Advancing Loss Reserving: A Hybrid Neural Network Approach for Individual Claim Development Prediction

Brandon Schwab[1, †] and Judith C. Schneider[1]

[1]Institute for Risk and Insurance, Leibniz University Hannover, Hannover, Germany
[†]Corresponding author. Address: Königsworther Platz 1, 30167 Hannover, Germany. E-mail: brandon.schwab@insurance.uni-hannover.de

This version: March 22, 2024

Abstract

The accurate estimation of loss reserves is critical for the financial health of insurance companies and informs numerous operational decisions, from pricing to strategic planning. We add to the literature by a proposing a novel neural network architecture that enhances the prediction of incurred loss amounts for reported but not settled (RBNS) claims. Moreover, in contrast to most other studies, we test our model on proprietary data sets from a large industrial insurer. Our analyses reveal the model's superiority in estimating reserves more accurately across different lines of business than standard benchmark models, like the chain ladder approach. Particularly, it exhibits nuanced performance at the branch level, reflecting its capacity to integrate individual claim characteristics effectively. Our findings underscore the potential of machine learning in enhancing actuarial forecasting and suggest a shift towards more granular data applications in the insurance industry.

JEL classification: C45, C58, G22, G58

Keywords: Loss reserving, RBNS reserves, Neural Networks, Multi-task learning, Deep Learning

1 Introduction

A fundamental aspect of an insurance companies operations is loss reserving. Loss reserving is mandated by regulatory frameworks such as Solvency II in Europe or ORSA in North America to maintain the financial health and solvency of insurance companies (EIOPA, 2019; NAIC, 2022). Traditional loss reserving methods like the chain ladder method are based on aggregating claims into a homogeneous portfolio structured in a triangular shape that captures the development of claims over time. Besides for its originally purpose risk management, information on loss reserves and thereby a good prediction of the ultimate claim amount, enters areas such as pricing, portfolio management and strategic business planing (Taylor, 2019). This need arises from the fact that accurate financial forecasting and decision-making depends on knowing the full extent of liabilities, not just current liabilities. By ensuring that the final claim amounts are correctly estimated, companies can more effectively set premiums, assess the health of the portfolio and develop strategies for future growth, and consequently improving overall financial stability and operational efficiency across the various functions. The traditional aggregated view treats all claims in a portfolio as homogeneous and unique characteristics of the claims are ignored. Thereby, relying on the assumption that past patterns will continue in the future, which can be problematic with changing external factors like inflation, which is usually not included in the predictions. This information, however, is of great importance to provide granular insights on the claim developments and allow for an up to date handling of processes within the insurance. The importance of loss reserving within the insurance coupled with new advances in data collection and computational efficiency call for more granular and flexible approaches for loss reserving based on individual claims. Such methods offer the flexibility and adaptability to effectively respond to evolving market conditions and emerging risk patterns.

Our paper adds to this discussion by developing a new machine learning based model architecture, which maintains a level of complexity comparable to recently proposed models and which we test on two real claim portfolios of a large industrial insurance. Although there are a couple of other papers using machine learning techniques, a notable challenge is the limited availability of public individual claims data. For this reason, empirical studies on loss reserving often refer to the stochastic simulation machine by Gabrielli and V. Wüthrich (2018). Therefore, our data sets offer a particularly interesting perspective on investigating advanced machine learning techniques for loss reserving.

In the literature, newer models which take into account richer data on individual claims are either parametric or use machine learning techniques. However, none of the suggested models became a gold standard and advances are still needed. The approach proposed by Kuo (2020), while not surpassing chain ladder estimates at an aggregate level for a specific simulated data set, offered insightful individual claims forecasts, though without benchmark comparisons at the granular level. Gabrielli (2020) reported in his study individual claims reserves within 2% of true payments across all lines of business considered, yet did not benchmark against other models. Chaoubi et al. (2022) found that, depending on the data set, their Long Short-Term Memory (LSTM) model either slightly overestimated reserves compared to the chain ladder method or, in real data scenarios, provided closer approximations to actual reserves, highlighting its potential for capturing claim trends more accurately than traditional methods. However, comparisons were limited to the chain ladder model. These studies underscore the ongoing need for advances in loss reserving methodologies, as no proposed model consistently outperforms the chain ladder approach when considering a broad spectrum of scenarios.

Central to the task of loss reserving is the prediction of two claim types: Claims that have been incurred but are not reported (IBNR) to the insurance and claims that have been reported but are not settled (RBNS). The paper at hand focuses on RBNS claims and suggests a machine learning algorithm tailored to the dynamic nature of the task at hand as well as the demand for more detailed analysis incorporating a wide range of granular claim characteristics. We consider incurred losses as payments plus individual case reserves, such that expert information on case reserves works as latent information which enters the machine learning algorithm to predict the cumulative incurred losses of the unknown periods. By leveraging standard deep learning techniques for static features and a LSTM model with an added attention mechanism for dynamic features, our model efficiently processes and combines these diverse inputs. The final predictions are made through a straightforward decision-making process based on the predicted probability of changes in cumulative losses, utilizing a composite loss function to optimize for both classification and regression tasks. We benchmark our model against traditional methods like the chain ladder approach[1], expert forecasts provided by an experienced loss reserving actuary, reflecting industry standards and a standard econometric model. In addition, we test against our machine learning algorithm but based on incremental claim amounts, which is commonly employed as outlined by Kuo (2020) and Gabrielli (2021) and the model proposed by Chaoubi et al. (2022). We evaluate the performance based on the percentage error of the estimated

[1]Despite its simplicity, the Chain Ladder method is still one of the most commonly used method for loss reserving (Wüthrich, 2016).

reserve and reveal the advantages of individual claim modeling by comparing results not only at the portfolio level but also within sub-portfolios.

Our comparative analysis demonstrates that the neural network model, especially when processing cumulative data, consistently outperforms traditional methods such as the chain ladder and linear regression models in estimating reserves across both property and liability lines. The granular analysis treats each claim individually, accounting for unique characteristics and specific risk factors. This offers a more detailed understanding of risk at the individual claim level, leading to more effective risk mitigation strategies. Moreover, the approach allows for detailed portfolio analysis and optimization, improving the understanding of portfolio performance and profitability within the insurance. The approach allows great versatility in application as a single model can predict individual losses and aggregate them to the portfolio or sub-portfolio levels, offering flexibility and reducing the need for multiple models. Lastly, the model can enhance pricing accuracy by facilitating more accurate and tailored pricing strategies by incorporating individual claim details, leading to better alignment with actual risk.

The paper is organized as follows: Section 2 presents the context and process of loss reserving and provides an overview of the existing literature on loss reserving models. Section 3 discusses the used data sets. Section 4 introduces the proposed neural network architecture to estimated the outstanding claim amounts. Section 5 presents the benchmark models used to compare the results. Section 6 discusses the results, and Section 7 concludes the paper.

2 Loss Reserving and Literature Overview

2.1 Loss Reserving: an economic perspective

Loss reserving is economically important and belongs to the core task within the insurance (Radtke et al., 2016). As depicted in Figure 1, the loss reserving process can be described as follows:[2] first, a claim occurs, then with a certain delay the insurance taker reports a claim to the insurance company. Based on the insurance knowledge at reporting, parts of the claim might be settled immediately or partially until the claim is closed. Before the final closure the claim could be reopened and further payments and recoveries can take place. In non-life insurance, actuaries go beyond analyzing raw claim payments by focusing on so-called incurred losses, which defines the sum of raw claim payments

[2] A similar description can be found in M. Pigeon et al. (2014) and Pigeon and Duval (2019).

and case reserves. The process of payments, case reserves, and recoveries upon closure/final closure is called incurred loss adjustments. Incurred losses change dynamically over time. Case reserves are set by experts and express their current estimate of the outstanding loss on individual claims. Payments and case reserves are not necessarily adjusted at the same point in time. Yet, it is important to note that incurred losses remain unaffected in scenarios where case estimates automatically adjust with claim payments, keeping the overall incurred loss constant. Further it is important to note that, particularly in the context of industrial insurance, incurred loss amounts may not change in every period, especially for complex claims characterized by long settlement horizons. Overall, the ultimate paid amount aligns with the ultimate incurred loss amount, ensuring a balance in the final assessment. Our later analyses will take into account these specifics of industrial insurances, and our machine learning methods is tailored to address these characteristics.

At the time of valuation, denoted by t^* insurance companies distinguish between Reported But Not Settled (RBNS) and Incurred But Not Reported (IBNR) claims. An IBNR claim has occurred before the valuation date, but is reported and settled afterwards. In the case of an RBNS claim, the occurrence and reporting of the claim occur before the valuation date and settlement takes place afterwards. As our model is specifically designed to leverage individual claim characteristics and given that the characteristics of IBNR claims are not observable at the time of valuation, we solely focus on predicting RBNS claims.

