NATIONAL ECONOMICS UNIVERSITY
FACULTY OF MATHEMATICAL ECONOMICS

---------o0o---------

BACHELOR THESIS

MEASURING UNCERTAINTY THROUGH BAYESIAN NEURAL NETWORKS

An Application in Insurance Claim Cost Prediction

Student : Mai Tra My
Student ID : 11202622
Class : DSEB 62
Supervisor : PhD. Vuong Van Yen

Hanoi, May 2024

NATIONAL ECONOMICS UNIVERSITY

FACULTY OF MATHEMATICAL ECONOMICS

BACHELOR THESIS

MEASURING UNCERTAINTY THROUGH BAYESIAN NEURAL NETWORKS

An Application in Insurance Claim Cost Prediction

Student : MAI TRA MY
Student ID : 11202622
Class : Data Science in Economics & Business 62
Supervisor : PhD. Vuong Van Yen

Hanoi, May 2024

Acknowledgment

While completing this thesis, I have received much support from my supervisor, friends, and family. In this acknowledgment of my thesis, I extend my appreciation to all those who have been part of this journey.

Firstly, I would like to express my deep gratitude to my supervisor, PhD. Vuong Van Yen for his inspiration, guidance, and support about knowledge and writing concepts. This is valuable knowledge and experience that is not only useful in this graduation thesis but also in my career path.

I could not have undertaken this thesis without all the teachers of National Economics University, and the teachers of the Faculty of Mathematical Economics, in particular. Thanks to the valuable knowledge that the teachers imparted and the dedicated support throughout the implementation period, I have accomplished the thesis and achieved remarkable results.

Special thanks go to my friends who have commented on the techniques used in my notebook and helped me optimize the result.

Lastly, I would like to mention my family who have always been extremely supportive of the decisions I made.

Thank you all for being the important parts of my academic path.

Hanoi, May 2024

Author

Mai Tra My

Contents

List of Figures

List of Tables

Glossaries

BNNs	Bayesian Neural Networks
GLMs	Generalized Linear Models
GBM	Gradient Boosting Machine
RMSE	Root Mean Square Error
IQR	Interquartile Range
NN	Neural Network
CNN	Convolutional Neural Network
MNIST	Modified National Institute of Standards and Technology database
LDA	Latent Dirichlet Allocation
BERT	Bidirectional Encoder Representations from Transformers
VI	Variational Inference
MCMC	Markov Chain Monte Carlo
RNN	Recurrent Neural Networks
FNN	Feedforward Neural Networks
MAE	Mean Absolute Error
OOD	Out-of-distribution
LSTMs	Long Short-Term Memory Networks
ELBO	Evidence Lower Bound

EXECUTIVE SUMMARY

Motivation

Currently, machine learning models in general and deep learning in particular are widely used in all prediction and forecasting problems. Thanks to these techniques, complex problems, especially in the fields of image recognition, natural language processing, and speech processing, are solved with high accuracy. However, traditional deep learning models can only make predictions based on the training data, limiting their ability to new situations or contexts that have not been represented in the training data.

In traditional deep learning, the model predicts a point estimate for each input. For example, in the classic problem of predicting labels of images in the CIFAR-10 set, we will have a label for each image: the animal in the image is a butterfly, elephant, dog, or cat, etc. In the stock price prediction problem, each specific time point corresponds to a constant forecast value. We can determine the performance of the model based on some metrics for the entire model such as RMSE, accuracy, precision, and recall, but we cannot know how much correct prediction is for a specific observation. With this forecasting method, the result contains no uncertain information about the forecast results. In reality, results always have some error and for a randomly varying quantity, so point estimate results cannot fully reflect the uncertain nature of the predicted object.

It may be observed that there are some decision-making problems for which point estimates are not sufficient. For example, knowing the uncertainty in self-driving cars can help determine when the system should request human intervention. In medical diagnostics, understanding the uncertainty of a diagnosis can influence treatment plans and risk assessment. Without this information, there's a higher risk of misdiagnosis or inappropriate treatment.

Uncertainty is an essential part of models, especially in finance and insurance, where numerous random factors influence outcomes. Financial markets are highly volatile

and can change rapidly due to a multitude of factors that may not be included in the features. Traditional neural networks are prone to overfitting, which can lead to poor generalization of new, unseen data, making them unreliable in volatile conditions. In insurance, accurately pricing policies requires understanding the uncertainty and risk associated with different claims. Not concerning about it potentially leads to mispricing and financial losses.

The Role of Claim Cost Prediction

Claim cost prediction is a cornerstone of the insurance industry, influencing a wide range of operational and strategic decisions. Accurate prediction models help insurers manage risks, set appropriate premiums, ensure financial stability, and enhance customer satisfaction.

Predicting claim costs allows insurers to identify potential high-risk policyholders and take proactive measures to mitigate these risks. This might include adjusting policy terms, setting aside adequate reserves, or offering risk management advice to clients.

