

Article

Advancing the Use of Deep Learning in Loss Reserving: A Generalized DeepTriangle Approach

Yining Feng¹ and Shuanming Li^{2,*} ¹ AXA, 20 Gracechurch Street, London EC3V 0BG, UK; lucy.feng@axa-uk.co.uk² Centre for Actuarial Studies, Department of Economics, The University of Melbourne, Melbourne, VIC 3010, Australia* Correspondence: shli@unimelb.edu.au

Abstract: This paper proposes a generalized deep learning approach for predicting claims developments for non-life insurance reserving. The generalized approach offers more flexibility and accuracy in solving actuarial reserving problems. It predicts claims outstanding weighted by exposure instead of loss ratio to remove subjectivity associated with premium weighting. Chain-ladder predicted outstanding claims are used as part of the multi-task learning to remove the dependence on case estimates. Grid-search is introduced for hyperparameter tuning to improve model performance. Performance-wise, the Generalized DeepTriangle outperforms both traditional chain-ladder methodology, the automated machine learning approaches (AutoML), and the original DeepTriangle model.

Keywords: loss reserving; actuarial reserving techniques; machine learning; deep learning; DeepTriangle; artificial neural networks

1. Introduction

Loss reserving is the process of estimating the reserve an insurer should hold to meet the future claims payments arising from policies which it has under-written. Insurers underwrite risks and receive premiums to cover claims arising over a specified period. The amount and timing of claim payments are uncertain, and so the insurer is required to set aside sufficient reserves to meet these obligations as and when they fall due. An insurer mitigates the risks to an extent by pooling similar risks. However, there is still uncertainty regarding the timing and quantum of payments, which may cause liquidity strain for the insurer. The failure to generate sufficient liquid assets to meet liabilities in a timely manner will affect business continuity. Therefore, it is crucial for insurers to project future claims payments and estimate the associated volatility in an accurate manner.

The accurate projection of future claims liabilities is important for numerous aspects of an insurer's operations. From a pricing perspective, an understanding of the expected amount and timing of future claims liabilities enables more precise technical pricing. This allows an insurer to price risks more appropriately and improves its competitiveness within the market. From a reserving perspective, being able to more accurately project future claims will reduce uncertainty and risk margin, which is an amount or margin reflecting an assessment of uncertainty associated with insurance risk ([Risk Margin Working Group 2009](#)). From a capital perspective, greater accuracy in claims projections will enable better allocation of capital to its most appropriate use. Therefore, loss reserving is critical for an insurer as it plays a vital role in informing underwriting, pricing, capital, and planning decisions. For shareholders, reserving and related items form a material portion of an insurer's financial statements. Mis-reserving constitutes insurance/actuarial risk, which leads to increased capital requirements (for example, in the Solvency II regime in the European Union). Under-reserving will have a direct impact on an insurer's profitability. However, over-reserving is also problematic, as capital is not directed to its most appropriate use to



Citation: Feng, Yining, and Shuanming Li. 2024. Advancing the Use of Deep Learning in Loss Reserving: A Generalized DeepTriangle Approach. *Risks* 12: 4. <https://doi.org/10.3390/risks12010004>

Academic Editor: Shengkun Xie

Received: 31 October 2023

Revised: 11 December 2023

Accepted: 23 December 2023

Published: 26 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

generate returns. The regulators are also very interested in the sufficiency of reserves to ensure business continuity and the protection of policyholders.

The amount and timing of a claim are highly uncertain for several reasons. Firstly, there is a delay between when an event leading to a loss occurs and the notification of the event to the insurer. The claim may also develop over time, leading to multiple losses being generated. Further delay exists between claim notification, assessment, and settlement. The amount of payment varies depending on the development of the claim over time.

Traditional reserving approaches developed to estimate future claims liabilities are largely deterministic, including the chain-ladder and Bornhuetter–Ferguson techniques (Bornhuetter and Ferguson 1972). Stochastic methodologies linked to the chain-ladder technique have also been developed to better estimate loss reserve variability. These include the chain-ladder approach in Mack (1993) and the bootstrap method in England and Verrall (1999).

With advancements in computer processing, machine learning approaches are increasingly adopted to solve problems for which large quantities of data are available. Predictive modeling from generalized linear models (Haberman and Renshaw 1996) to machine learning techniques (Gao et al. 2019) have been widely explored and applied in insurance. For insurance reserving, non-parametric individual claim reserving using decision trees is first explored in Baudry and Robert (2017). Wüthrich refines Mack’s chain-ladder method using neural networks (2018). More recently, Kuo (2019) proposes a novel approach to loss reserving based on deep neural networks in the form of DeepTriangle. The deep neural network approach in Kuo (2019) jointly models reserving paid losses and outstanding claims with minimal feature engineering. The model has shown improvements in predictive accuracy (as measured by the root mean squared percentage error and mean absolute percentage error) compared to existing stochastic methods across multiple lines of business.

This paper builds on the loss reserving approach in Kuo (2019) and generalizes the DeepTriangle for non-life insurance reserving. The generalized approach offers more flexibility and accuracy in solving actuarial reserving problems than existing techniques. It predicts claims outstanding weighted by exposure instead of loss ratio to remove subjectivity associated with premium weighting. Chain-ladder predicted outstanding claims are used as part of the multi-task learning to remove the dependence on case estimates. Enhancements to the categorical embedding component of the model architecture may further enhance model accuracy. Grid-search is introduced for hyperparameter tuning to improve model performance. The performance of the generalized approach is compared to traditional Chain-ladder, AutoML, and the original DeepTriangle. Results show that the Generalized DeepTriangle approach outperforms the traditional and existing machine learning methods.