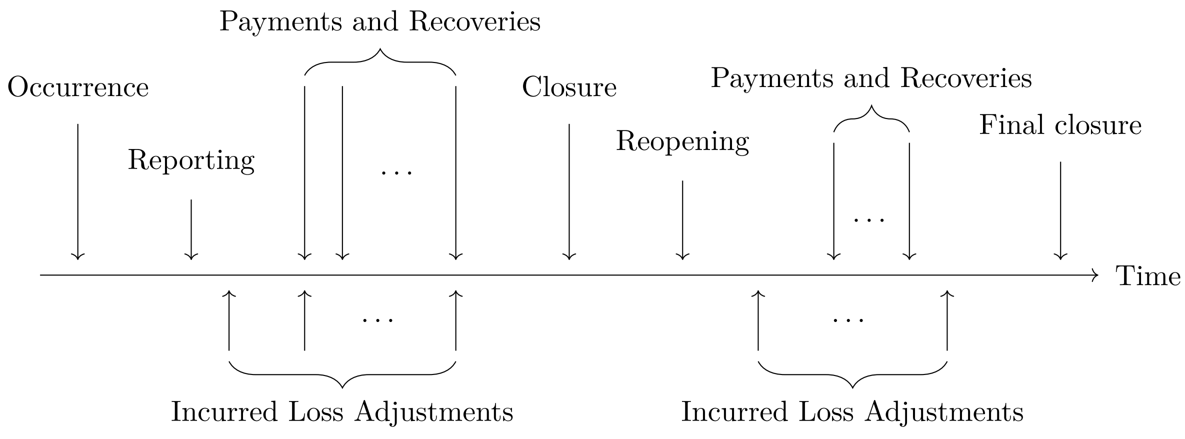

Figure 1: Claim settlement process

We consider a portfolio consisting of K reported claims, indexed by $k = 1, \dots, K$, each reported after a specified point in time. In our analysis, we consider discrete points in time, with development periods of equal length, each spanning one year. We also assume that each claim in the portfolio is fully settled within n development periods. Therefore, for each claim, we have development periods indexed by $j = 0, \dots, n$. Our objective is to predict the incurred losses for the unknown periods $t^* + 1, \dots, n$, for each claim k.

Given an evaluation date t^* the reserving task entails estimating the ultimate loss amount $S_n^{(k)}$, for each claim k. This ultimate loss amount represents the cumulative incurred amount at period n and is equivalent to the ultimate paid amount at settlement, as the case reserve at settlement is zero. This estimation, denoted as $\hat{S}_n^{(k)}$, is based on all information available at t^*. The estimated reserve for an individual claim at t^* is thus the difference between the estimated ultimate loss amount and the incurred amount up to t^* and can be expressed by

$$\hat{R}_{t*}^{(k)} = \hat{S}_n^{(k)} - S_{t*}^{(k)}. \tag{1}$$

Of course, predicting the cumulative incurred losses for the periods $t^* + 1, \dots, n$ is equivalent to predicting the incremental amounts for those periods. We compare predicting cumulative incurred losses to incremental forecasts, which is defined as

$$\hat{S}_n^{(k)} = S_{t^*}^{(k)} + \sum\nolimits_{j=t^*+1}^{n} \hat{Z}_j^{(k)}, \tag{2}$$

where $\hat{Z}_j^{(k)}$ is the predicted incremental incurred loss for claim k in development period j. Equation 1 can thus be expressed in terms of Equation 2 as the sum of the incremental incurred losses forecasted for the unknown periods:

$$\hat{R}_{t*}^{(k)} = \sum\nolimits_{j=t^*+1}^{n} \hat{Z}_j^{(k)}. \tag{3}$$

This equivalence between modeling cumulative and incremental losses underpins our approach, allowing for flexibility in predicting claim developments. By exploring both approaches, our model offers a robust and versatile tool for reserve estimation in the dynamic landscape of non-life insurance.

2.2 Loss Reserving: a methodological perspective

Traditional loss reserving methods like the chain ladder method are based on aggregating claims into a homogeneous portfolio structured in a triangular shape that captures development of claims over time. These methods have been extensively employed and studied. For example, Mack (1993) introduce a distribution-free stochastic framework of the chain ladder method to quantify uncertainty in the reserve estimates. Additionally, Verrall (2000) explore a variety of stochastic models that align with the chain ladder reserve estimates. Further important contributions by England and Verrall (2002, 2006) and Pinheiro et al. (2003) introduce bootstrap a Bayesian techniques, allowing for the integration of expert judgments combined with a general linear model framework. For a comprehensive overview of existing methods, we refer to Wüthrich (2008). New advances in data collection and complexity lead to extensions of the traditional approaches and shape the further development of the univariate chain ladder method to multivariate extensions (Merz & Wüthrich, 2008; Pröhl & Schmidt, 2005; Shi, 2017; Y. Zhang, 2010). The multivariate extensions are able to develop a portfolio of several correlated subportfolios by accounting for both contemporaneous correlations and structural relationships. However, the methods require strong structural assumptions on the relation between subportfolios and aggregate portfolio, which are often rather adhoc. That is one huge advantage of machine learning methods, where non-linear dependencies are captured without strong adhoc assumptions.

Early contributions towards granular modeling on loss reserving are based on parametric models (Buhlmann et al., 1980; Norberg, 1986). Arjas (1989) and Norberg (1993, 1999) model the development of individual claims to compute reserves based on **P**osition **D**ependent **M**arked **P**oisson **P**rocesses (PDMPP). Larsen (2007) refine their approach and decompose the complex stochastic process of claim developments into independent segments, allowing for a more nuanced and detailed analysis of the stochastic reserving process. Zhao and Zhou (2010) extend this idea by accounting for the dependence of claim event times and covariates. In addition, Huang et al. (2015, 2016) developed a stochastic model based on individual data that takes into account information such as the occurrence, reporting and settlement times of individual claims. The model showed a significant reduction in mean squared error loss compared to classical models based on aggregated data. The results presented in this study therefore support the use of models with individual data.

In particular, Wüthrich (2016) summarizes the benefits of machine learning in individual claims reserving in his review article. Pigeon and Duval (2019) explores these ideas by using different tree-based approaches like the XGBoost algorithm to predict the total paid amount of individual claims. While Baudry and Robert (2019) use another tree-based ensemble technique, namely ExtraTrees, to compute IBNR and RBNS claims reserves at the individual level.

While there is a significant evolution in models for loss reserving, especially with the integration of machine learning techniques, a notable challenge is the limited availability of public individual claims data. To address this issue, Gabrielli and V. Wüthrich (2018) creating a stochastic simulation machine based on neural networks to synthesize claims data and provide back-testing capabilities. Based on these stochastic simulation machine Gabrielli (2020) and Gabrielli et al. (2020) embed classical loss reserving models into neural networks by initially aligning the network with a traditional model such as the over-dispersed Poisson, then refining it through training to reduce prediction errors, effectively leveraging a boosting-like method. Gabrielli (2021) presents an individual claims reserving model for reported claims, which uses summarized past information to predict the expected future payments. Yet the approach develops a dual-output neural networks, which model both the probability of a payment and its corresponding amount. Instead Kuo (2020) again focuses on the aggregated losses but takes into account the time series nature of the underlying data. His framework facilitates the joint modeling of paid losses and claims outstanding, adapting to incorporate various data types. He tests his model on aggregated data sets and demonstrates potential for broader application with more detailed data. Further Kuo (2020) introduce an individual claims model for RBNS claims incorporating an encoder LSTM for past payments and a decoder LSTM for generating paid loss distribution, augmented by a Bayesian neural network for uncertainty quantification. However, none of the proposed models consistently outperformed the classical chain ladder method systematically. Chaoubi et al. (2022) demonstrate a different model architecture, using a LSTM network followed by two fully connected layers to jointly predict the probability of a payment or recovery and the corresponding amount. Their model focuses on the prediction of incremental payments and treats static features as dynamic within the network. While there seems to be an advantage of their model over the chain ladder method, an extensive set of benchmark models as well as large samples of actual data is missing.

3 Data

In this section, we first detail the underlying data sets used in our analysis and describe the different claim features and then outline how we use the data for training and testing purposes.

3.1 Data Description

Our data comprises proprietary claims data provided by a large industrial insurance company, covering both short-tail and long-tail lines of business (LoB), in particular property and liability lines.[3] The scarcity of publicly available, individual claims data in empirical studies of insurance loss reserving often leads to the use of simulated claims data sets, to offer the ability of back-testing the proposed models. Baudry and Robert (2019), Kuo (2020) and Gabrielli (2021), for example, used all synthetically generated data on individual claims for their analysis. Moreover, when real claim data is utilized, the focus tends to be narrowly on a single line of business, predominantly examining general liability insurance for private individuals.[4] The use of real-world, individual claim data from two distinct LoBs within the industrial insurance domain is therefore particularly interesting for analyzing the use of machine learning methods for insurance loss reserving.