The financial stability of insurance companies hinges on their ability to accurately predict and reserve for future claims. Overestimating claim costs can result in unnecessarily high reserves, reducing the capital available for other investments and potentially leading to lower profitability. Conversely, underestimating claim costs can result in insufficient reserves, risking the company's ability to pay out claims and threatening its solvency. Accurate claim cost predictions help ensure that reserves are appropriately sized, maintaining the financial health of the insurer.

Predicting claim costs accurately can enhance customer satisfaction and retention. When insurers understand the true cost of claims, they can offer more accurate and fair premium rates, fostering trust among policyholders. Additionally, efficient claim processing and prompt payouts, supported by accurate predictions, improve the customer experience.

Objectives

The primary objective of this thesis is to develop a comprehensive understanding of Bayesian Neural Networks. This involves exploring the theoretical foundations of Bayesian neural networks, including the principles of prior and posterior distributions and the Bayesian inference process. By reviewing existing literature and methodologies,

the research will establish a solid conceptual framework for understanding how Bayesian neural networks can be effectively utilized in various applications.

A significant goal of the thesis is to implement Bayesian neural networks specifically for measuring uncertainty. The research will design and implement Bayesian neural networks, comparing their performance against traditional neural networks and a statistic model - Generalize Linear Model.

An important application of this research is in the field of insurance claim cost prediction. The thesis will develop a Bayesian neural network tailored for this specific application, incorporating relevant features and data specific to insurance claims. The aim is to create a model that not only predicts claim costs accurately but also provides meaningful uncertainty estimates that can be used in practical decision-making processes within the insurance industry.

Additionally, the thesis will provide recommendations for future research and development in the field of Bayesian neural networks and their application in insurance. This will include identifying potential areas for further investigation and suggesting improvements and innovations that could enhance the accuracy and applicability of uncertainty measurements in predictive modeling.

Research Questions

1. Why is uncertainty quantification important in modeling? How have previous studies approached this issue?
2. What models have been used to predict insurance claim costs?
3. How do Bayesian neural networks work?
4. How to process data and build a Bayesian neural network to predict insurance claim costs?
5. What are the advantages of the Bayesian neural network compared to other traditional models when applying to this problem?

Structure of the Thesis

The thesis consists of five chapters:

- Chapter 1: Literature Review. We review basic concepts of Uncertainty and the theory of Bayesian Neural Networks. Some approaches to the insurance claim cost prediction problem and challenges are provided.
- Chapter 2: Bayesian Neural Networks. In this part, we review traditional neural networks, the necessity of quantifying uncertainty, some types and approaches to uncertainty, and the theoretical basis of Bayesian Neural Networks. Some examples are included with the aim of emphasizing the advantages of models.
- Chapter 3: Methodology. We introduce the problem and dataset and present the pipeline of analysis and models.
- Chapter 4: Data Analysis and Results. We conduct exploratory data analysis to analyze and investigate the dataset and show the results of the models.
- Chapter 5: Conclusion and Future Research.

Chapter 1

LITERATURE REVIEW

1.1 Bayesian Neural Networks

Deep Learning is a major turning point in the field of artificial intelligence. It allows other data scientists to build many high-accuracy models in various fields including computer vision, natural language processing, speech recognition, and recommendation systems (Alzubaidi et al., 2021). Recently, with the explosion of data volume, Deep Learning has replaced and gradually occupied traditional Machine Learning problems.

In the past ten years, many contributions and research directions have been made to improve models based on deep neural networks. We can mainly divide them into two categories: the use of deep learning in specific problems, or developing general models, helping models have higher accuracy and robustness.

Traditional deep learning models often only give a certain point estimate of parameters and predictions for the input and are unable to determine whether the model is certain about its output. Therefore, one research direction is to develop a way to quantify model uncertainty, which is included in the topic of this thesis.

"Uncertainty: The Importance of Knowing What We Don't Know" (Gal, 2016)

Uncertainty is a multifaceted concept that can significantly impact model performance and decision-making. Understanding and quantifying uncertainty allows models to make more informed predictions and provides insights into their reliability. There are two types of uncertainty appearing in machine learning models: aleatoric and epistemic uncertainty. (Kiureghian & Ditlevsen, 2009)

Aleatoric uncertainty, also known as statistical or irreducible uncertainty, arises from the inherent noise in the data. It represents the variability in the data that cannot be reduced, even with more data or better models. Aleatoric uncertainty can be further divided into homoscedastic (constant across different inputs) and heteroscedastic (variable across different inputs) uncertainties. (Kendall & Gal, 2017)

Epistemic uncertainty, also known as model uncertainty represents the model's lack of knowledge. The difference between the two types of uncertainty is that aleatoric uncertainty cannot be reduced only identified and quantified, while epistemic uncertainty can be reduced through more comprehensive study. (Bjarnadottir et al., 2019)

Measuring Epistemic Uncertainty

To capture epistemic uncertainty in predictions, there are three main approaches:

- *Bayesian approaches:* Bayesian methods incorporate prior distributions over model parameters and update these distributions in light of the observed data. This approach inherently accounts for uncertainty in the model parameters (MacKay, 1992).