The rest of the paper is organized as follows: Section 2 describes the evolution of actuarial reserving methods over time leading up to this paper, and Section 3 describes our generalized model architecture. Section 4 describes the dataset used, details the evaluation metrics for assessing model performance, and discusses results. Lastly, Section 5 concludes this paper and suggests potential future developments.

2. Related Work, Notation, and Terminologies

This section describes the evolution of reserving methods leading up to our paper. It also introduces the notation and terminology associated with actuarial reserving. Note that only a high-level description of reserving methods relevant to this paper is provided. For a comprehensive overview of the development of reserving approaches over time, refer to Carrato and Visintin (2019).

2.1. The Chain-Ladder Method on Cumulative Data

The most common reserving approach for estimating the ultimate cost in non-life insurance is the chain-ladder approach. The chain-ladder method in Mack (1993) is often considered as a fundamental form of the approach. It forecasts future claims development

based on historical cumulative claims development aggregated by accident and development periods. A distribution-free formula for evaluating the standard error of chain-ladder reserve estimates is also derived.

Let $P_{i,j}$ be the incremental claim paid for accident year i and development year j , for $0 \leq i \leq I$ and $0 \leq j \leq I$ and let $C_{i,j}$ be the cumulative claim of accident year i and up to development year j . Then $C_{i,t} = \sum_{j=0}^t P_{i,j}$. It is assumed that there exists:

- developing factors f_0, f_1, \dots, f_{I-1} such that

$$E(C_{i,j}|C_{i,0}, C_{i,1}, \dots, C_{i,j-1}) = E(C_{i,j}|C_{i,j-1}) = C_{i,j-1}f_{j-1}, \tag{1}$$

- and $\sigma_0^2, \sigma_1^2, \dots, \sigma_{I-1}^2$ such that

$$\text{Var}(C_{ij}|C_{i,0}, C_{i,1}, \dots, C_{i,j-1}) = \text{Var}(C_{i,j}|C_{i,j-1}) = C_{i,j-1}\sigma_{j-1}^2. \tag{2}$$

Mack (1993) proposes the following estimators:

$$\hat{C}_{i,I} = C_{i,I-i} \prod_{j=I-1}^{I-i} \hat{f}_{j-1}, \tag{3}$$

$$\hat{f}_{j-1} = \frac{\sum_{i=0}^{I-j} C_{i,j}}{\sum_{i=0}^{I-j} C_{i,j-1}}, \quad j = 1, 2, \dots, I, \tag{4}$$

$$\hat{\sigma}_{j-1}^2 = \frac{1}{I-j} \sum_{i=0}^{I-j} C_{i,j-1} \left(\frac{C_{i,j}}{C_{i,j-1}} - \hat{f}_{j-1} \right)^2, \quad j = 1, 2, \dots, I-1, \tag{5}$$

where an estimator for σ_{I-1}^2 may be obtained by means of extrapolation.

2.2. Regression on Individual Loss Data

The chain-ladder approach is a reserving method based on aggregated claims experience by accident and development periods. Regression based on individual loss data first proposed by Norberg (1993) and Hesselager (1994) enables more granular data to be used for predicting future claims developments.

Let n_i be the number of claims for accident year i . Denote $C_{i,j}^h, h = 1, 2, \dots, n_i$, the cumulative payment up to time $i + j$ of the h -th claim of accident year i . The total cumulative payment up to time $i + j$ for the accident year i is:

$$C_{i,j} = \sum_{h=1}^{n_i} C_{i,j}^h. \tag{6}$$

Therefore, for an individual claim, the following equation holds true:

$$E\left(C_{i,j}^h|C_{i,j-1}^h\right) = f_{j-1}C_{i,j-1}^h. \tag{7}$$

Then, the following estimator can be used to predict the ultimate loss:

$$\hat{C}_{i,j}^h = C_{i,j-1}^h \hat{f}_{j-1}. \tag{8}$$

2.3. Clustering on Individual Loss Data

The chain-ladder model in Mack (1993) assumes that claims are homogeneous, which does not always hold for an entire population in practice. To address this, clustering of claims into homogeneous groups is proposed, assuming that a linear model is applicable

for each group of claims. Let K be the total number of clusters for a portfolio, the total cumulative claim payment up to time $i + j$ for accident year i and cluster k is:

$$C_{i,j}^k = \sum_{h=1}^{n_i} C_{i,j}^{k_h}, \quad (9)$$

where k_h represents the h -th claim which belongs to the k -th cluster in calendar year $i + j$, and the total cumulative payment up to time $i + j$ for accident year i is:

$$C_{i,j} = \sum_{k=1}^K C_{i,j}^k. \quad (10)$$

Therefore, for each cluster, the following equation holds true:

$$E\left(C_{i,j}^{k_h} | C_{i,j-1}^{k_h}\right) = f_{j-1}^k C_{i,j-1}^{k_h}. \quad (11)$$

The following estimator can be used to predict the ultimate loss:

$$\hat{C}_{i,j}^{k_h} = \hat{f}_{j-1}^k C_{i,j-1}^{k_h}, \quad (12)$$

where \hat{f}_{j-1}^k has the similar definition for \hat{f}_j in (4), but with C_{ij} being replaced by C_{ij}^k .