The property claim data set contains a total of 66,208 claims arising from 16,713 distinct policies reported from January 1, 2011, to December 31, 2016, while the long tailed liability claims data set includes 403,461 claims from 24,084 distinct policies, reported from January 1, 2000, to December 31, 2011. Both data sets include observations up to the end of 2021, which ensures a comprehensive analysis framework with at least six and twelve discrete observation periods for property and liability lines, respectively. This observation timeline aligns with the presumption that claims are fully settled within these specified periods — six for property and twelve for liability — which corresponds with the insurance company's internal reserving practices. This depth of data enables us to fully analyze the development trajectory of each claim, a critical advantage for back-testing our model against real-world outcomes.

The data can be dissected into two types of data. The first type includes the following static features, which are fixed and do not vary over time:[5] The *Contract category*, which identifies whether an insurance policy is part of an international program or a local policy (P/L), the *Contract type*, which is differentiated into standard, primary, layer, and master policies (P), the *Business type*, segmented into sole, lead, coinsurance, and indirect business (P/L), the *Policy share*, which captures the share of the overall insured risk (ranging from 0% to 100%) (P/L), the *Risk category*, which classifies each

[3]Short tailed means that claims are generally reported and settled more quickly, while long tailed means that a significant proportion of total claims payments take a long time to be settled by the insurer.

[4]See for example M. Pigeon et al. (2014), Pigeon and Duval (2019) or Chaoubi et al. (2022).

[5]The designation (P) denotes attributes that are exclusively present in the property claims data set, while (P/L) indicates attributes that are present in both the property and liability data sets.

claim into one of 12 distinct risk categories for the Property LoB, whereas for the Liability LoB, the classification is into one of 7 categories[6], the *Risk class*, with values from 1 to 10, where 10 signifies a higher risk (P), the *Coverage*, which is detailed with 3 different coverages for the Property and 12 for Liability LoB, the *Covered perils*, which include 3 classes (P), the *Claim type*[7], differentiated into attritional losses, large losses and natural catastrophes (P/L), the *Notification duration* of the claim, measured in days (P/L) and the *Branch*, indicating which of the 12 branches the corresponding policy belongs to (P/L).

The second type of data is dynamic, i.e. it may vary over time: The *Development period* of the claim (P/L), the *German real GDP* (base year 2011) (P/L) and an internal *Inflation mixture indices*, created internally by the insurance company to reflect inflationary trends relevant for the specific LoB (P/L). The features used in our model for the two data sets are summarized in Appendix A.

We initially refined the data sets by removing claims categorized at the reporting date as large losses. At the point of notification, each claim is classified by the claims handler as either an attritional loss, a large loss or a natural catastrophe. The classification is based on whether the expected ultimate loss amount exceeds a predetermined threshold specific to the LoB. Large losses, especially in the industrial insurance, can introduce significant volatility to the data, which is likely to distort the predictions of the model aimed at more typical claim behavior and is therefore usually analyzed and modeled separately (Denuit & Trufin, 2018; Riegel, 2014). In practice, the reserving for large losses often involves a significant amount of expert judgment and adjustments. Due to this complexity and the unique handling required for these claims, we have excluded them from our analysis. Excluding large losses narrows our focus to attritional patterns, enhancing model precision for typical claims, yet it also highlights the necessity of separately addressing large losses for a holistic loss reserving approach. It is important to note that some claims may ultimately develop into large losses, even though they were initially categorized as attritional losses and are therefore still part of our data sets.

3.2 Training- and Testing-Setup

Grouping claims not only by their development period j but additionally by their reporting year, indexed by $i = 1, \dots, n$, facilitates the structural organization of the data in a manner that aligns with traditional loss reserving methods based on loss triangles. This structure enables us to define the

[6]The risk category classification for each claim is determined based on the industry sector in which the insured company operates.

[7]Only attritional and large losses are present in the Liability data set.

cumulative incurred loss amount for claim k, in reporting year i, and in development period j as $S_{i,j}^{(k)}$.

Figure 2 illustrates exemplary how the property data set is organized in the form of a common loss triangle, where each cell represents $S_{i,j}^{(k)}$ for a particular claim. This representation is helpful for visualizing the comparison between our individual loss reserving approach and the traditional aggregated methods. Moreover, we can visualize how we split our data into a training, validation and test set.

Given that our data sets contain the complete development trajectory of each claim up to period n, we first split the data into a training and testing set by setting the evaluation date t^* to December 31, 2016 and December 31, 2011 for the Property data set and the liability data set, respectively.[8] Up to these dates the cumulative incurred loss amounts $S_{i,j}^{(k)}$ for $i + j \leq n$ are observable for each claim k and serve as our training data (represented in grey in Figure 2). The incurred loss amounts beyond this valuation date, for $i + j \geq n + 1$, which are not observable as of t^*, form the test set (represented in orange in Figure 2).

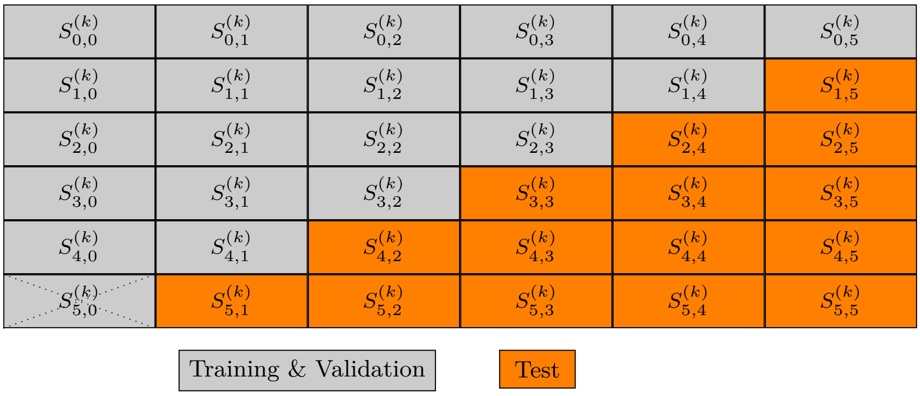

Figure 2: Construction of training- and testset

Further, we split the claims inside the training data randomly into the final training and validation sets.[9] Specifically, 80% of this data is used for model training, while the remaining 20% is used for model validation to evaluate the model's performance and generalizability.

[8] It is worth noting that with a greater observation timeline, a rolling window approach could be used to further test the stability of the model over time.

[9] Claims which are reported at the most recent reporting year are excluded from the training process, because here only one development period is known.

4 Methodology

To model the development of individual claims, we propose a specific neural network architecture tailored to the claim settlement process. Key to our neural network model are two prediction goals: estimating the cumulative incurred loss amounts of the unknown periods and estimating the probability of changes in the cumulative incurred loss amounts within those periods. Neural networks, with their capacity to discern complex patterns within large data sets, have become pivotal in insurance applications, from fraud detection (Gomes et al., 2021), pricing (Wüthrich, 2019), to enhancing customer service interactions (Ansari & Riasi, 2016). Our approach incorporates different deep learning building blocks, particularly suited for modeling the sequential nature of claim developments and leveraging the information stored in the static features.

This section outlines our model architecture and presents the benchmark methods and evaluation metrics to assess the predictive performance. The code for the application of the proposed model using a synthetic data set is available online.[10]

4.1 Model Architecture

The model architecture closest to ours is introduced by Chaoubi et al. (2022). While inspired by the architecture, we introduce significant changes: We treat static and dynamic features inside the network individually, we include an attention mechanism on top of our sequential model to further enhance the network's ability to focus on the relevant parts of the input sequence and we use a decision rule for updating the predicted claim amounts.

In essence, our model takes a two-part approach: it processes static and dynamic features separately before integrating them to produce the two task outputs. This allows the model to capture the unique characteristics of each type of feature. In our setting the final prediction is based on a simple decision rule incorporating the two outputs: If the probability of a change in the cumulative incurred loss amount exceeds a predefined threshold, the model updates its forecast by using the output of the regression task and otherwise the current estimate is retained. The specific threshold is optimized during the training phase and ensures that the forecast for the next period depends on a substantial probability of a change and mitigates the risk of over-adjustment in periods of claim stability, therefore implicitly optimizing operational efficiency.

[10] https://github.com/brandonschwab/advancing_loss_reserving.