- *Uncertainty representation through the joint distribution of predictions:* In the research from Google Deepmind (Osband et al., 2023), it is shown that joint predictions are essential for accurately assessing uncertainty and making informed decisions. It is suggested that minimizing joint log-loss leads to optimal actions, whereas minimizing marginal log-loss does not. The paper also introduces a new epistemic NNs architecture called **epinet**, which enhances existing neural networks with a small auxiliary network to generate uncertainty estimates.

- *Conformal prediction:* Conformal prediction is a straightforward way to generate prediction sets for any model (Angelopoulos & Bates, 2022). The key idea behind conformal prediction is to produce prediction intervals or sets that cover the true outcome with a pre-defined probability. This is achieved through a nonparametric, distribution-free method that can be applied to any machine learning algorithm.

Bayesian Neural Networks

Bayesian Neural Networks (BNNs) are a class of neural networks that integrate Bayesian inference into their framework. This integration provides a robust method for

quantifying uncertainty and improves the reliability of model predictions.

BNNs are grounded in Bayesian statistics. Bayes' theorem forms the basis for updating the posterior distribution of model parameters. By maintaining distributions over the model parameters instead of point estimates, BNNs can provide probabilistic predictions that reflect both aleatoric and epistemic uncertainties.

Training BNNs involves approximating the posterior distribution of the model parameters. Due to the high dimensionality and complexity of neural networks, exact Bayesian inference is often intractable, necessitating the use of approximation methods such as:

Variational Inference (VI): VI approximates the true posterior distribution with a simpler, parametric distribution by optimizing the variational parameters to minimize the Kullback-Leibler (KL) divergence between the true posterior and the approximate distribution. Blundell et al. (2015) introduced a variational approach for weight uncertainty in neural networks.

Markov Chain Monte Carlo (MCMC): MCMC methods, such as Hamiltonian Monte Carlo (HMC), sample from the posterior distribution using a series of random walks guided by the gradient of the posterior distribution. While MCMC provides more accurate posterior estimates, it is computationally expensive (Neal, 1996).

In this thesis, we use Variational Inference to approximate the posterior distributions.

Applications of Bayesian Neural Networks

BNNs have been successfully applied in various fields, demonstrating their ability to enhance predictive performance and reliability by incorporating uncertainty.

In autonomous driving and robotics, BNNs enhance the reliability of control systems by incorporating uncertainty into predictions. This is crucial for safety-critical applications where understanding the uncertainty of the model's decisions can prevent catastrophic failures. McAllister et al. (2017) discussed the advantages of Bayesian deep learning in autonomous vehicle safety.

The way to evaluate computer vision models has become more comprehensive with the emergence of BNNs. BNNs can be used to analyze medical images, such as X-rays, MRIs or breast histopathology image, providing uncertainty estimates that help radiologists assess the confidence of the detected anomalies (Khairnar et al., 2020).

BNNs are used for risk assessment, stock price prediction, and portfolio optimization. By quantifying uncertainty, BNNs help in making more informed financial decisions,

accounting for the inherent risks and variabilities in financial markets (Hauzenberger et al. (2023), Back and Keith (2019), Chandra and He (2021), Olsgärde (2021)).

In this thesis, we use BNNs as a technique to measure uncertainty in insurance claim cost prediction. The next section will review several approaches that were applied to give predictions in this problem.

1.2 Approaches to the Insurance Claim Cost Prediction Problem

Insurance claim cost prediction is critical for insurers, enabling accurate premium pricing, risk assessment, and financial planning. The complexity and variability of insurance claims necessitate sophisticated predictive models that can handle diverse data types and capture intricate relationships. We have compiled several previous studies on insurance cost prediction models, including traditional models, machine learning approaches, and deep learning.

Traditional Statistical Methods

Generalized Linear Models (GLMs) are widely used in the insurance industry due to their simplicity and interpretability. GLMs extend linear regression models to accommodate different types of response variables and distributions, making them versatile for various types of insurance data. These models are highly valued for their interpretability, allowing actuaries to understand the influence of individual predictors on the outcome. An analysis of the portfolio of vehicle insurance data using a GLM is performed by Kafková and Krivankova (2014). Quijano Xacur and Garrido (2015) explored the differences between using Tweedie and the compound Poisson-gamma distribution, contrasting the advantages and disadvantages of each. The authors found that with regards to the pure premium, Tweedie should be used, whenever the parsimony principle applies. By contrast, in general, one should choose other models in order to conclude the claim frequency and severity.

Although it is easy to explain the effect of independent variables on the result, the disadvantage is that GLMs need many assumptions about the distribution of the data. If these assumptions are not satisfied, the predictions will basically be inaccurate.