Individual claim reserving models using large amount of granular information sit on the opposite end of the spectrum of loss reserving approaches to aggregate reserving methods like the chain-ladder, which uses relatively limited data. Clustering enables the forecasting of claims reserves at a segment level, balancing the granularity of reserving at an individual level with the reduced volatility of aggregate reserving approaches.

2.4. Dual Input Paid-Incurred Model on Individual Loss Data

Incurred claims cost is the sum of the paid, to-date amount and case estimates on open claims. The inclusion of case estimates in future claims prediction is often beneficial as it allows for situations where few payments have been made but are expected in the future. Hence, joint models accounting for both paid-to-date and incurred costs increase accuracy.

In addition to cumulative claims paid, incurred claim amounts can also be included as an input to the modeling. Let K be the total number of clusters of the portfolio. The total incurred claim payment $I_{i,j}^k$, for accident year i , in calendar year $i + j$ for cluster k is:

$$I_{i,j}^k = \sum_{h=1}^{n_i} I_{i,j}^{k_h}, \quad (13)$$

where k_h indicates the h -th incurred claim for the k -th cluster in calendar year $i + j$.

The following estimator can be used to predict incurred loss:

$$\hat{I}_{i,j}^{k_h} = \hat{f}_{j-1}^k I_{i,j-1}^{k_h}. \quad (14)$$

However, not all lines of business have case reserves, so the approach is not universally applicable.

2.5. Artificial Neural Networks (ANN)

Advancements in artificial intelligence and machine learning have led to novel and modern approaches of solving actuarial problems through big data. [Wüthrich \(2018\)](#) proposes the application of neural networks to the chain-ladder reserving.

The DeepTriangle architecture in [Kuo \(2019\)](#) uses a feed-forward network with fully connected layers; see the illustration in Figure 1. Output y is predicted from the input vector x . Hidden layers, as represented by $h_j^{[l]}$ transform the input into representations

which gradually increase in the predictive power of the output as we move across each layer $l \in \{1, \dots, L\}$. Each node $h_j^{[l]}$ is computed iteratively as:

$$h_j^{[l]} = g^{[l]}(w_j^{[l]\top} h^{[l-1]} + b_j^{[l]}), \quad l = 1, 2, \dots, L, j = 1, 2, \dots, n^{[l]},$$

where L represents the total number of layers, $n^{[l]}$ represents the total components of the l -th layer, $g^{[l]}$ is the activation function which is chosen to be nonlinear, $h^{[l]} = (h_1^{[l]}, h_2^{[l]}, \dots, h_{n^{[l]}}^{[l]})^\top$ is the activation column vector, $w_j^{[l]\top}$ is the row weights vector, and $b_j^{[l]}$ is the biases scalar.

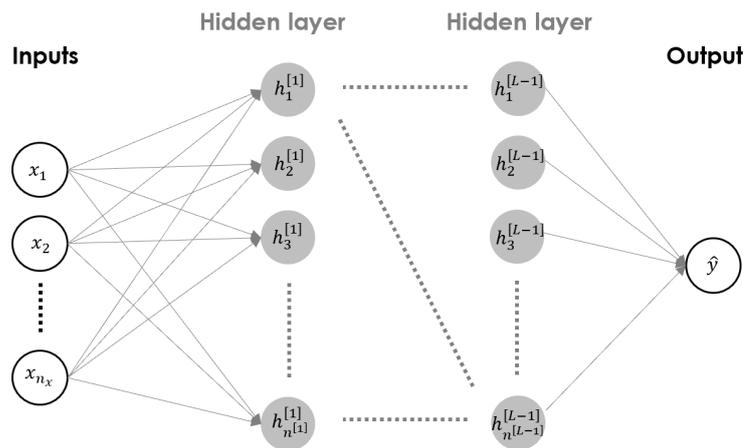


Figure 1. Feedforward neural network.

Conventionally, $h^{[0]} = x$ and $\hat{y} = h^{[L]}$. The weights and biases are the parameters of the neural network learned during training. They are selected by the neural network to maximize prediction accuracy.

The chain-ladder factors for artificial neural networks are found by minimizing a given appropriate loss function. Each development period j has its own neural network architecture to be optimized with respect to the loss function. The loss function is used to measure how close the model predictions are to the actual values.

2.6. DeepTriangle

Kuo (2019) proposed the DeepTriangle as a novel approach for loss reserving based on the deep neural network described in Section 2.5. It jointly models paid losses and claims outstanding stated in Section 2.3 and incorporates heterogeneous inputs in Section 2.2. The key components of the model architecture are described below.

2.6.1. Sequence-to-Sequence Architecture

The architecture uses a class of algorithms called sequence-to-sequence learning (Sutskever et al. 2014). Instead of relying on single data points, the model takes a sequence of ordered events as input and predicts a sequence into the future, making it suitable for reserving claim development predictions.