The first step in our network consists of processing the static features, which contain both quantitative and categorical types. All static quantitative features are standardized, and each of the static categorical features is initially indexed and then processed by embedding layers, as introduced by Bengio et al. (2000). According to Kuo (2020) and Gabrielli (2021), we embed categorical variables into two-dimensional vectors. The processed static features are concatenated into a single vector, denoted by $\tilde{\mathbf{F}}_0^{(k)}$, and fed into a fully connected layer[11] (FC) with a Rectified Linear Unit (ReLU) activation function, which results in the output vector $\tilde{\mathbf{F}}_0'^{(k)}$.

Dynamic features, denoted by $\mathbf{D}_j^{(k)}$, are processed within our model by a Long Short-Term Memory (LSTM) layer, introduced by Hochreiter and Schmidhuber (1997). This architecture is designed to handle sequential data by maintaining a memory of past information, which is particularly useful for modeling the time-dependent behavior in the claims reserving process. The LSTM operates on a sequence of dynamic features, updating two crucial components at every time step: the cell state $\boldsymbol{C}_j$ and the hidden state $\boldsymbol{h}_j$. These states integrate new information provided by the current input and are essential for the model to remember and forget information through a process involving three gates: The forget gate controls the extent to which the previous cell state is retained. The input gate updates the cell state by adding new information and the output gate decides what information from the cell state will be used to generate the output hidden state, which is further used for the prediction of the next development period. These operations allow the LSTM to maintain and manipulate its internal state over time, providing the ability to remember information across sequences of variable lengths. We refer to Figure 3 for a visual representation of a single LSTM cell, which delineates the workflow through its various gates. As the LSTM processes the sequence of dynamic features up to the last observed period, it outputs a series of hidden states. The last hidden state is particularly important as it encapsulates the information from the entire input sequence and is used for making the one-step-ahead prediction for the coming unknown period.

To further refine the model's focus on relevant temporal patterns, we introduce an attention mechanism on top of the set of hidden states. Attention mechanisms, initially popularized in the context of neural machine translation (Bahdanau et al., 2014; Luong et al., 2015), allow a model to dynamically focus on different parts of the input sequence to generate each element of the output sequence. They work by assigning weights to different input elements, indicating their relative importance for the task

[11] For an in-depth understanding of deep learning principles, including dense layer functionalities, see Goodfellow et al. (2016) and Chollet and Allaire (2018).

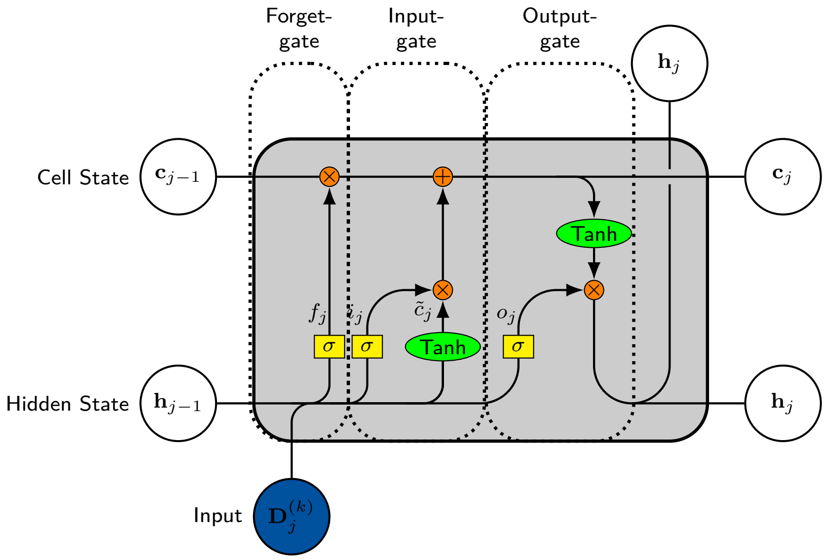

Figure 3: LSTM cell

at hand. This approach has been shown to significantly improve the performance of time series tasks, as demonstrated by Lai et al. (2018), who empirically showed an enhancement in performance across various time series domains. Further evidence of its effectiveness can be seen in financial time series predictions, where attention mechanisms have been successfully used in conjunction with LSTMs (Kim & Kang, 2019; X. Zhang et al., 2019), underscoring the adaptability and efficiency of this approach in diverse applications. Our dot product attention mechanism is identical to the one used by Lai et al. (2018) and is designed in a way that the hidden states can be only influenced by its predecessors without affecting them in return, maintaining the temporal ordering of the sequence.

Finally, the processed static feature vector is replicated and concatenated with the attended hidden states, to form a comprehensive feature set for each period. This resulting feature vector is then fed into another FC layer using the ReLU activation function, followed by two separate pathways to produce the one-step ahead prediction. One pathway involves a FC layer with the identity activation function to produce the prediction of the cumulative incurred loss amount of the next period $\hat{Y}_{j+1}^{(k)}$. The second pathway utilizes a FC layer with a sigmoid activation function to get the probability of a change in the cumulative incurred loss amount in the next period $\hat{p}_{j+1}^{(k)}$. The proposed architecture is illustrated in Figure 4.

To optimize the network's weights, we performed a two-staged approach. Initially, Random Search was utilized to broadly explore the hyperparameter space, followed by a refinement process using Bayesian Search. A detailed description of our training methodology and hyperparameter tuning

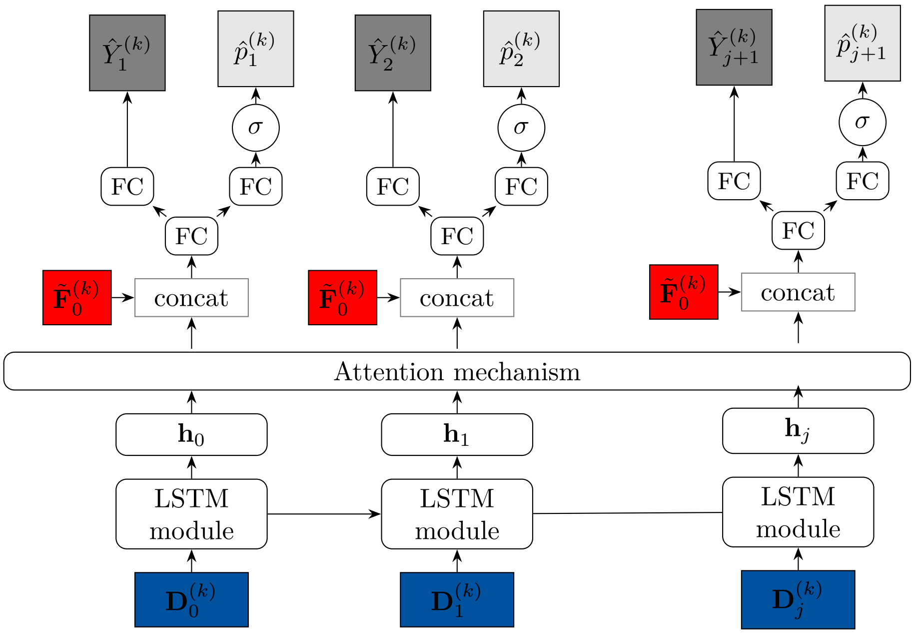

Figure 4: Network architecture

strategy, including the specifics of the two-staged approach, is provided in the Appendix B.

5 Benchmark Models

To assess the predictive performance of our proposed model, we compare the results, based on the test sets, to several benchmark models, including aggregated and individual models.

We apply the chain ladder method to our two aggregated data sets from the property and the liability lines of business. By employing the chain ladder method to our data sets, we aim to establish a foundational comparison for our model, particularly given its wide acceptance and usage in the industry for reserve estimation.

Additionally, we provided the aggregated data sets to an experienced reserve actuary for a runoff analysis. The expert, relying on classical methods such as the chain ladder, integrated their professional judgment into the forecasting process. This approach allows to benchmark our model against human expertise in the field. It is important to note that the expert did not have access to exposure information, such as premiums. Typically, exposure data offer additional insights and are considered valuable in actuarial analysis. However, our provided data sets only include this information partially, which could potentially impact the expert's ability to make more informed predictions.

Further, we utilize the simplest form of econometric modeling: a linear regression. This choice allows to highlight the contrast in predictive performance between a straightforward econometric approach and the complex neural network architecture on an individual claim level. Here, rather than using the complete feature set, which could introduce noise and overfitting, we employed a simple step-wise backward elimination based on the Akaike Information Criterion (AIC) for selection. Starting with the full model, we sequentially dropped predictors, while incorporating fixed effects for both the reporting years and development periods to account for their temporal dynamics (Friedman, 2009).

In addition, we include the model architecture proposed by Chaoubi et al. (2022) as a critical point for comparison. For this benchmark, we utilized incremental data, following the modeling strategy used by the authors, but adapt the training process to mirror that of our model. Chaoubi et al. (2022) suggest a method where the final prediction arises from the product of the predicted probability of a non-zero loss and the outcome of the regression task. This approach contrasts with our use of a decision rule, where we separately evaluate the two tasks before making a prediction. This inclusion is particularly insightful as it allows us to directly assess the impact of the modifications we introduced.