Machine Learning Approaches

Decision trees and their ensemble forms, such as Random Forests and Gradient Boosting Machines (GBMs), have gained popularity due to their ability to handle complex interactions. Wuthrich (2016) employed a decision tree for auto insurance claim predictions, obtaining claims reserves on individual claims respecting all available relevant feature information (which also differentiates types of claims). In the other research, the authors used three different classifiers including the decision tree, Random Forest and XGBoost. They found that models based on XGBoost outperformed models of other classifiers (Sahai et al., 2023).

Deep Learning Approaches

With advancements in computational capabilities, deep learning has emerged as a powerful tool for tackling the problem. It is useful for capturing non-linear interactions and can achieve high accuracy. Feedforward Neural Networks (FNNs) are among the most basic forms of deep learning models. They consist of multiple layers of neurons where the data moves in one direction—from input to output. Abdulkadir and Fernando (2024) demonstrated the use of a feed-forward neural network for predicting insurance claims.

Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs) are designed to handle sequential data and temporal dependencies, making them suitable for time-series prediction in insurance. According to the research of Goundar et al. (2020) using artificial neural networks, RNN model also outperformed FNN in terms of accuracy.

Challenges

Data quality and availability pose significant challenges in predicting insurance claim costs. Insurance data is often noisy, incomplete, and imbalanced, making it difficult to train reliable models. Ensuring high-quality data and dealing with missing values are critical steps in building robust predictive models. Techniques such as data cleaning, preprocessing, and imputation methods are essential to address these issues.

The trade-off between interpretability and accuracy is another significant challenge. While advanced models like neural networks provide higher accuracy, they often lack interpretability, making it difficult to understand how predictions are made. This is

particularly problematic in the insurance industry, where decisions based on model predictions can have substantial financial implications.

This chapter has presented an overview of research on ways to measure Uncertainty and approaches to the problem of predicting claim costs in insurance. The following chapter will present the theory as well as the necessary proofs that lay the foundation for the Bayesian Neural Network - an approach to measuring uncertainty level.

Chapter 2

BAYESIAN NEURAL NETWORKS

2.1 A Review of Neural Networks

Deep learning is a subset of artificial intelligence (AI) and machine learning (ML) that focuses on developing algorithms inspired by the structure and function of the human brain's neural networks. These algorithms, known as *artificial neural networks*, are composed of layers of interconnected nodes. Deep learning has gained significant attention and popularity due to its ability to learn from vast amounts of unstructured data, such as images, sound, text, and video.

Deep learning models, or neural networks in general, have multiple layers. They contain three types: the input layer, the hidden layer(s), and the output layer.

Considering an input vector $x \in \mathbb{R}^p$ and an output vector $y \in Y$, a feed-forward neural network with m hidden layers, each containing h_i nodes, along with weight w, bias b and a non-linear activation function ϕ:

$$
\begin{aligned}
h^{(1)} &= \phi(w^{(1)}x + b^{(1)}) \\
h^{(2)} &= \phi(w^{(2)}h^{(1)} + b^{(2)}) \\
&\vdots \\
h^{(m)} &= \phi(w^{(m)}h^{(m-1)} + b^{(m)}) \\
y^* &= g(w^{(m+1)}h^{(m)} + b^{(m+1)}).
\end{aligned}
$$

The function g can be linear, which is often for regression tasks or non-linear for classification. (Back & Keith, 2019)

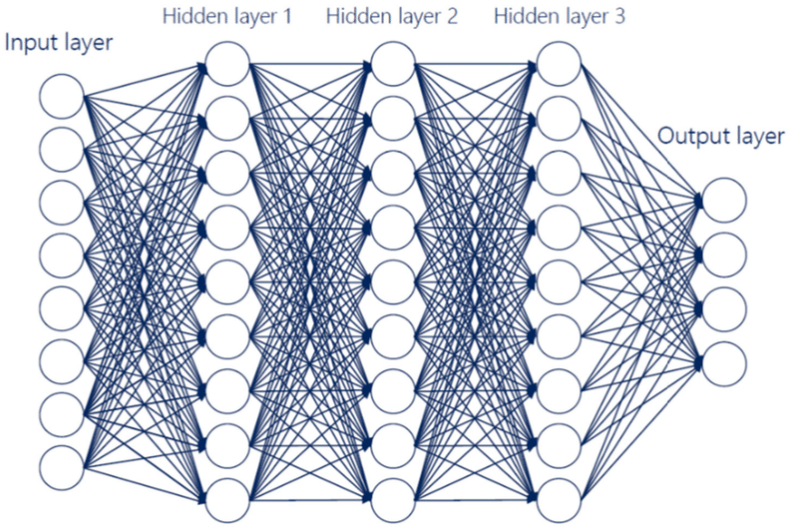

Figure 2.1: An Artificial Neural Network (NN) model
(Source: Internet)

Establishing a deep learning system involves two key stages. First, we select an architecture. Then, we tune the model's weights to optimally represent the training data. This fitting step typically employs a method known as gradient descent. (Duerr et al., 2020)

The main reason why neural network models stand out compared to machine learning models is their ability to solve problems of non-linear separable data. With the aim of approximating complex non-linear transformations in a neural network, some form of non-linearities are used. Two of the most familiar ones are the sigmoid function, which has the form:

$$\phi(x) = \frac{1}{1+e^{-x}}$$

and hyperbolic tangent:

$$\phi(x) = tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-1}}$$

However, these functions can lead to a serious problem called vanishing gradients. When gradients converge to 0, the network will be unable to update and learn from the data. To solve this problem, Rectified Linear Units (ReLU) was proposed.

$$\phi(x) = ReLU(x) = max(0, x)$$

ReLU has been preferred in recent years because of its higher speed of calculating and the faster convergence speed.