We have previously defined $P_{i,j}$ to be the incremental claims paid. Here we define $OS_{i,j}$ to be the total claims outstanding for accident year i and development year j where $1 \leq i \leq I$ and $1 \leq j \leq I$. Then, at the end of calendar year I , we have access to the observed data

$$\{P_{i,j} : i = 1, \dots, I, j = 1, \dots, I - i + 1\},$$

$$\{OS_{i,j} : i = 1, \dots, I, j = 1, \dots, I - i + 1\}.$$

Then

$$UL_i = \sum_{j=1}^{I-i+1} P_{i,j} + \sum_{j=I-i+2}^I OS_{i,j} \tag{15}$$

is the ultimate loss for accident year $i = 1, 2, \dots, I$, which can be estimated by

$$\widehat{UL}_i = \sum_{j=1}^{I-i+1} P_{i,j} + \sum_{j=I-i+2}^I \widehat{OS}_{i,j}. \tag{16}$$

The gated recurrent unit (GRU) in Chung et al. (2014) is used to process the paid losses and claims outstanding sequences. Here, we use the notation as in Kuo (2019) and define the activation function $h^{<t>}$ at time t as follows:

$$h^{<t>} = \Gamma_u^{<t>} \tilde{h}^{<t>} + (1 - \Gamma_u^{<t>}) h^{<t-1>}, \tag{17}$$

where:

- $\tilde{h}^{<t>} = \tanh(W_h [\Gamma_u^{<t>} h^{<t-1>}, x^{<t>}] + b_h)$,
- $x^{<t>}$ represents the input values,
- $\Gamma_u^{<t>} = \sigma(W_u [h^{<t-1>}, x^{<t>}] + b_u)$,
- $\Gamma_r^{<t>} = \sigma(W_r [h^{<t-1>}, x^{<t>}] + b_r)$,
- $\sigma(x) = \frac{1}{1 + \exp(-x)}$ represents the logistic sigmoid function,
- W_r, W_u, W_h represent weight matrices,
- and b_r, b_u, b_h represent biases to be learnt.

Each activation function $h^{<t>}$ retains values from earlier values of the input sequence and gives a certain weight $\Gamma_u^{<t>}$ to the estimated current state $\tilde{h}^{<t>}$ and the previous state $h^{<t-1>}$.

2.6.2. Multi-Task Learning

DeepTriangle simultaneously models two sequences as input and two as output. This means that one task can reuse insights derived from the other. Kuo (2019) proposes the use of paid losses and case reserve by accident and development year as the dual input sequences. Kuo (2019) defines the two sequences of inputs and outputs as:

$$Y_{i,j}, Y_{i,j+1}, \dots, Y_{i,I-i+1},$$

where $Y_{i,j} = (P_{i,j}/NPE_i, OS_{i,j}/NPE_i)$ and NPE_i represents the net earned premium for accident year i . Note that the model takes in and predicts loss ratios to normalize the inputs and outputs.

2.6.3. Categorical Embedding

Company codes are passed to an embedding layer, with each company represented by a vector in \mathbb{R}^{49} as in Guo and Berkhahn (2016). Company codes are mapped onto a multi-dimensional vector space, where segments with similar implicit behaviors are placed closer together. In other words, it implicitly finds the relationships between segments, serving as a proxy for company characteristics.

3. Model Architecture

Extensions to the existing DeepTriangle model architecture in Section 2.6 are introduced in this section to address current shortcomings and generalize the approach.

3.1. The Generalized DeepTriangle Approach

In this section, several components of the DeepTriangle model architecture in Kuo (2019) are modified to enable the generalization of the methodology for reserving problems and improve prediction accuracy. A comparison of results with five alternative methods,

including the chain-ladder and the AutoML, shows that the generalized DeepTriangle method outperforms existing reserving methodologies.

3.1.1. Chain-Ladder Predicted Claims Outstanding Sequences for Multi-Task Learning

DeepTriangle uses incremental paid and total claims outstanding as input and output sequences. Claims outstanding is determined based on individual case reserves. This is only possible for portfolios where case estimates exist.

For lines of business with no case estimates, we propose the use of the chain-ladder approach to generate total claims outstanding sequences, thereby generalizing the approach. As the model does not pre-suppose a model structure, the use of chain-ladder estimates as part of the input sequence is useful in guiding the claims predictions, essentially giving weight to the importance of development factors.

For the purpose of our modeling, we have evaluated development factors based on the notation developed in Section 2.1. The chain-ladder factors for each development year j is determined as the sum of the cumulative payment for development year j over the sum of the cumulative payment for development year $j - 1$, across the most recent three accident years up to the evaluation date.

3.1.2. Exposure-Weighted Inputs and Targets

In Section 2.6, we noted that the model takes loss ratios (total claims paid over premium earned) as inputs and outputs for normalization purposes. However, loss ratio prediction is less desirable compared to modeling claims paid. [Werner and Guven \(2007\)](#) demonstrate the need to use exposure (sum in years of in-force policies) as the weight for claims projections instead of premium. A summary is provided below:

- Loss ratio needs to be calculated on the current rate level and re-calculated every time underwriting rules are changed. This is likely to be extremely difficult for many companies.
- The distribution of the loss ratio varies depending on the rating structure of each company and does not follow a typical error structure, making it difficult to model accurately.
- The exercise of judgement is possible when using exposure-weight, e.g., trends by age curve. Loss ratios are expected to be the same if and only if the rates are perfect.
- Loss ratio models become obsolete once changes to the rating structure are implemented, meaning that prior experience cannot be used as a starting point for later reviews.