Lastly, we extended our model's application to incremental data. The training process remained largely analogous to that used for cumulative data, with a change in setting up the classification task and the final decision rule. In the context of incremental data, our classification objective transformed to predicting the probability that the incurred loss amount in the upcoming period is non-zero. The final prediction of the model is therefore given by the decision rule: If the probability of a non-zero incurred loss amount exceeds the threshold, we use the regression task output and otherwise we predict a zero amount.

In summary, these benchmark models – the chain ladder method, expert forecasts, linear regression, the model proposed by Chaoubi et al. (2022) and our model's adaptation to incremental data – provide a robust foundation for evaluating our novel approach. To compare the models, we assess the predictive performance using the percentage error of the estimated reserve for the complete portfolio and for sub-portfolios. We also evaluate the accuracy of the individual cumulative loss forecasts by comparing the mean absolute error (MAE) and the root mean squared error (RMSE) for the regression task and the balanced accuracy for the classification task of our model. The MAE is particularly informative, providing a straightforward metric directly comparable to the data's scale and inherently less sensitive to outliers. Conversely, the RMSE offers a complementary perspective, penalizing larger errors more severely and thus providing insights into the variability of the predictions. Together, these metrics furnish a holistic view of the model performance, encompassing both average accuracy and the distribution of the prediction errors. In addition to MAE and RMSE, the balanced accuracy is used to evaluate the classification task of the models and is especially informative in the context of insurance claims, where the occurrence of claim adjustments, i.e., changes in the incurred loss amount, may not be uniformly distributed across all claims. This metric adjusts for any imbalances by considering both the True Positive Rate and True Negative Rate, thus providing a more accurate measure of the model's ability to correctly identify periods with and without changes in claim amounts.

6 Results

In this section, we present the empirical results derived from the application of the proposed models to our data sets. We begin by examining the overall performance of each model, quantifying this via the percentage error of the estimated reserves, shown in Table 1.

Table 1: Percentage error of the estimated reserve

Model	**Line of Business**	
	Property	Liability
Chain Ladder	-14.36	-16.32
Expert	-13.66	-12.83
Neural Network	**-1.37**	**-1.18**
Neural Network (Incr)	7.59	-16.28
C-LSTM	11.34	-18.58
Linear Regression	-20.73	-61.37

The results reveal a huge heterogeneity of performance across the models. Our neural network models demonstrate superior accuracy, with the one based on cumulative data yielding a minimal percentage error of -1.37% for property and -1.18% for liability. This outperforms both the chain ladder and expert models, which show larger negative errors, indicating an underestimation of reserves. Interestingly, our neural network model applied to incremental data exhibits a notable deviation from the network based on cumulative data with a positive percentage error of 7.59% for property, suggesting an overestimation, while showing a comparable underestimation to the chain ladder and expert forecasts for liability with a -16.28% error. The C-LSTM model, reflecting the methodology proposed by Chaoubi et al. (2022), exhibits a similar pattern to ours, but larger in terms of the percentage error. With a percentage error of 11.34% for property, it slightly outperforms the chain ladder and expert model, but overestimates the underlying claims portfolio. Conversely, for the liability line, the C-LSTM model shows a percentage error of -18.58%, showing a greater underestimation compared to the traditional approaches. This result underpins the benefits of our modifications in the model architecture and the reserve estimation strategy.

Lastly, the performance of the linear regression model underscores its limitations in handling the complexity of claim data. With substantial underestimations of -20.73% for property and -61.37% for liability, the results indicate that the model's simplicity is inadequate for capturing the complex, possible non-linear patterns present in the data.

To further provide insights into the predictive performance of our models at the individual claim level, the mean absolute error (MAE), the root mean squared error (RMSE) and the balanced accuracy are calculated for each development period. The MAE and RMSE results for the property line are given in Tables 2 and 3. These tables not only present the errors for the final neural network predictions, but also isolate the MAE and RMSE for the regression task alone (NN_{Reg}, NN-$incr_{Reg}$ and C-$LSTM_{Reg}$), enabling us to distinguish the impact of the classification task on prediction accuracy. Importantly, it should be noted that in the case of the C-LSTM model, the final prediction is the product of the predicted claim amount and the predicted probability of a non-zero amount. This approach inherently differs from our classification task, as it directly integrates the probability into the regression output to refine the prediction. This distinction is crucial as it allows to infer the significant role of the classification task in enhancing the model's performance, especially when it successfully predicts periods without a change in the incurred claim amount, effectively reducing the error for those instances, or even becoming zero if the last state is known.[12]

Table 2: Mean absolute errors by development period - Property

Development Period	**MAE**						
	NN	NN_{Reg}	NN-incr	NN-$incr_{Reg}$	C-LSTM	C-$LSTM_{Reg}$	LR
0	-	-	-	-	-	-	-
1	**6471.18**	7613.51	6705.73	8208.20	10120.01	12133.79	22435.62
2	**4216.01**	5808.57	4420.60	6551.13	9649.82	11726.76	21202.63
3	**3164.51**	5506.75	3506.85	5970.45	7468.88	9006.52	19844.57
4	**2628.17**	5919.26	3017.99	5664.54	5744.01	7520.66	22632.86
5	**2295.07**	6078.58	2696.94	5393.38	5483.91	5573.62	21526.49

For the property line, the MAE and RMSE across all models, except for the linear regression, tends to decrease over the development periods, affirming the model's increasing accuracy as it gains more information about the claims' trajectory over time. Notably, the final predictions of the neural

[12]Note that in the modeling strategy proposed by Chaoubi et al. (2022), the predicted probability or the predicted loss amount must be exactly zero for the prediction to be zero.

Table 3: Root mean squared error by development period - Property

Development Period	RMSE						
	NN	NN_{Reg}	NN-incr	NN-$incr_{Reg}$	C-LSTM	C-$LSTM_{Reg}$	LR
0	-	-	-	-	-	-	-
1	**66125.00**	86505.19	93149.23	93588.95	97038.45	99947.38	107989.5
2	**48794.30**	55422.16	60020.38	61275.58	83845.39	86284.51	108565.4
3	**41513.47**	45098.61	49241.62	51511.82	63573.74	65392.23	108976.3
4	**38861.15**	41219.73	42994.12	46896.47	55384.53	56392.23	118673.2
5	**36505.84**	42208.82	40235.19	45568.77	44925.44	46392.63	117269.0

network models (NN, NN-incr and C-LSTM) consistently outperform the regression-only predictions (NN_{Reg}, NN-$incr_{Reg}$ and C-$LSTM_{Reg}$), strengthening the significance of the classification task in the predictive process. This is because the classification task enables the model to identify periods where the incurred claim amounts remain unchanged, thus potentially reducing the MAE/RMSE in those instances. Moreover, it is noteworthy that for the C-LSTM model, the differences between the final predictions and the regression outputs are smaller than observed in our proposed models. This could be attributed to the C-LSTM's approach, where the classification task is primarily used to scale the regression results and subtly improve the prediction without drastically changing the regression outputs.

The balanced accuracy results, calculated for our neural networks (NN and NN-incr), given in Table 4, follow a similar pattern, with a general increase over time, indicating an improving capability of the model to correctly classify the change in claim amounts as claims progress. Moreover, it is important to note that not all claims follow a uniform settlement pattern. While our models assume a fixed number of periods for full settlement, some claims may settle earlier. As the model adapts to these settlement patterns over time, the task of classification potentially becomes simpler with each passing period. This increased ease in classification over time, especially for claims settling earlier than the assumed period, likely contributes to the observed decrease in MAE/RMSE, reflecting the model's growing adeptness in predicting claim closure timings.

When examining the liability line, the models reveal a consistent decrease in the MAE over development periods, see Table 5. This can be confirmed by increasing balanced accuracy scores, shown in Table 7. This trend aligns with the results found in the property data and indicates a steady enhancement in the model's capacity to predict the central tendency of claim amounts accurately. The RMSE, given in Table 6, also diminishes over time, albeit with less uniformity, hinting at the influence of outliers in the reserving process. In industrial insurance the occurrence of outliers is generally more

Table 4: Balanced accuracy by development period - Property

Development Period	Balanced accuracy	
	NN	NN-incr
0	-	-
1	66.63	**67.60**
2	**69.17**	67.55
3	**71.53**	70.98
4	**73.86**	72.34
5	**79.44**	77.45

prevalent in the liability than in the property line. Liability claims often involve complex legal issues and long-tail liabilities, leading to a higher propensity for large, unpredictable claims that result in outliers. It is important to note that, despite initially excluding claims marked as large losses, some claims evolve into large losses during their lifecycle. This aspect can partially explain the observed variability in RMSE values, indicating areas where our model's performance could be improved in predicting these evolving large loss claims within the liability line.