Due to the objectives and characteristics of a project such as image classification and language translation, we utilize different architectures. Some popular models are Convolutional Neural Networks (CNNs), Long Short Term Memory Networks (LSTMs), Recurrent Neural Networks (RNNs), Generative Adversarial Networks (GANs), Transformers (GPT, BERT), Autoencoders, etc.

Because of the complex neural networks, these models often lead to overfitting. There are some methods to prevent this problem called *regularizations*.

L^1-regularization

L^1-regularization is also called Lasso regularization. It penalizes large weights in terms of absolute value with parameter $\lambda \in R^+$.

$$L(y^*|X^*, w) + \lambda \sum_{i}^{m} \sum_{j}^{h_i} |w_{i,j}|$$

where $L(y^*|X^*, w)$ denotes the loss function of a new prediction.

L^2-regularization

L^2-regularization is also known as Ridge regression. It penalizes large weights by adding L^2-penalty to the model loss function.

$$L(y^*|X^*, w) + \lambda \sum_{i}^{m} \sum_{j}^{h_i} (w_{i,j}^2)$$

Dropout

Dropout is a technique that adds noise to the model by randomly deactivating selected neurons during training. This means that the contributions of these neurons to the activation of subsequent neurons are temporarily excluded during the forward pass, and their weights are not updated during the backward pass in backpropagation.

During training, dropout regularizes the model by randomly deactivating neuron activations with a specified probability p, which is a hyperparameter. In the testing phase,

no neurons are deactivated, but the activations of all neurons in the network are scaled by p.

The use of dropout is the first attempt to introduce uncertainty into deep neuron networks and is often considered a special case of Probabilistic Deep Learning. Monte Carlo dropout is widely popular for its simplicity and effectiveness among the several options to quantify uncertainty. (Verdoja & Kyrki, 2021)

2.2 Uncertainty

Neural networks are very good at processing complex input data in high-dimensional space, making the best point estimate based on the trained data. However, it encounters many problems when forced to make predictions for data points that are out-of-distribution (OOD).

Ideally, when faced with such out-of-distribution data, we expect that the model will not only predict but also indicate some lights on the deviation of a data point from its trained range (similar to outputting confidence interval in regression problems in statistics). In essence, we want the model to exhibit high levels of uncertainty with respect to these inputs if we do not have enough information to make the predictions (or conversely, high confidence when it is enough).

Sources of uncertainty in machine learning models

In the following, we list some situations that can cause uncertainty:

- *Noisy data*: The observed labels might be noisy, leading to uncertainty in predictions. We call this **aleatoric uncertainty**.
- *Model parameter uncertainty*: When multiple models can effectively explain a data set, choosing the optimal model parameters for prediction becomes uncertain (uncertainty in cognitive aspect).
- *Uncertainty about model architecture*: This uncertainty is related to the choice of model architecture itself. How to design a model for optimal extrapolation or interpolation?

The latter two uncertainties in the model are called **epistemic uncertainty**, as distinct from aleatoric uncertainty. Both types contribute to the uncertainty of predictions, reflecting our overall confidence in the model's predictions.

Aleatoric uncertainty is the noise level during the data-generating process (*inherent noise*). Whenever there is randomness involved, and the observed result cannot be entirely determined by the input, it is necessary to address this intrinsic uncertainty in data.

Epistemic uncertainty (or model uncertainty) arises because it is impossible to estimate the model parameters without any doubt. This is due to the limitation of having finite training data; typically, the training data does not encompass all conceivable scenarios.

Classic neural networks can not distinguish these two types of uncertainty. In the following, we can observe a simple example presented in a recent article from Google Deepmind on Epistemic Neural Networks (Osband et al., 2023). Assume that we have a picture 2.2 below. The task is to build a model that can classify whether the object centering in the picture is a rabbit or a duck.

