of reserves to open reported claims and closed claims, the latter are reserves for re-openings for relapses. We note that in our data there is quite some influence by relapses (in terms of reserves), though less than 3% of all closed claims are re-opened, see Figure 7.

6 Claim amounts for IBNR claims

We calculated RBNS reserves for reported claims in Sections 4 and 5. In Section 3 we predicted the number of IBNR counts, and we expect 402 IBNR claims, see Table 1. In the present section we assign claim amounts to these IBNR claims. For this we again turn to a weekly time scale as in Section 3, and the time ceiling function $\lceil \cdot \rceil$ will refer to the weekly time scale in this section.

6.1 Description of our IBNR claim modeling approach

The modeling of IBNR claim amounts is done differently compared to the claim payments modeling of RBNS claims. This difference in modeling is common in actuarial practice because

the nature of the underlying data is different. For RBNS claims we have partially observed individual claim settlement processes, and the main task in RBNS reserving is to complete these individual claim histories in a path-dependent way. We have done this by successive one-period-ahead predictions, see Section 4.1. For IBNR claims we do not have partially observed histories, and there is no point in performing multiple path-dependent Monte Carlo simulations, but, instead, ultimate claim amounts of individual claims are estimated directly, still considering individual (static) claim feature information if available. In our case this information is exactly the (weekly) feature information $(i, j, v(i), \vartheta) = (\lceil T \rceil, \lceil U \rceil - \lceil T \rceil, v(\lceil T \rceil), \vartheta)$ from IBNR claim counts modeling, see (3.2), and $\lceil \cdot \rceil$ is the weekly time ceiling function.

6.2 IBNR claim amount modeling

To estimate IBNR claim amounts we set up networks 5 and 6. The 5th network serves at predicting whether we have a claim without any payments (called zero claim), and the 6th network will assess the total claim amount in case of a non-zero claim. For both networks we use exactly the same weekly feature information $(i, j, v(i), \vartheta) = (\lceil T \rceil, \lceil U \rceil - \lceil T \rceil, v(\lceil T \rceil), \vartheta)$ as for IBNR claim counts modeling, see (3.2); this feature information corresponds to the claim occurrence week, the reporting delay in weekly units, the annual seasonality and the static feature information $\vartheta = (\text{LoB}, \text{claim code})$. The total IBNR reserves are then obtained by

$$\sum_{\vartheta} \sum_{i+j>\lceil \tau \rceil} \widehat{\Lambda}(i, j, v(i), \vartheta) \left(1 - \widehat{q}(i, j, v(i), \vartheta)\right) \widehat{\gamma}(i, j, v(i), \vartheta), \tag{6.1}$$

where $\widehat{\Lambda}(i, j, v(i), \vartheta)$ is the estimated number of IBNR claims (estimated by the 1st network given in Section 3.2), $\widehat{q}(i, j, v(i), \vartheta) \in (0, 1)$ is the estimated probability of having a zero claim (coming from the 5th network), and $\widehat{\gamma}(i, j, v(i), \vartheta) > 0$ is the estimated claim amount in case of a non-zero claim (coming from the 6th network).

We start by describing the available data for this prediction. Above, we have designed networks that allow us to complete the path-dependent trajectories of claims reported at the cutoff date τ. Thus, for all reported claims we now have a fully developed (predicted) claim history over the $d + 1 = 108$ settlement months which provides total payments per reported claim. There is one tricky issue that has already been mentioned in Section 4.1: in the upper triangle we have observed payments Y_k and in the lower triangle we have predicted payments $\widehat{Y}_k = \mathbb{E}[Y_k] = \mu(\boldsymbol{x}_k)$. Thus, we have an inconsistency between upper and lower triangle, and we are not allowed to simply aggregate these payments and predicted payments to assess the total individual claim amounts, because claims with different settlement delays would be of different quality. For this reason we replace payments Y_k in the upper triangle by corresponding (in-sample) predictions $\widehat{Y}_k = \mu(\boldsymbol{x}_k)$. The last column of Table 6 tells us that this replacement is unbiased, i.e. the total prediction is exactly equal to the total payments in the upper triangle. After this replacement in the upper triangle we aggregate all predicted payments to receive the total predicted claim amounts $\widehat{C}_\ell = \sum_{0 \le k \le d} \widehat{Y}_{\ell,k}$ for all reported claims $\ell \ge 1$. These predicted claim amounts $\widehat{C}_\ell$ are complemented with the static feature information $(\lceil T_\ell \rceil, \lceil U_\ell \rceil - \lceil T_\ell \rceil, v(\lceil T_\ell \rceil), \vartheta_\ell)$.