Table 5: Mean absolute errors by accident years - Liability

Development Period	MAE						
	NN	NN_{Reg}	NN-incr	NN-$incr_{Reg}$	C-LSTM	C-$LSTM_{Reg}$	LR
0	-	-	-	-	-	-	-
1	**3589.24**	4002.85	4166.45	4872.63	6827.35	7462.35	9695.94
2	3700.28	**3696.14**	4763.73	4754.11	6429.82	7134.98	9915.20
3	**3491.77**	3512.71	4459.53	4728.03	6349.02	6692.48	9469.88
4	**3244.88**	3333.25	4150.49	4244.23	6388.21	6348.36	8965.10
5	**2956.08**	3077.01	3850.99	4320.46	5937.11	5798.37	8564.41
6	**2635.39**	2968.76	3618.07	3699.32	5242.14	5320.29	8020.77
7	**2406.20**	3182.96	3262.69	3510.52	4846.38	4937.72	7821.07
8	**2151.72**	3371.99	2965.87	3422.93	4511.44	4593.76	7489.86
9	**1926.77**	3438.47	2716.22	3156.28	4189.13	4248.58	7255.78
10	**1709.56**	3310.13	2474.39	2784.45	3843.22	3982.56	6996.52
11	**1541.10**	3160.89	2362.00	2914.93	3685.23	3938.23	6734.15

Next we shift our focus to a distinct advantage of our neural network model: its ability to perform runoff analysis at various granular levels, such as branch level — a particularly relevant aspect for practical actuarial applications. Table 8 shows the percentage error in estimated reserves for the property line across different branches. The neural network models (NN, NN-incr and C-LSTM) generally exhibit improved accuracy over the chain ladder method, with notably precise estimates in certain branches. Interestingly, for branch 300, the neural network and the C-LSTM based on incremental data (NN-incr and C-LSTM) show a lower percentage error compared to the cumulative model. This

Table 6: Root mean squared errors by accident years - Liability

Development Period	RMSE						
	NN	NN_{Reg}	NN-incr	NN-$incr_{Reg}$	C-LSTM	C-$LSTM_{Reg}$	LR
0	-	-	-	-	-	-	-
1	**43911.27**	44993.64	44707.68	45183.12	49846.34	49963.02	49746.39
2	**35967.97**	36975.27	36682.61	37393.16	39865.45	40345.74	41394.34
3	**28236.36**	29338.32	29118.09	29583.55	35208.20	36937.37	42923.49
4	**30231.94**	31257.50	31911.10	32000.49	34849.50	34823.45	43503.59
5	**39451.62**	39453.37	40853.51	41211.23	43349.97	44482.45	46037.23
6	**39380.21**	40381.08	41313.00	42399.94	43336.25	44794.55	48362.56
7	**39010.46**	39015.04	41062.08	41267.37	42483.69	42864.09	46826.47
8	**38185.56**	38206.45	39900.92	39985.65	41194.46	41503.33	44912.83
9	**36377.83**	37414.53	37970.84	38219.04	40370.40	41632.45	43562.29
10	**34043.78**	35092.36	35535.22	35537.94	37473.23	38234.88	44923.39
11	**32227.50**	32284.65	33594.83	34661.49	36104.57	36199.74	41824.45

Table 7: Balanced accuracy by accident years - Liability

Development Period	Balanced accuracy	
	NN	NN-incr
0	-	-
1	**66.61**	64.13
2	**66.63**	63.93
3	**66.94**	62.24
4	**67.45**	65.34
5	**67.12**	65.85
6	**67.89**	67.34
7	**67.57**	66.75
8	**70.20**	68.21
9	**73.75**	70.88
10	**75.18**	72.43
11	**76.12**	73.81

improved accuracy for branch 300 in the incremental data model may be due to its tendency to over-predict, which, in the context of large claims that disproportionately affect reserve estimates, could result in closer alignment with actual incurred losses. The cumulative model's underprediction of -10% suggests that it may not be capturing the volatility associated with such large claims effectively. Overall, the performance across branches is heterogeneous, but despite this variability, the neural network model based on cumulative data generally delivers better performance when compared to its incremental counterpart.

Table 9 presents the percentage error of the estimated reserve by branch for the liability data set, providing insights into the performance of various models. The results for the liability lines show a similar pattern to the property lines as the neural network models demonstrate a mixed but

Table 8: Percentage error of the estimated reserve by branch - Property

Branch	**% Error**				
	CL	NN	NN-Incr	C-LSTM	LR
100	-16.09	**11.70**	16.18	17.83	-18.34
130	-15.55	**-2.64**	3.07	7.24	-45.75
200	-13.40	**9.65**	29.21	34.48	-37.48
300	-18.20	-10.00	**1.91**	3.24	-43.11
400	-31.14	**0.88**	19.95	20.01	-65.97
430	-3.51	**-2.43**	3.43	7.11	-21.55
460	-14.58	**-2.49**	7.22	8.13	-58.61
650	-20.08	**1.80**	9.74	9.18	-41.03
700	-33.34	**12.65**	33.78	40.05	-77.50
800	-56.46	**-3.70**	10.06	15.33	-68.05
850	55.28	**38.25**	98.61	97.12	-98.79

generally more accurate performance than the chain ladder and linear regression methods across several branches. Here, the model based on incremental data (NN-Incr) exhibits a lower absolute percentage error in branches 430 and 650, indicating a more precise estimation in these instances. However, the difference in performance compared to the cumulative model (NN) is not markedly large, suggesting that both models have their respective strengths in handling the data for these specific branches.

Table 9: Percentage error of the estimated reserve by branch - Liability

Branch	**% Error**				
	CL	NN	NN-Incr	C-LSTM	LR
100	-56.90	**-20.65**	-43.05	-51.48	-87.47
130	-10.37	**0.42**	-33.51	-34.12	-89.12
200	-54.28	**6.45**	13.13	-16.79	-69.04
300	-23.11	**12.49**	-21.14	-22.55	-64.98
400	-38.46	**-3.79**	-19.77	-325.74	-59.26
430	-16.89	16.16	**-16.00**	-17.12	-71.83
460	-22.13	**-15.90**	-25.37	.26.42	-36.77
650	-38.78	-3.99	**1.70**	4.91	-55.39
700	-32.48	**-15.34**	-29.56	-33.23	-51.03
800	-18.62	**17.71**	-20.54	-22.38	-64.78
850	**15.23**	-16.61	-28.95	-30.01	-57.39

The neural network model, particularly when employing cumulative data, has consistently outperformed traditional actuarial methods such as the chain ladder and simple econometric models like linear regression across both property and liability lines. While the performance across branches exhibits heterogeneity, the trend suggests that the neural network's sophisticated architecture is adept at capturing the complexities of claims data more accurately than the benchmark models. The incremental neural network model also shows promise in certain segments, indicating its potential applicability under specific conditions. The deviation observed in the model applied to cumulative data could be linked to the inherent variability of incremental claims data. Incremental data, capturing changes in claims from period to period, may exhibit a more erratic trajectory, compared to a smoother progression of cumulative data. This complexity can lead to increased prediction errors. These results highlight the neural network's versatility and superior predictive capabilities, affirming its value as a powerful tool for actuarial forecasting and reserve estimation.

7 Conclusion

Nowadays, loss reserving is not only important for regulatory compliance and financial solvency. Besides for its originally purpose risk management, information on loss reserves and thereby a good prediction of ultimate claim amounts, enters areas such as pricing, portfolio management and strategic business planning. This expansion of the scope of application requires the need for granular and flexible approaches for loss reserving based on individual claims data. While the extant literature often assesses synthetic data to facilitate the back-testing of such models, the application to real claim data is often limited to general liability claims.

This study proposes a novel model architecture aimed at predicting the development of incurred loss amounts for RBNS claims. Key to our model is the prediction of the cumulative incurred loss amount for the upcoming periods and the probability of a change in this amount. By using a data driven decision rule – updating the forecast based on a predefined probability threshold – our model offers a refined approach to loss reserving. We further conducted a comparative analysis with established models, including the chain ladder method, expert forecasts, the proposed model by Chaoubi et al. (2022) and a simple linear regression model, as well as the proposed model trained on incremental data. To test the models, we utilized two proprietary data sets stemming from a short-tail and long-tail line of business from a large industrial insurance, a sector which, to our knowledge, has not previously been explored using individual loss reserving.