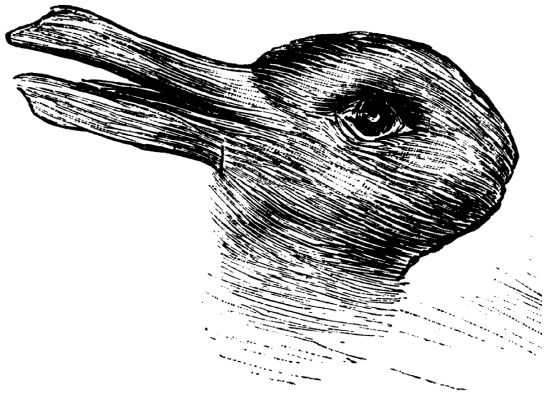

Figure 2.2: Question: Duck or Rabbit?
(Source: Internet)

Assume that the model generates a prediction that provides probabilities 50-50 for both classes. Now it is uncertain whether this equality arises from a random sampling of labels or if the network would lean towards a single class with more training data. Traditional neural networks do not differentiate between these scenarios, yet decision-making systems must understand their uncertainties. Joint predictions can be taken into account to identify whether it is an ambiguous object or it is something that can be

learned.

Now we ask the trained model to make two predictions. The joint distribution of predictions is over pairs of labels $(y_1, y_2) \in (R, D) * (R, D)$ for the same image. For any joint prediction, Bayes' rule defines a conditional prediction for y_2 given y_1. If all outcomes have equal probabilities $(0.25, 0.25, 0.25, 0.25)$, it indicates that more training data does not impact the result, and conditioning on the first label does not affect the second prediction. However, if it is epistemic, the second prediction depends on the first one and they are either rabbits or ducks. In this case, additional training could help address the issue of uncertainty.

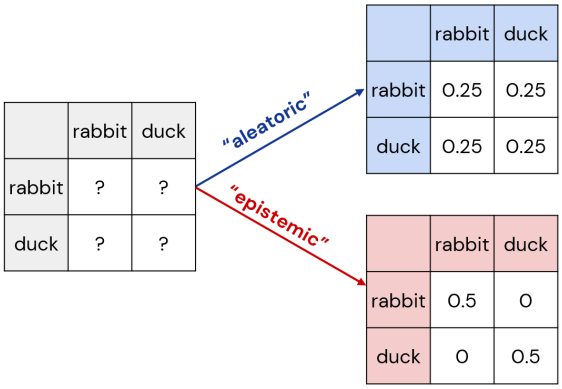

Figure 2.3: Joint Distribution in Two Cases (Osband et al., 2023)

2.3 Bayesian Learning

2.3.1 Bayesian Inference

Given a set of data $D = (X, Y)$. Let $p(\theta)$ be the prior distribution of the parameter $\theta \in \Theta$ of the model that generated the data. The likelihood distribution $p(D|\theta)$ is the link between the parameters and the observed data. When observing new data, we use likelihood distribution and prior beliefs to calculate the posterior distribution (according to Bayes' theorem):

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)} \tag{2.1}$$

$$= \frac{p(\theta)\prod_i p(y_i|x_i, \theta)}{p(D)} \tag{2.2}$$

The denominator of equation (2.1) is called marginal likelihood.

$$p(D) = \int_{\Theta} p(D|\theta)p(\theta)d\theta$$

Then, we can predict a new data point (x^*, y^*) following the posterior predictive distribution (Gal, 2016):

$$\begin{aligned} p(y^*|x^*, D) &\approx E_{p(\theta|D)}[p(y^*|x^*, \theta)] \\ &= \int_{\Theta} p(y^*|x^*, \theta)P(\theta|D)d\theta \end{aligned}$$

In Bayesian inference, the posterior distribution represents our uncertainty about the model. Based on the prior distribution of the model parameters, it is possible to update the uncertainty when observing the data and arrive at the posterior distribution. Given a new observation, x^*, the posterior prediction distribution $p(y^*|x^*, D)$ represents our belief about the label y^*. We can use the variance of this distribution to quantify our uncertainty. According to the law of total variance:

$$Var(X) = E[Var(X|Y)] + Var(E[X|Y])$$

We can factor the variance into two terms (Zhu & Laptev, 2017):

$$Var(p(y^*|x^*, D)) = E[Var(p(y^*|x^*, D, \theta)] + Var(E[p(y^*|x^*, D, \theta])$$

The first term is the noise level during the data-generating process (*inherent noise*), which is referred to as *aleatoric uncertainty*. The second one reflects *epistemic uncertainty*.

2.3.2 Computation Methods

To perform Bayesian inference, we need efficient methods for calculating the posterior distribution. In fact, it is a difficult problem. Here we will list some commonly used methods to approximate the posterior distribution.

Markov Chain Monte Carlo is a powerful technique that originated from computational problems in statistical physics. The fundamental concept behind MCMC involves drawing samples from a Markov chain that is ergodic, meaning it has the actual posterior distribution as its stationary distribution. As a result, over time, the sampling process will approach and converge to the true distribution. MCMC algorithms excel at

sampling from the exact posterior distribution. Nonetheless, their limited scalability has diminished their appeal for Bayesian Neural Networks.

The second approach is *Variational Inference*, which scales much better than MCMC. We will discuss in more detail about this approximation method in the next section on Bayesian Neural Networks.