Based on feature information $(i, j, v(i), \vartheta)$ we design networks 5 and 6 to estimate the zero claim probability $q(i, j, v(i), \vartheta) \in (0, 1)$ and the claim amount $\gamma(i, j, v(i), \vartheta) > 0$ in case of a non-zero claim. Similar to Section 3.2, we design a network with logistic Bernoulli responses for assessing

zero claims, and one with gamma responses for the total claim amounts for non-zero claims. The resulting calibration statistics are given in Table 7.

	run time	# param.	in-sample loss	estimated averages $\widehat{q}(\cdot)$ and $\widehat{\gamma}(\cdot)$, resp.
IBNR zero claim indicator: homogeneous model	–	1	0.5217	7.28%
IBNR zero claim indicator: neural network	18s	644	0.4551	7.28%
IBNR claim amounts: homogeneous model	–	1	2.1498	11'109
IBNR claim amounts: neural network	13s	644	1.6905	11'109

Table 7: IBNR zero claim indicators and IBNR claim amounts (in case of non-zero claims): run time of calibration, number of model parameters, loss, estimated averages.

	RBNS reserves	IBNR reserves	total reserves
neural network	23'160	3'338	26'498
ODP chain-ladder method	NA	NA	25'011
– RMSEP (dispersion $\phi = 108'944$)			2'730

Table 8: Predicted payments (claims reserves) after the cutoff date restricted to the domain of the chain-ladder method (units in 1'000).

Established with these two additional networks we complete the predicted IBNR claim counts of Section 3.2 with total individual claim amounts, see (6.1). This results in IBNR reserves of $3'338 \cdot 10^3$, thus, altogether we receive $26'498 \cdot 10^3$ RBNS and IBNR reserves from the network approach. We compare this estimate to the one received from the chain-ladder method, see Table 8. The difference between the two methods is surprisingly small, i.e. only slightly more than half of one RMSEP of $2'730 \cdot 10^3$, received by the bootstrap method of England–Verrall [10] in the ODP model. This small difference is a second proof of concept that our neural network approach provides reasonable results. In fact, we predict slightly higher claims than in the chain-ladder method; this can easily be justified because the chain-ladder method has difficulties to appropriately consider seasonality, change of portfolio mix, etc. Moreover, an advantage of our approach is that it allows us to analyze the reserves as a function of all available features.
The top row of Figure 20 shows the two methods split w.r.t. accident years: allocation of reserves to accident years is very similar between the chain-ladder method and our network approach. The middle row illustrates the number of claims and the resulting claim averages (based on the neural network prediction). We predict a slight decrease in average claim amounts from 2016 compared to 2017. The bottom row of Figure 20 shows the claims split according to accident quarters (lhs) and claim codes (rhs). Interestingly, we have a much bigger claim payment exposure in the first quarters of the calendar years, and summer time is a comparably quiet period. Moreover, reserves for older claims are mainly set for the first accident quarters. From Figure 21 we observe that claim codes 1, 2 and 7 have mainly exposure in winter months, which justifies the claim codes split in Figure 19 (bottom rhs).
Our final remark on IBNR reserves estimation concerns our model choice. We have started from a Poisson model for IBNR claim counts modeling which provided Poisson mean estimates $\widehat{\Lambda}$.