3.1.3. Feature Selection and Optimization for Categorical Embedding

The original DeepTriangle architecture utilizes company code as input for categorical embedding. We examine the result of passing claim code as an alternative categorical input into the embedding layer. We then compare the output against a portfolio-level prediction, i.e., one without categorical embedding, to assess the benefit of the embedding layer.

We note that as a further extension, principal component analysis can be used on key categorical variables to determine the optimal segmentation to feed into the embedding layer. Alternatively, the model architecture can be modified to embed multiple categorical variables.

3.1.4. Grid Search for Hyperparameter Optimization

A model's parameter is an internal characteristic of the model. Its value can be estimated from the data. In contrast, a model's hyperparameter is an external characteristic whose value cannot be estimated based on the data. Therefore, the value of the hyperparameter needs to be pre-set prior to the model setup ([Joseph 2018](#)).

Grid search is a traditional method of hyperparameter optimization. It completes a full search over a given subset of the hyperparameters' space on the training set to find the most

appropriate hyperparameters for the model build, see Liashchynskiy and Liashchynskiy (2019).

Grid search can improve model performance on several hyperparameters within the ANN model architecture. We specifically focus on optimizing the batch size. In the original DeepTriangle model architecture, the batch size is set to 250. However, depending on the categorical embedding adopted, the optimal batch size parameter differs. We aim to better understand the impact of batch size on model performance.

Due to computer processing power and computation time limitations, we have only conducted hyperparameter optimization on the batch size. Expanding the grid-search process to other hyperparameters is recommended for future iterations, including, for example, the ANN's hyperparameters, the encoder's hyperparameters, and the activation function used.

3.2. The Generalized DeepTriangle Model Architecture

Figure 2 outlines the generalized DeepTriangle model architecture. The generalized approach adopts a sequence-to-sequence architecture, taking in sequences of ordered input variables and predicting sequences of outputs across time steps. The architecture uses multi-task learning as explained in Section 3.1.

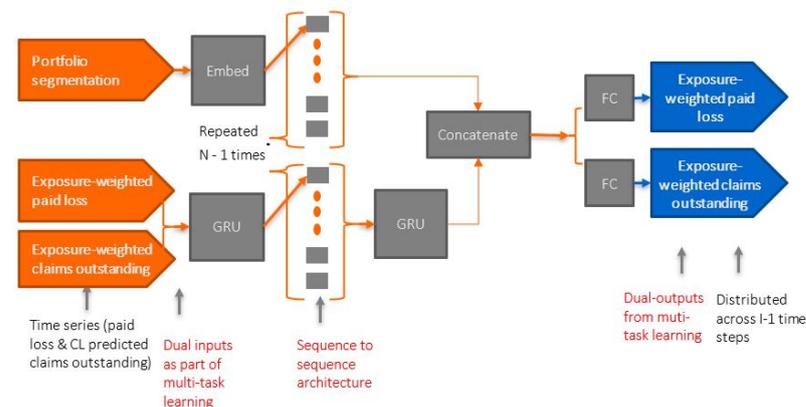


Figure 2. Generalized DeepTriangle model architecture.

The input and output sequences are the incremental claims paid $P_{i,j}$ and total chain-ladder predicted claims outstanding $OS_{i,j}$, for accident year i ($1 \leq i \leq I$) and development year j ($1 \leq j \leq I$).

The input sequence can be represented by:

$$\{Y_{i,j} : i = 1, \dots, I, j = 1, \dots, I - i + 1\},$$

where $Y_{i,j} = (P_{i,j}/E_i, OS_{i,j}/E_i)$ and E_i represents the total exposure in accident year i . The response sequence can be represented by:

$$\{Y_{i,j} : i = 1, \dots, I, j = I - i + 2, \dots, I\}.$$

A categorical variable is separately fed into the embedding layer, enabling segmentation. The implicit relationships between the categorical segments are then modeled. For our dataset, we examine the use of claim code as input into the categorical embedding layer. Separately, we have also examined the impact of no segmentation on the model's prediction accuracy. A grid search is performed on the batch size to optimize the hyperparameter for different levels of granularity for the embedded categorical variables.

The input sequence is encoded with a GRU such that a summary encoding is obtained. This is repeated $I - 1$ times with the output of the initial encoding before it is decoded via a decoder GRU, where $I - 1$ represents the timesteps into the future for which the forecast is required. As in Kuo (2019), we define the following hyperparameters for the encoder:

128 hidden units and a dropout rate of 0.2, and the following for the decoder: 64 hidden units and a dropout rate of 0.2. For both, the rectified linear unit (ReLU) activation in [Nair and Hinton \(2010\)](#) is used, defined by $g(x) = \max(0, x)$. Recoveries are removed from the claims dataset as the activation function used results in non-negative predictions.

4. Data, Experiments, and Results

This section first details the data source and data pre-processing and then describes the evaluation metrics used to assess the model performance against the benchmark models before illustrating the results.