	MCMC	Variational Inference
Examples	• Gibbs Sampling • Metropolis Hasting • Hamiltonian MC	• Stochastic Variational Inference • Bayes by Backprop

Table 2.1: Some Computation Bayesian methods (Jospin et al., 2022)

2.4 Bayesian Neural Networks

2.4.1 Introduction to Bayesian Neural Networks

Bayesian Neural Networks are a type of probabilistic machine learning model that incorporates Bayesian inference into the training and prediction process. They are particularly useful when dealing with uncertainty estimation in neural networks.

In a Bayesian neural network, with an input structure similar to that of a classic neural network, the model comprises m hidden layers with h_i nodes in each layer. Unlike classic neural networks, the weights W are represented as distributions rather than fixed (unknown) values, and the biases B are also represented as distributions:

$$
\begin{aligned}
h^{(1)} &= \phi(W^{(1)}x + B^{(1)}) \\
h^{(2)} &= \phi(W^{(2)}h^{(1)} + B^{(2)}) \\
&\vdots \\
h^{(m)} &= \phi(W^{(m)}h^{(m-1)} + B^{(m)}) \\
y^* &= g(W^{(m+1)}h^{(m)} + B^{(m+1)}).
\end{aligned}
$$

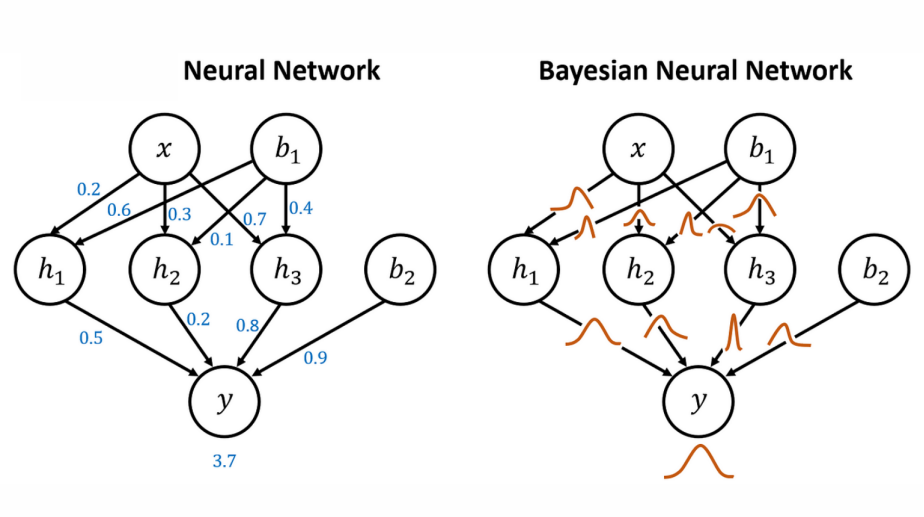

Figure 2.4: Traditional Neural Network vs. Bayesian Neural Network
(Source: Internet)

2.4.2 Variational Inference

The idea behind the Variational Inference Bayes method is that the complicated posterior distributions of the weights are approximated by a simple distribution called variational distribution, such as Gaussian distribution.

Besides, we also need to define a prior distribution. A common choice is the standard normal, N(0, 1), as the prior.

Now the parameter θ replaces the weights and biases of the non-Bayesian neural network (w, b). The parameter θ in a Bayesian network isn't fixed but follows a distribution. We approximate posterior with a simple, variational distribution, $q(\theta)$ such as a Gaussian. For each Gaussian, you have two parameters: $\lambda = (\mu, \sigma)$. These are called variational parameters. The job of Variational Inference is to tune the variational parameter λ so that $q(\theta)$ gets as close as possible to the true posterior $p(\theta|D)$.

These are some important concepts in the theory of Variational Inference.

Evidence Lower Bound

Evidence lower bound (ELBO) is also called variational lower bound. Starting with log of evidence, we have:

$$\begin{aligned} log p(D) &= log \int_\theta p(D,\theta) \\ &= log \int_\theta p(D,\theta)\frac{q(\theta)}{q(\theta)} \\ &= log \mathbb{E}_q \left[\frac{p(D,\theta)}{q(\theta)}\right] \\ &\geq \mathbb{E}_q \left[log \frac{p(D,\theta)}{q(\theta)}\right] \\ &= \mathbb{E}_q \left[log p(D,\theta)\right] - \mathbb{E}_q \left[q(\theta)\right] \end{aligned}$$

with $q(\theta)$ can be viewed as the variational distribution.

$\mathbb{E}_q[q(\theta)]$ is called Entropy of q and the right-hand side of the equation is the ELBO.