Our results show a superior performance based on the aggregated data, measured by the percentage error of the estimated reserve. Interestingly, the neural network model applied to incremental data exhibits a notable deviation from the network based on cumulative data, underperforming in comparison. Looking at the individual level, we could demonstrate the importance of the integration of the classification task which leads to a progressive decrease in the mean absolute error across successive development periods. The study further illustrates the model's adeptness at conducting runoff analysis across diverse granularity levels, particularly at the branch level. Such versatility in analysis underscores the model's adaptability to varied informational needs and organizational structures within the insurance sector.

While our results focus in the predictive modeling of RBNS claims, a comprehensive approach to claims reserves would also address IBNR reserves and the treatment of large losses, suggesting directions for future research. To address claims that may evolve into large losses, we suggest investigating the incorporation of predictive indicators and the potential use of ensemble techniques to improve prediction accuracy. Although this approach adds complexity, it may offer a nuanced understanding and management of large loss claims. Furthermore, the integration or development of an additional model tailored to handle the unique challenges posed by IBNR claims, which lack the detailed claim features available for RBNS claims, remains a crucial area for future exploration. In addition, the use of explainable artificial intelligence techniques to explain the model's decision-making process could further increase its usefulness and acceptance in practice and promote transparency and trust in automated claims reserving processes.

References

Ansari, A., & Riasi, A. (2016). Modelling and evaluating customer loyalty using neural networks: Evidence from startup insurance companies. *Future Business Journal*, *2*(1), 15–30.

Arjas, E. (1989). The claims reserving problem in non-life insurance: Some structural ideas. *ASTIN Bulletin: The Journal of the IAA*, *19*(2), 139–152.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade*, *7700*, 437–478.

Bengio, Y., Ducharme, R., & Vincent, P. (2000). A neural probabilistic language model. *Advances in Neural Information Processing Systems*, *13*.

Buhlmann, H., Schnieper, R., & Straub, E. (1980). Claims reserves in casualty insurance based on a probabilistic model. *Mitteilungen der Vereinidung Schweizerischer Versicherungsmathematiker*, *1*, 21–46.

Chaoubi, I., Besse, C., Cossette, H., & Côté, M.-P. (2022). Micro-level reserving for general insurance claims using a long short-term memory network. *Applied Stochastic Models in Business and Industry*, (39(3)), 382–407.

Chollet, F., & Allaire, J. J. (2018). Deep learning with R. Shelter Island. *Manning Publications Co. Biometrics*, *76*, 361–362.

Crawshaw, M. (2020). Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv: 2009.09796*.

Denuit, M., & Trufin, J. (2018). Collective loss reserving with two types of claims in motor third party liability insurance. *Journal of Computational and Applied Mathematics*, *335*, 168–184.

EIOPA. (2019). *Annual Report 2018*. European Insurance and Occupational Pensions Authority.

England, P. D., & Verrall, R. J. (2002). Stochastic claims reserving in general insurance. *British Actuarial Journal*, *8*(3), 443–518.

England, P. D., & Verrall, R. J. (2006). Predictive distributions of outstanding liabilities in general insurance. *Annals of Actuarial Science*, *1*(2), 221–270.

Fan, J., Upadhye, S., & Worster, A. (2006). Understanding receiver operating characteristic (ROC) curves. *CJEM*, *8*(1), 19–20.

Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Second edition). Springer-Verlag New York.

Gabrielli, A. (2020). A neural network boosted double overdispersed poisson claims reserving model. *ASTIN Bulletin: The Journal of the IAA*, *50*(1), 25–60.

Gabrielli, A. (2021). An individual claims reserving model for reported claims. *European Actuarial Journal*, *11*(2), 541–577.

Gabrielli, A., Richman, R., & Wuthrich, M. V. (2020). Neural network embedding of the over-dispersed poisson reserving model. *Scandinavian Actuarial Journal*, (2020(1), 1-29).

Gabrielli, A., & V. Wüthrich, M. (2018). An individual claims history simulation machine. *Risks*, *6*(2), 29.

Gomes, C., Jin, Z., & Yang, H. (2021). Insurance fraud detection with unsupervised deep learning. *Journal of Risk and Insurance*, *88*(3), 591–624.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. The MIT Press.

Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Huang, J., Qiu, C., & Wu, X. (2015). Stochastic loss reserving in discrete time: Individual vs. aggregate data models. *Communications in Statistics - Theory and Methods*, *44*(10), 2180–2206.

Huang, J., Wu, X., & Zhou, X. (2016). Asymptotic behaviors of stochastic reserving: Aggregate versus individual models. *European Journal of Operational Research*, *249*(2), 657–666.

James Bergstra & Yoshua Bengio. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(10), 281–305.

Kim, S., & Kang, M. (2019). Financial series prediction using Attention LSTM. *arXiv preprint arXiv: 1902.10877*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv: 1412.6980*.

Kuo, K. (2020). Individual claims forecasting with bayesian mixture density networks. *arXiv preprint arXiv: 2003.02453*.

Lai, G., Chang, W.-C., Yang, Y., & Liu, H. (2018). Modeling long-and short-term temporal patterns with deep neural networks. *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104.

Larsen, C. R. (2007). An individual claims reserving model. *ASTIN Bulletin*, *37*(1), 113–132.

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv: 1508.04025*.

Mack, T. (1993). Distribution-free calculation of the standard error of chain ladder reserve estimates. *ASTIN Bulletin: The Journal of the IAA*, *23*(2), 213–225.

Merz, M., & Wüthrich, M. (2008). Prediction error of the multivariate chain ladder reserving method. *North American Actuarial Journal*, *12*(2), 175–197.

NAIC. (2022). *NAIC Own Risk and Solvency Assessment (ORSA) Guidance Manual*. National Association of Insurance Commissioners.

Norberg, R. (1986). A Contribution to Modelling of IBNR Claims. *Scandinavian Actuarial Journal*, (3-4), 155–203.

Norberg, R. (1993). Prediction of outstanding liabilities in non-life insurance. *ASTIN Bulletin: The Journal of the IAA*, *23*(1), 95–115.

Norberg, R. (1999). Prediction of Outstanding Liabilities II. Model Variations and Extensions. *ASTIN Bulletin: The Journal of the IAA*, *29*(1), 5–25.

Pigeon & Duval. (2019). Individual loss reserving using a gradient boosting-based approach. *Risks*, *7*(3), 79.

Pigeon, M., Antonio, K., & Denuit, M. (2014). Individual loss reserving using paid–incurred data. *Insurance: Mathematics and Economics*, *58*, 121–131.

Pinheiro, P. J. R., Silva, João Manuel Andrade e, & Centeno, M. d. L. (2003). Bootstrap methodology in claim reserving. *Journal of Risk and Insurance*, *70*(4), 701–714.

Pröhl, C., & Schmidt, K. D. (2005). Multivariate chain-ladder. *Professoren des Inst. für Math. Stochastik*.

Radtke, M., Schmidt, K. D., Schnaus, A., & Schmidt, K.-D. (Eds.). (2016). *Handbook on loss reserving* (1st ed. 2016). Springer International Publishing.

Riegel, U. (2014). A bifurcation approach for attritional and large losses in chain ladder calculations. *ASTIN Bulletin: The Journal of the IAA*, *44*(1), 127–172.

Shi, P. (2017). A multivariate analysis of intercompany loss triangles. *Journal of Risk and Insurance*, *84*(2), 717–737.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, *25*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Taylor, G. (2019). Loss reserving models: Granular and machine learning forms. *Risks*, *7*(3), 82.

Verrall, R. J. (2000). An investigation into stochastic claims reserving models and the chain-ladder technique. *Insurance: Mathematics and Economics*, *26*(1), 91–99.

Wüthrich, M. V. (2008). *Stochastic claims reserving methods in insurance*. John Wiley & Sons.

Wüthrich, M. V. (2016). Machine learning in individual claims reserving. *Swiss Finance Institute Research Paper*, (16-67).

Wüthrich, M. V. (2019). Bias regularization in neural network models for general insurance pricing. *European Actuarial Journal*, *10(1)*, 179–202.

You, K., Long, M., Wang, J., & Jordan, M. I. (2019). How does learning rate decay help modern neural networks? *arXiv preprint arXiv: 1908.01878*.

Zhang, X., Liang, X., Zhiyuli, A., Zhang, S., Xu, R., & Wu, B. (2019). AT-LSTM: An Attention-based LSTM Model for Financial Time Series Prediction. *IOP Conference Series Materials Science and Engineering*, *569*(5), 052037.

Zhang, Y. (2010). A general multivariate chain ladder model. *Insurance: Mathematics and Economics*, *46*(3), 588–599.

Zhao, X., & Zhou, X. (2010). Applying copula models to individual claim loss reserving methods. *Insurance: Mathematics and Economics*, *46*(2), 290–299.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, & Andrew Rabinovich. (2018). Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *International Conference on Machine Learning*, 794–803.