4.1. Data

[Kuo \(2019\)](#) uses the National Association of Insurance Commissioners (NAIC) Schedule P dataset ([Meyers and Peng 2019](#)). The dataset includes claims over accident years 1988–1997 and 10 development years for each accident year. Schedule P data are aggregated by accident year, development year by line of business, and group code. It includes both aggregated premium and claims information. However, Schedule P data have the following two limitations:

- The dataset does not include information on the number of lives or policy start and end dates, meaning that it is not possible to use exposure years as a weight
- The dataset is aggregated with only line of business and company code segmentations, making it difficult to conduct modeling and analysis at a more granular level. This also limits our ability to understand the drivers of the experience

Extensive research has been conducted into publicly available insurance data sources for the most suitable dataset. The individual claims history simulation machine in [Gabrielli and Wüthrich \(2018\)](#) produces insurance datasets that are more suitable and addresses the limitations of Schedule P data. [Gabrielli and Wüthrich \(2018\)](#) developed a stochastic simulation machine that generates individual claims histories of non-life insurance claims. The simulation machine enables users to simulate a synthetic insurance portfolio of individual claims histories based on real non-life insurance data.

The final dataset is a simulated dataset that corresponds to claims over accident years 1994 to 2005, with over 12 development years of experience. It contains the feature information for each claim in [Table 1](#).

Table 1. Feature information on individual claims history simulation machine.

Field	Label	Description
Claim Number	CINr	Unique claim identifier
Line of Business	LoB	Categorical with labels in $\{1, 2, 3, 4\}$
Claim Code	CC	Categorical with labels in $\{1, 2, \dots, 53\}$ denoting the claimant's labor sector
Accident Year	AY	Numeric set in $\{1994, 1995, \dots, 2005\}$
Accident Quarter	AQ	Numeric set in $\{1, 2, 3, 4\}$
Age	age	Age of the claimant in 5 year buckets: $\{15, 20, \dots, 70\}$
Injured Part	IP	Categorical with labels in $\{10, 11, \dots, 99\}$ denoting the injured body part
Reporting Year	RY	Numeric set in $\{1994, 1995, \dots, 2016\}$

The benefits of the simulation machine dataset include:

- The dataset is at an individual claim line level, enabling exposure-weighting. It also offers more flexibility in the level of granularity used for modeling

- The existence of multiple feature information (line of business, claims code, age, and injury part) offers more information on the claims and enables more granular segmentation

4.2. Data Processing

Table 2 outlines the parameters adopted to simulate the individual claims dataset for our analysis. The only potential limitation for such parameter adoptions is not varying the claims volatility, but this was a deliberate decision so as not to add further complexity when interpreting model results, given that each LOB already has its intrinsic characteristics within its data.

Table 2. Parameters adopted for simulating the non-life individual claims history dataset.

Parameter	Value	Description
Claim number (num_claims)	100,000	Expected total number of claims in the simulation output
Distribution for LOB (lob_distribution)	(0.25, 0.25, 0.25, 0.25)	Simulated claims evenly distributed across the four lines of business
Inflation	(0.03, 0.01, 0.01, 0.01)	A different inflation parameter is adopted for LOB 1 to enable assessment of inflation impact on predicted outputs
Claim volatility (sd_claim)	0.5	Volatility of claim amount (default parameter adopted)

Recovery payments have been excluded from the input dataset as the model adopts an activation function that predicts nonnegative cash flows.

The individual claims dataset is aggregated for the purpose of this paper. Aggregation is performed by accident year, development lag, line of business, and claims code, with a number of claims, exposure years, and paid losses summarized for analysis. We have separately repeated the model by lines of business only to understand the impact of segmentation on model predictiveness. More details on the methodology are provided in the following subsection.

We have split the data into the following segments for model prediction and validation:

- Training set: calendar years 1994–2002
- Validation set: 2003–2004
- Test set: 2005+

We assess the model predictiveness based on cumulative predicted payments for development year 10.

4.3. Performance Evaluation Metrics

A range of validation methods have been proposed for evaluating the performance of reserving models. This paper uses the Mean Absolute Percentage Error (MAPE) and Root Mean Square Percentage Error (RMSPE) in the model evaluation process. MAPE and RMSPE are adopted for consistency with Kuo (2019). Percentage errors enable unit-free measurement over each segment. In this case, the segment is defined by the categorical variable passed through the embedding layer. The actual and predicted cumulative ultimate losses as at development year 10 by segment are compared to evaluate model performance.

For line of business l ,

$$\text{MAPE}_l = \frac{1}{|C_l|} \sum_{c=1}^{C_l} \left| \frac{\widehat{UL}_c - UL_c}{UL_c} \right|, \quad (18)$$

and

$$\text{RMSPE}_l = \sqrt{\frac{1}{|C_l|} \sum_{c=1}^{C_l} \left(\frac{\widehat{UL}_c - UL_c}{UL_c} \right)^2}, \tag{19}$$

where

- $\{c : 1 \leq c \leq C_l\}$, represents the count of all possible values that the categorical variable can take and, C_l is the set of possible levels which the categorical input can take, $|C_l|$ is the number of elements in C_l , and
- UL_c and \widehat{UL}_c are the actual and predicted cumulative ultimate loss for the c^{th} categorical variable as at development year 10.

4.4. Benchmark Models

To assess the performance of the generalized DeepTriangle approach, the model’s MAPE and RMSPE are compared against that for the chain-ladder method in Mack (1993) and the AutoML model adopted in the original DeepTriangle approach in Kuo (2019).