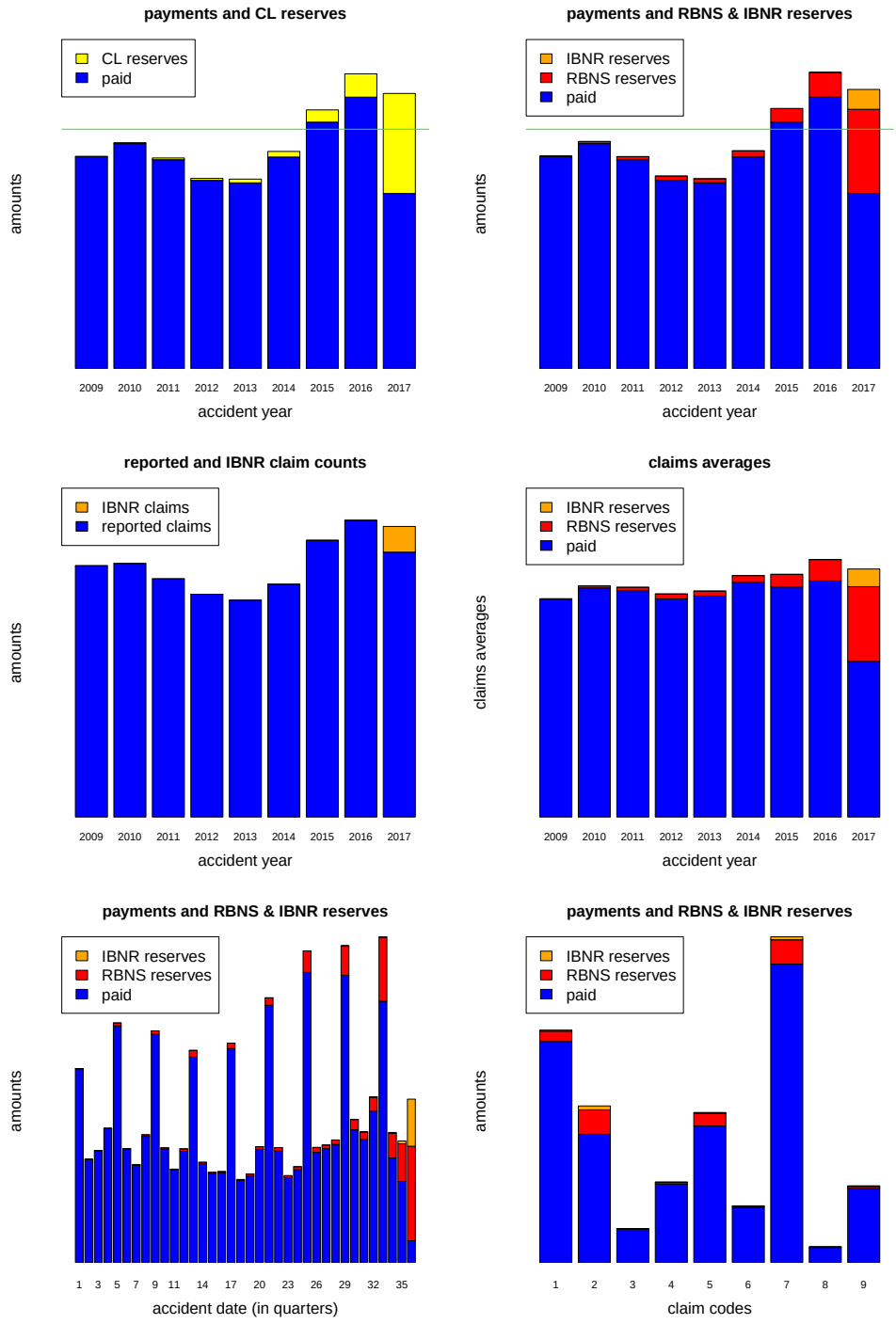

Figure 20: (top lhs) Payments and CL reserves, (top rhs) payments, RBNS and IBNR reserves from network approach, (middle lhs) number of claims, (middle rhs) claims averages, (bottom lhs) payments and reserves per accident quarter, (bottom rhs) payments and reserves per claim codes.

These IBNR claim counts have been thinned by the zero claim probabilities $\widehat{q}$. The remaining thinned IBNR claim counts with mean estimates $\widehat{\Lambda}(1-\widehat{q})$ have been equipped with independent

gamma claim amounts. Implicitly, we have assumed that the shape parameters of the gamma claim amount distributions do not depend on the considered features, and as a result we obtain Tweedie's compound Poisson model for IBNR claims reserves, see Jørgensen [13]. This model would also allow for direct parameter estimation, see Smyth–Jørgensen [26]. This direct parameter estimation would reduce the three networks for IBNR reserve estimation used above to a single one. However, the latter approach is usually not explored in insurance practice for several reasons. Firstly, one loses tractability of the claim count and claim amount separation. This separation is important if one wants to partition into attritional and large claims. Secondly, typically, different features are used for claim count and claim amount modeling which proposes to model claim counts and claim amounts in separate regression models. In our approach here we have decided to hold on to this practice, also because run times to fit the two additional networks are small, see Table 7. Another point is that we estimate a frequency-severity model with two networks each having a one-dimensional parameter, Tweedie's compound Poisson model would estimate a two-dimensional parameter in one network. It is still an open question whether the latter approach leads to sufficient robustness in estimation and prediction.