A Data description

Tables 1 and 2 describe the set of features for the two datasets used in our model.

Table 1: List of covariates used for the property line of business dataset

Feature	Datatype	Static/Dynamic (s/d)	# of categories
Contract category	categorical	s	2
Contract type	categorical	s	4
Business type	categorical	s	4
Policy share (%)	numerical	s	-
Risk category	categorical	s	12
Risk class	categorical	s	10
Coverage	categorical	s	3
Covered peril	categorical	s	4
Claim type	categorical	s	2
Notification duration (days)	numerical	s	-
Real GDP[1]	numerical	d	-
Internal inflation mixture indices[1]	numerical	d	-
Branch	categorical	s	11

[1]Forecasted values were used for unknown periods to prevent data leakage.

Table 2: List of covariates used for the liability line of business dataset

Feature	Datatype	Static/Dynamic (s/d)	# of categories
Contract category	categorical	s	2
Business type	categorical	s	4
Policy share (%)	numerical	s	-
Risk category	categorical	s	12
Coverage	categorical	s	13
Claim type	categorical	s	2
Notification duration (days)	numerical	s	-
Real GDP[1]	numerical	d	-
Internal inflation mixture indices[1]	numerical	d	-
Branch	categorical	s	11

[1]Forecasted values were used for unknown periods to prevent data leakage.

B Hyperparameters

The objective of the learning process is to optimize the network weights to minimize a loss function tailored to the specific tasks - regression and classification. In our setting, the model weights are updated after each mini batch using the Adaptive Moment Estimation (Adam) optimization algorithm, introduced by Kingma and Ba (2014). We use its default parameters, recommended by the author's, and only tune the learning rate. Additionally, we adapt the learning rate throughout the training process by implementing a decay strategy, which helps the network converge to a local minimum (You et al., 2019). Specifically, we use the default settings and apply a learning rate decay every 10 epochs, using a multiplicative factor set to 0.1.

A mini batch refers to a subset of the training data consisting of a predefined number of claim sequences, each of varying lengths. We denote the lengths of the sequence for claim k as L^k. Thus, L^k periods for claim k are known. The claim sequences within a batch are processed and evaluated together in one iteration of the training algorithm. The learning rate and the batch size are therefore hyperparameters and are fine-tuned.

For each claim sequence within the batch, we adopt a one step ahead prediction strategy. This means, for a claim sequence of length L^k, we only feed the first $L^k - 1$ periods into the model to ensure we always have a target value available for prediction. Therefore, for each period j of the claim sequence, the model receives the static feature vector $\mathbf{F}_0^{(k)}$ and the set of dynamic features $\{\mathbf{D}_1^{(k)}, \mathbf{D}_2^{(k)}, \ldots, \mathbf{D}_j^{(k)}\}$, to predict the values for period $j + 1$.

The predictions are evaluated using two distinct loss functions, suitable for the two learning tasks. For the regression task, we use the Mean Squared Error (MSE) loss function. In parallel, for the binary classification task, we use the Binary Cross-Entropy (BCE) loss function.

The dual task learning approach offers multiple benefits. It reduces overfitting by utilizing shared representations, facilitates faster learning by leveraging auxiliary information and improves data efficiency (Crawshaw, 2020). However, this approach requires the definition of a single, unified loss function that effectively combines the individual losses. The critical challenge lies in appropriately balancing these two losses to ensure neither task dominates the training process, thereby maintaining the model's ability to learn both tasks effectively.

To address this challenge, we implement the Gradient Normalization (GradNorm) algorithm, introduced by Zhao Chen et al. (2018). GradNorm dynamically balances multi-task learning by adjusting each task's loss weight based on its learning rate.[13] Furthermore, we evaluate the loss function on the validation dataset to measure the network's ability to adapt to new data and prevent overfitting. The total number of epochs for our training process is set to 100 and we integrated an early stopping mechanism that stops the training if the validation loss does not improve for 10 consecutive epochs. For tuning we focus on the following hyperparameter: the learning rate (lr), the hidden size of the static feature vector $\tilde{\mathbf{F}}_0^{\prime(k)}$ (q_{static}), the size of the LSTM hidden states $\mathbf{h}_j$ (q_{lstm}), the hidden size of the combined feature vector $\tilde{\mathbf{X}}_j^{(k)}$ (q_{comb}), the batch size (b), and the dropout rate (d).[14] We start our search process by using Random Search to explore the hyperparameter space rapid and broadly (James Bergstra & Yoshua Bengio, 2012). The search space for our model's hyperparameters is guided by general recommendations from Bengio (2012) and Greff et al. (2017) and is configured for both data sets as follows:

- The learning rate is sampled from a log-uniform distribution, defined over the interval $[1\text{e}{-5}, 0.1]$. Here, the log-uniform distribution is used to explore candidate values that vary over several orders of magnitude.
- The hidden sizes q_{static}, q_{lstm} and q_{comb} are sampled from the discrete set $\{32, 64, 128, 256\}$.
- The batch size is sampled from the discrete set $\{1024, 2048\}$.
- The dropout rate is uniformly sampled from the interval $[0.1, 0.5]$.

After running the Random Search process with a total of 32 configurations, we refined our hyperparameter tuning using Bayesian Opimization (Snoek et al., 2012). Based on the outcomes of the Random Search, we identified the top three configurations and used their parameter ranges to define the search space for Bayesian Search. In constructing the search space for Bayesian Optimization, we took into consideration the parameter ranges observed in the top configurations from the Random Search. The upper and lower bounds for each parameter in the Bayesian search space were set based on the extremities of these ranges, ensuring a focused yet comprehensive exploration in the subsequent optimization phase. To thoroughly explore the refined search space, we evaluated 32 further configurations. The results of this two-stage hyperparameter tuning approach, including the top three

[13] For a detailed explanation of GradNorm, we refer to Zhao Chen et al., 2018.

[14] Dropout randomly sets a predefined percentage of neurons to zero during training, which helps prevent overfitting by ensuring that the network does not become overly reliant on any specific neuron (Srivastava et al., 2014).

configurations from the Random Search, and the final optimal configurations identified, are detailed in tables 1, 2 and 3 for both data sets analyzed in our study.

Table 1: Top three hyperparameter settings resulting from Random Search based on the property dataset.

	lr	q_{static}	q_{lstm}	q_{comb}	$batch$	$dropout$
Cumulative						
1.	0.004	64	128	128	1024	0.429
2.	0.001	32	64	128	1024	0.315
3.	0.003	128	64	64	1024	0.442
Incremental						
1.	0.00025	256	32	256	1024	0.1245
2.	0.00022	256	128	128	1024	0.3971
3.	0.00215	128	256	256	1024	0.2467

Table 2: Top three hyperparameter settings resulting from Random Search based on the liability dataset.

	lr	q_{static}	q_{lstm}	q_{comb}	$batch$	$dropout$
Cumulative						
1.	0.0935	128	128	128	1024	0.4854
2.	0.0439	64	128	128	1024	0.4081
3.	0.0043	128	256	64	1024	0.0.3854
Incremental						
1.	0.0296	64	32	32	1024	0.2516
2.	0.00088	128	64	64	1024	0.2487
3.	0.0273	128	64	32	1024	0.2340

To determine the threshold for our model's final prediction, we utilize the predicted probabilities obtained from the validation set. These probabilities were generated by the model, trained one the training set with the best hyperparameter configuration found through the tuning process. Specifically, we stored the predicted probabilities of a change in the cumulative incurred loss amounts and the corresponding true labels of the classification task and used a Receiver Operating Characteristic (ROC) curve to choose the threshold setting which maximizes the difference between the True Positive Rate and the False Positive Rate.[15]

Finally, we trained the model with the best hyperparameters on both the training and validation sets to predict the test set. For predicting future data points in the test set, we employed a recursive multi-step forecasting approach. In this method, the model uses its own predictions from previous

[15] For a comprehensive explanation of ROC curves, see Fan et al. (2006).

Table 3: Best hyperparameter settings from Bayesian Search

	Property		Liability	
	Cumulative	Incremental	Cumulative	Incremental
lr	0.004	0.00043	0.0851	0.0273
q_{static}	128	256	128	64
q_{lstm}	128	128	128	128
q_{comb}	128	128	128	128
$batch$	1024	1024	1024	1024
$dropout$	0.378	0.338	0.458	0.2509

time steps as input to forecast subsequent time steps. The time required for this training on a *Amazon-EC2-G5.8xlarge* instance is given in table 4 for for both data sets, separated by Random and Bayesian Search.

Table 4: Training time in seconds

	Property		Liability	
	Cumulative	Incremental	Cumulative	Incremental
Random Search	11764	12759	29460	33289
Bayesian Search	14521	16632	36235	39854