The chain-ladder method in Mack (1993) enables a comparison to traditional, judgement-free reserving technique. The AutoML model enables a comparison of model performance against alternative machine learning techniques, which is developed through automated searches over common machine learning techniques. It is trained over an ensemble involving a random forest, an extremely randomized forest, and a random grid of gradient boosting machines, a random grid of deep feedforward neural networks (H2O.ai 2018). An iterative forecasting scheme is used to predict each timestep.

4.5. Parameterization and Implementation

Table 3 below details the key model parameters used for training the model.

Table 3. Model parameters.

Parameter \ Segmentation	Portfolio Level	By Claim Code
Batch-size for grid-search	2^i , where $i = 1, \dots, 6$	2^i , where $i = 5, \dots, 10$
Learn rate	0.0005	0.0005
Maximum epoch	1000	1000
Early stopping	200	200

We use the average mean squared error over the forecasted time steps as the loss function of the prediction. For each accident and development year set (i, j) , the per-sample loss function is defined as:

$$\frac{1}{I - i + 1 - (j - 1)} \sum_{k=j}^{I-i+1} \frac{(\widehat{P}_{i,k} - P_{i,k})^2 + (\widehat{OS}_{i,k} - OS_{i,k})^2}{2}. \tag{20}$$

The model is implemented using the following open source keras R packages (Chollet and Allaire 2017) and TensorFlow (Abadi et al. 2015).

We create an ensemble of 10 models trained with the same model architecture but different initial seeds. We take the average predicted ultimate claims at development year 10 for performance evaluation. This is accomplished to reduce the variation in predicted targets associated with neural network models. Note that increasing the number of models will lead to further variance reductions but requires a longer training time.

4.6. Results and Discussions

We have applied the benchmark models and the generalized DeepTriangle architecture to predict ultimate claims payment.

Table 4 provides a comparison of model performance here DeepTriangle (Kuo 2019) is the ultimate claims prediction using the original DeepTriangle methodology, Generalized DeepTriangle (aggregated) is the prediction at an aggregate level, meaning without claim code segmentation; generalized DeepTriangle is the prediction using claim code categories as categorical embedding. It can be seen that the generalized DeepTriangle outperforms the benchmark models both at a portfolio level and across each line of business.

Table 4. Performance comparison by batch sizes using MAPE - claim code as categorical embedding.

MAPE		Line of Business				
Model	Batch Size	1	2	3	4	All
Chain-ladder	-	6.49%	6.37%	6.83%	6.49%	6.54%
AutoML	-	5.87%	8.22%	6.76%	5.55%	6.60%
DeepTriangle (Kuo 2019)	250	5.44%	7.22%	11.11%	3.61%	6.84%
Generalized DeepTriangle (aggregated)	2	7.23%	7.02%	8.44%	7.06%	7.44%
Generalized DeepTriangle	32	2.23%	2.56%	4.88%	3.42%	3.27%

For the Generalized DeepTriangle (aggregated), the results using batch size 2 are used as they yield the best overall performance at an aggregate level. The results under batch size 32 are used for the generalized DeepTriangle for the same reason. It is worth noting that line of business 3 has lower exposure and greater volatility than the other lines of business. This has led to higher prediction uncertainties when using machine learning approaches compared to traditional Mack's chain ladder approach.

There is also an optimal range for batch size depending on the level of segmentation. Tables 5 and 6 compare MAPE by batch sizes. The optimal batch size for the aggregate prediction (between 2 and 8) is materially lower than for the more granular prediction by claim code (between 32 and 256). This is intuitive as larger batch sizes group more claim codes together, reducing the variance. Therefore, the addition of a grid search for hyperparameter optimization enhances model performance.

Table 5. Performance comparison by batch sizes—prediction at an aggregate level (no categorical embedding).

Batch size	2	4	8	16	32	64
MAPE	7.44%	7.47%	7.63%	10.37%	10.94%	7.44%

Table 6. Performance comparison by batch sizes—claim code as categorical embedding.

Batch size	16	32	64	128	256	512
MAPE	7.90%	3.27%	4.35%	4.19%	3.96%	5.47%

Analysis has also been performed on RMSPE on top of MAPE in Table 7, and it also yields better overall performance. However, due to RMSPE being a square error, it emphasizes uncertainty on more volatile portfolios and leads to better performance on less volatile portfolios. As can be seen on the volatilities for AutoML, Kuo's DeepTriangle, as well as the generalized approach, the MAPE enables a better comparison of results.

Table 7. Performance comparison by batch sizes using RMSPE- claim code as categorical embedding.

RMSPE Model	Batch Size	Line of Business				
		1	2	3	4	All
Chain-ladder	-	9.09%	10.17%	9.85%	9.09%	9.55%
AutoML	-	7.88%	10.96%	11.64%	5.55%	9.01%
DeepTriangle (Kuo 2019)	250	7.10%	10.39%	13.49%	4.84%	8.96%
Generalized DeepTriangle (aggregated)	2	9.47%	10.19%	9.71%	9.50%	9.72%
Generalized DeepTriangle	32	4.16%	4.14%	10.18%	4.48%	5.74%

5. Conclusions and Potential Further Extensions

This paper proposes some extensions to the DeepTriangle methodology developed in Kuo (2019) in several aspects as described in Section 3.

On a practical note, reserving requires significant regulatory oversight, making applications of machine learning techniques difficult. Not only does the result need to be accurate, but it also needs to be explainable and stable. Improving model interpretability and reducing volatility have been continued areas of research as we develop more advanced machine learning techniques.