7 Conclusion

We provide a full individual claims reserving approach for calculating reserves on individual claims having different features. In total, we have used six neural networks:

- network 1 models claim counts of incurred but not reported (IBNR) claims;
- network 2 serves at modeling the payment indicator and the claim status processes of reported but not settled (RBNS) claims;
- network 3 serves at modeling an indicator process for recovery payments of RBNS claims;
- network 4 serves at modeling expected claim and recovery payments of RBNS claims;
- network 5 models an indicator for having an IBNR claim without payments (zero claim);
- the final network 6 models claim amounts for non-zero IBNR claims.

The basic network architecture is the same for all six networks, only the input dimensions differ (depending on the available feature information in each modeling step), and the output functions differ (depending on the type of distribution function we would like to model). We fit this model on weekly data for IBNR claims and on monthly data for RBNS claims. The run times of these modeling steps are reasonably short (as is indicated in the corresponding tables), and the resulting reserves almost meet the benchmark received from the chain-ladder method. The latter is surprising and a critical reader may question the complex model if a simple one comes to the same conclusions. We want to emphasize that the advantage of the complex model is that we obtain more granular information which is crucial in insurance pricing. Moreover, the chain-ladder method typically fails on non-stationary data, if, for instance, the portfolio mix changes. Unfortunately, we do not have such data, but in future research it would be interesting to explore this point.

We conclude with an outlook of possible next steps. Clearly, the weakness of our approach is that we cannot directly simulate from our parametric model. Our model is sufficient for claims

reserving because it provides reasonable mean estimates, however, the distributional assumptions made do not capture the right tail behavior. We see two ways to improve this point. Either we choose a more complex claim size model than our gamma distribution. This will work but it will come at the price of longer run times and more complexity (which may also negatively impact model stability). Another option could be to explore non-parametric bootstrapping (in an appropriate way). A second research direction should explore model uncertainty, possibly using bootstrapping. Run times of our approach are (almost) on the level to make a bootstrap analysis computationally feasible. A further topic of research is to explore more extended feature information for claims reserving. This may include claims incurred, premium and exposure information, more detailed claim information, payment types, etc.

References

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

[2] Arjas, E. (1989). The claims reserving problem in non-life insurance: some structural ideas. *ASTIN Bulletin* **19/2**, 139-152.

[3] Badescu, A.L., Chen, T., Lin, X.S., Tang, D. (2019). A marked Cox model for the number of IBNR claims: estimation and application. *ASTIN Bulletin* **49/3**, 709-739.

[4] Baudry, M., Robert, C.Y. (2019). A machine learning approach for individual claims reserving in insurance. *Applied Stochastic Models in Business and Industry* **35/5**, 1127-1155.

[5] Bischofberger, M.S., Hiabu, M., Isakson, A. (2019). Continuous chain-ladder with paid data. *Scandinavian Actuarial Journal* **2019/10**, 1-26.

[6] Breiman, L. (2001). Statistical modeling: the two cultures. *Statistical Science* **16/3**, 199-215.

[7] De Felice, M., Moriconi, F. (2019). Claim watching and individual claims reserving using classification and regression trees. *Risks* **7/4**, 102.

[8] Delong, Ł., Wüthrich, M.V. (2020). Neural networks for the joint development of individual payments and claim incurred. *Risks* **8/2**, 33.

[9] Duval, F., Pigeon, M. (2019). Individual loss reserving using a gradient boosting-based approach *Risks* **7/3**, 79.

[10] England, P.D., Verrall, R.J. (1999). Analytic and bootstrap estimates of prediction errors in claims reserving. *Insurance: Mathematics and Economics* **25/3**, 281-293.

[11] Gabrielli, A. (2020). A neural network boosted double overdispersed Poisson claims reserving model. *ASTIN Bulletin* **50/1**, 25-60.

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

[13] Jørgensen, B. (1987). Exponential dispersion models. *Journal of the Royal Statistical Society. Series B (Methodological)* **49/2**, 127-145.

[14] Kuo, K. (2019). Individual claims forecasting with Bayesian mixture density networks. Preprint.

[15] Larsen, C.R. (2007). An individual claims reserving model. *ASTIN Bulletin* **37/1**, 113-132.

[16] Lindholm, M., Verrall, R. (2019). On distribution-free reserving and partial information. Preprint.

[17] Lopez, O., Milhaud, X., Thérond, P.-E. (2019). A tree-based algorithm adapted to microlevel reserving and long development claims. *ASTIN Bulletin* **49/3**, 741-762.

[18] Maciak, M., Okhrin, O., Pešta, M. (2019). Infinitely stochastic micro forecasting. *arXiv*:1908.10636v2

[19] Martínez-Miranda, M.D., Nielsen, J.P., Verrall, R. (2012). Double chain ladder. *ASTIN Bulletin* **42/1**, 59-76.

[20] Martínez-Miranda, M.D., Nielsen, J.P., Wüthrich, M.V. (2012). Statistical modelling and forecasting of outstanding liabilities in non-life insurance. *SORT – Statistics and Operations Research Transactions* **36/2**, 195-218.

[21] Norberg, R. (1993). Prediction of outstanding liabilities in non-life insurance. *ASTIN Bulletin* **23/1**, 95-115.

[22] Norberg, R. (1999). Prediction of outstanding liabilities II. Model variations and extensions. *ASTIN Bulletin* **29/1**, 5-25.

[23] Okine, A.N.-A., Frees, E., Shi, P. (2019). Joint model prediction and application to individual-level loss reserving. Preprint.

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

[25] Rosenlund, S. (2012). Bootstrapping individual claim histories. *ASTIN Bulletin* **42/1**, 291-324.

[26] Smyth, G.K., Jørgensen, B. (2002). Fitting Tweedie's compound Poisson model to insurance claims data: dispersion modeling. *ASTIN Bulletin* **32/1**, 143-157.

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

[28] Ticconi, D. (2019). Individual claims reserving in creditor protection insurance using machine learning. *SSRN Preprint* ID 3458826.

[29] Tukey, J.W. (1977). *Exploratory Data Analysis*. Addison-Wesley, Reading, MA.

[30] Wahl, F., Lindholm, M., Verrall, R. (2019). The collective reserving model. *Insurance: Mathematics and Economics* **87**, 34-50.

[31] Wüthrich, M.V. (2003). Claims reserving using Tweedie's compound Poisson model. *ASTIN Bulletin* **33/2**, 331-346.

[32] Wüthrich, M.V. (2018). Machine learning in individual claims reserving. *Scandinavian Actuarial Journal* **2018/6**, 465-480.

[33] Wüthrich, M.V. (2019). From generalized linear models to neural networks, and back. *SSRN Preprint* ID 3491790.

[34] Zhao, X.B., Zhou, X., Wang, J.L. (2009). Semiparametric model for prediction of individual claim loss reserving. *Insurance: Mathematics and Economics* **45**, 1-8.