To best enable advancement in this field, we need to develop both short-term applications as well as ongoing model improvements to make it usable in a corporate context. In the short term, the Generalized DeepTriangle can be used as a guide to supplement existing reserving methodologies. The Generalized DeepTriangle picks up on the subtler changes in claims behavior and claims profiles, which may be difficult to identify in a timely manner under traditional aggregated reserving approaches. Compared to other machine learning methods for predicting claims behaviors, the Generalized DeepTriangle is the closest in structure and more comparable to traditional reserving methods as it predicts by accident and development periods on historic claims experience. Therefore, it may supplement existing reserving methodologies and inform on reserving trends in a rapidly changing post-pandemic environment.

There is potential for further model enhancements. The first option is to conduct principal component analysis on key categorical variables to determine the optimal segmentation to feed into the embedding layer. Alternatively, the model architecture could be modified to embed multiple categorical variables.

Author Contributions: Methodology, Y.F.; Validation, Y.F.; Formal analysis, Y.F.; Resources, Y.F.; Writing—original draft, S.L.; Writing—review and editing, S.L.; Supervision, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset are obtained by simulation and are available on request from the corresponding author.

Conflicts of Interest: Author Yining Feng was employed by the company AXA UK. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The AXA UK had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

- Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, and et al. 2016. TensorFlow: Large-scale machine learning on heterogeneous systems. *arXiv*, arXiv:1603.04467.
- Baudry, Maximilien, and Christian Robert. 2017. Non Parametric Individual Claim Reserving in Insurance. *Preprint*. Available online: <http://www.ressources-actuarielles.net/EXT/ISFA/1226.nsf/0/6b3d579479584e35c12581eb00468777%24FILE/Reserving-article.pdf> (accessed on 11 February 2023).
- Bornhuetter, Ronald L, and Ronald E. Ferguson. 1972. The actuary and IBNR. *Proceedings of the Casualty Actuarial Society* 59: 181–95.

- Carrato, Alessandro, and Michele Visintin. 2019. From the chain ladder to individual claims reserving using machine learning techniques. *ASTIN Colloquium* 1: 1–19.
- Chollet, Francois, and Joseph. J Allaire. 2017. R Interface to Keras. GitHub, Inc. Available online: <https://github.com/rstudio/keras> (accessed on 11 February 2023).
- Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv*, arXiv:1412.3555.
- England, Peter, and Richard Verrall. 1999. Analytic and bootstrap estimates of prediction errors in claims reserving. *Insurance: Mathematics and Economics* 25: 281–93. [CrossRef]
- Gabrielli, Andrea, and Mario V. Wüthrich. 2018. An individual claims history simulation machine. *Risks* 6: 29. [CrossRef]
- Gao, Guangyuan, Shengwang Meng, and Mario V. Wüthrich. 2019. Claims frequency modelling using telematics car driving data. *Scandinavian Actuarial Journal* 2019: 143–62. [CrossRef]
- Guo, Cheng, and Felix Berkhahn. 2016. Entity embeddings of categorical variables. *arXiv*, arXiv:1604.06737.
- Haberman, Steven, and Arthur E. Renshaw. 1996. Generalized linear models and actuarial science. *The Statistician* 45: 407–36. [CrossRef]
- Hesselager, Ole. 1994. A Markov Model for Loss Reserving. *ASTIN Bulletin* 24: 183–93. [CrossRef]
- Joseph, Rohan. 2018. Grid Search for Model Tuning. Towards Data Science. Available online: <https://towardsdatascience.com/grid-search-for-model-tuning-3319b259367e> (accessed on 20 February 2023).
- Kuo, Kevin. 2019. DeepTriangle: A deep learning approach to loss reserving. *Risks* 7: 97. [CrossRef]
- Liashchynskiy, Petro, and Pavlo Liashchynskiy. 2019. Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS. *arXiv*, arXiv:1912.06059.
- Mack, Thomas. 1993. Distribution-free calculation of the standard error of chain ladder reserve estimates. *ASTIN Bulletin. The Journal of the International Actuarial Association* 23: 213–25. [CrossRef]
- Meyers, Glenn, and Shi Peng. 2019. Loss Reserving Data. NAIC Schedule P. Available online: http://www.casact.org/research/index.cfm?fa=loss_reserves_data (accessed on 20 February 2023).
- Nair, Vinod, and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, June 21; pp. 807–814.
- Norberg, Ragnar. 1993. Prediction of outstanding liabilities in non-life insurance. *ASTIN Bulletin* 23: 95–115. [CrossRef]
- Risk Margin Working Group. 2009. *Measurement of Liabilities for Insurance Contracts: Current Estimates and Risk Margins*. Ottawa: International Actuarial Association.
- Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems* 27: 3104–3112.
- The H2O.ai team. 2018. *h2o: R Interface for H2O*. R package version 3.0.9.8. Mountain View: H2O.ai.
- Werner, Geoff, and Serhat Guven. 2007. GLM basic modelling: Avoiding common pitfalls. *Casualty Actuarial Society Forum*, Winter 2007, pp. 257–72. Available online: https://www.casact.org/sites/default/files/database/forum_07wforum_07w263.pdf (accessed on 11 February 2023).
- Wüthrich, Mario V. 2018. Neural networks applied to chain-ladder reserving. *European Actuarial Journal* 8: 407–36. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.