A Graphical illustrations

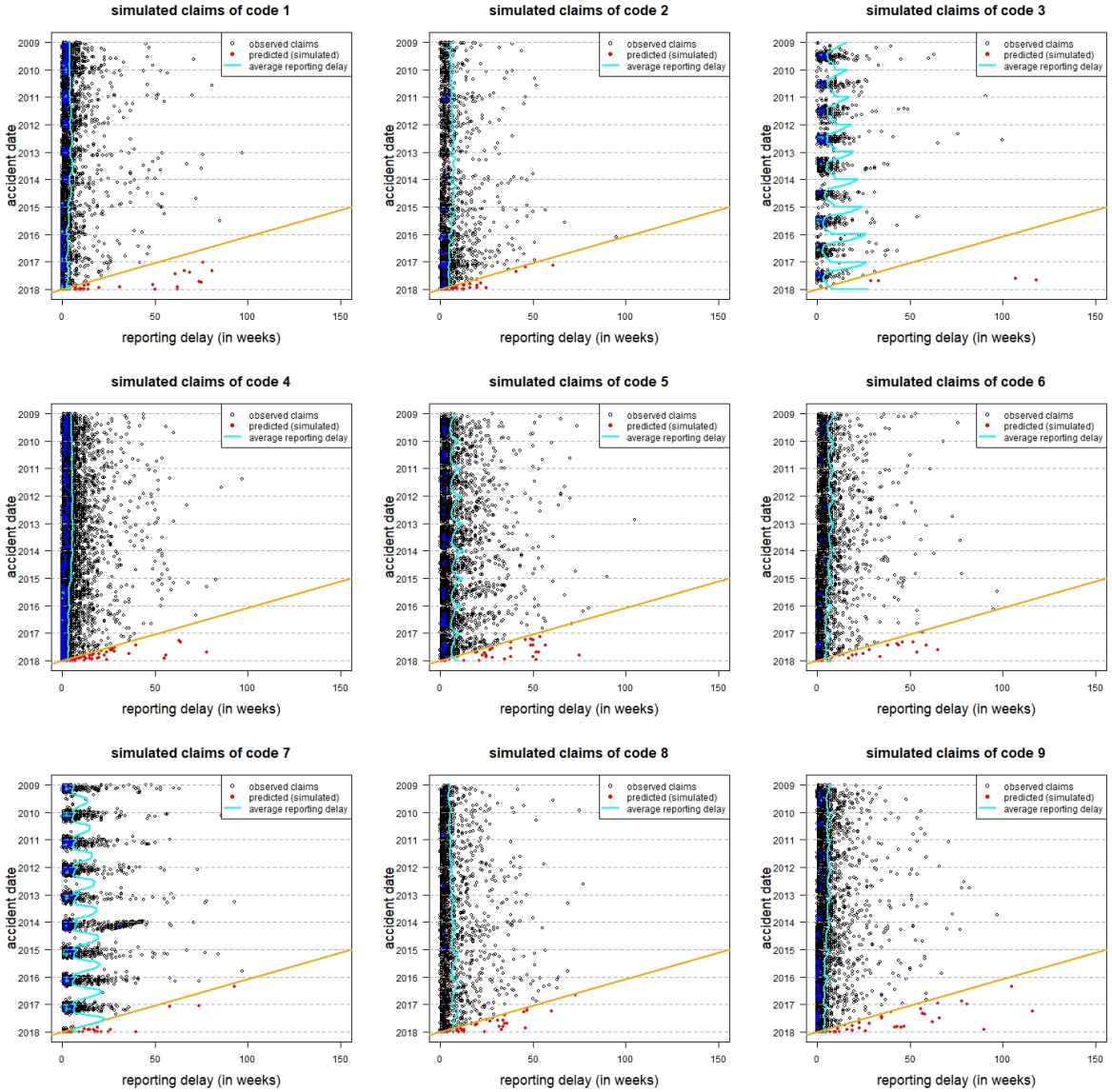

Figure 21: Black,blue and green dots: observed reported claims in the upper triangle at the cutoff date τ (above orange line); red dots: predicted claims in the lower triangle for the different claim codes 1-9; cyan lines estimated average reporting delays per occurrence date.

B Code

Listing 2: Keras code for claim occurrence modeling.

```
library(keras)
#
design  <- layer_input(shape = c(4), dtype = 'float32', name = 'design')
code    <- layer_input(shape = c(1), dtype = 'int32', name = 'code')
#
codeEmb = code %>%
        layer_embedding(input_dim = 9, output_dim = 2, input_length = 1, name = 'codeEmb') %>%
        layer_flatten(name='code_flat')
#
output = list(design, codeEmb) %>% layer_concatenate(name='concat') %>%
        layer_dense(units=20, activation='tanh', name='layer1') %>%
        layer_dense(units=15, activation='tanh', name='layer2') %>%
        layer_dense(units=10, activation='tanh', name='layer3') %>%
        layer_dense(units=1, activation='exponential', name='output')
#
model <- keras_model(inputs = list(design, code), outputs = c(output))
#
CBs <- callback_model_checkpoint("path", monitor = "val_loss", verbose = 0,
        save_best_only = TRUE, save_weights_only = TRUE)
#
model %>% compile(loss = 'poisson', optimizer = 'nadam')
fit <- model %>% fit(list(designNN, codeNN),  NN, validation_split=0.2,
        batch_size=5000, epochs=1000, verbose=0, callbacks=CBs)
```

Listing 3: Bias regularization step for claim counts.

```
z.layer  <- keras_model(inputs=model$input, outputs=get_layer(model, 'layer3')$output)
dat[,c("z1","z2","z3","z4","z5","z6","z7","z8","z9","z10")] <-
               data.frame(z.layer %>% predict(list(designNN, codeNN)))
dat$NN <- NN
glm(NN ~ z1 + z2 + z3 + z4 + z5 + z6 + z7 + z8 + z9 + z10, data=dat, family=poisson())
```

Listing 4: Bias regularization step for claim indicator process.

```
z.layer  <- keras_model(inputs=model$input, outputs=get_layer(model, 'layer3')$output)
dat[,c("z1","z2","z3","z4","z5","z6","z7","z8","z9","z10")] <-
               data.frame(z.layer %>% predict(list(designNN, codeNN)))
dat$NN <- as.factor(NN)
multinom(NN ~ z1 + z2 + z3 + z4 + z5 + z6 + z7 + z8 + z9 + z10, data=dat, maxit=500)
```