Kullback–Leibler divergence

Kullback–Leibler divergence (KL divergence, denoted by $D(f\|g)$) is a metric to measure the similarity between two density distributions, also known as relative entropy (Hershey & Olsen, 2007). It is defined as:

$$D(f\|g) = \int f(x) log \frac{f(x)}{g(x)} dx$$

KL divergence always needs to satisfy the following three properties:

- Self similarity: $D(f\|f) = 0$
- Self identification: $D(f\|g) = 0$ only when $f = g$
- Positivity: $D(f\|g) \geq 0$ for all f, g

Using KL divergence for continuous distributions - the posterior p and the variational

distribution q, it can be rewritten as:

$$\begin{aligned} D(q(\theta)||p(\theta|D)) &= \mathbb{E}_q\left[log\frac{q(\theta)}{p(\theta|D)}\right] \\ &= -\mathbb{E}_q\left[log\frac{p(\theta|D)}{q(\theta)}\right] \\ &= -\left(\mathbb{E}_q\left[log\frac{p(\theta,D)}{q(\theta)}\right] - \mathbb{E}_q[logp(D)]\right) \\ &= -(\mathbb{E}_q\left[logp(\theta,D) - logq(\theta)\right]) + logp(D) \end{aligned}$$

The first term on the right-hand side of the equation is ELBO, denoted by $\mathcal{L}(q)$. When moving terms, the equation becomes:

$$logp(D) = \mathcal{L}(q) + D\left(q(\theta)||p(\theta|D)\right) \tag{2.3}$$

Firstly one has to notice that the log of evidence in equation (2.3) is constant. Therefore, the distribution p and q will be more similar, or $KL(q||p)$ reaches the minimum value when ELBO is maximized.

Mean-field Approximation

In the previous subsection, we use θ to denote an unknown variable. We can generalize it to N unknown variables: $\theta = (\theta_1, ..., \theta_N)$. Mean-field approximation aims to approximate the distribution q by partitioning variables into some independent parts. The variational distribution is factorized over the latent variables, $q(\theta_i)$.

$$p(\theta|D) \approx q(\theta) = q(\theta_1, ..., \theta_N) = \prod_{i=1}^{N} q_i(\theta_i) \tag{2.4}$$

The function above is the generic member of the mean-field variational family. As seen in equation (2.4), the variational family is independent of the data. The data only comes with ELBO.

2.4.3 Advantages of Bayesian Neural Networks over Traditional Neural Networks

Bayesian neural networks offer several advantages:

- **Incorporation of Prior Knowledge:** Prior information or beliefs can be included in the analysis through prior distributions. This is especially useful when there is existing knowledge about the parameters being estimated.
- **Uncertainty Quantification:** Traditional neural networks provide point estimates, but Bayesian neural networks provide a natural way to quantify uncertainty through distribution over predictions. This means that in addition to estimating parameters, we also get a sense of how confident we are in those estimates. They allow for more robust decision-making, especially in critical applications like medical diagnosis or autonomous driving.
- **Flexibility with Data:** Bayesian neural networks can be adjusted and updated as new data becomes available. This makes them particularly useful in dynamic environments where data is continually being collected.

2.4.4 TensorFlow Probability: A Library to Build Bayesian Neural Networks

To develop Bayesian models, we have some libraries such as Pyro, PyMC3, pgmpy, PyBN and TensorFlow Probability (TFP). In this thesis, we use TensorFlow, Keras and TensorFlow Probability as a tool to build such models.

TensorFlow Probability is a library built on top of TensorFlow that focuses on probabilistic modeling and Bayesian inference. It provides tools for building probabilistic models, including probability distributions, probabilistic layers, and inference algorithms.

This is followed by some layers popularly used to build Bayesian neural networks.

DenseVariational

This is a dense layer with a random kernel and bias. This layer uses variational inference to fit a "surrogate" posterior to the distribution over both the kernel matrix and the bias terms. It extends the standard Dense layer in TensorFlow by allowing weights and biases to be treated as random variables with specified prior and posterior distributions. This enables the incorporation of uncertainty in the weights, making the model more robust and capable of providing uncertainty estimates in predictions.

DenseReparameterization

DenseReparameterization is a densely-connected layer class with a reparameterization estimator. It uses the reparameterization estimator by Kingma and Welling (2022), which performs a Monte Carlo approximation of the distribution integrating over the kernel and bias.

DenseFlipout

DenseFlipout is a class for a densely-connected layer that uses the Flipout estimator. This layer performs Bayesian variational inference similar to a dense layer by treating the kernel and/or bias as being drawn from distributions. Typically, it executes a stochastic forward pass by sampling from the posterior distributions of the kernel and bias. Flipout estimator (Wen et al., 2018) performs a Monte Carlo approximation of the distribution integrating over the kernel and bias.

Flipout uses roughly twice as many floating point operations as the reparameterization estimator but has the advantage of significantly lower variance.

2.5 Some Toy Examples

To illustrate the use of Bayesian neural networks (BNNs), we will present simple examples of constructing BNNs for classical problems. First, a regression problem will be addressed using data generated from a complex function, followed by a classification problem using the well-known MNIST handwritten digit dataset.

2.5.1 Curve Fitting and Regression

Dataset

The first example involves simulated data, where the dependencies are well-defined, making it straightforward to observe the model's functioning. In this thesis, we choose a function to calculate y according to x, incorporating additional noise. This function represents a complex non-linear equation:

$$y(x) = x + 0.3 * \sin(2 * \pi * (x + \epsilon)) + 0.3 * \sin(4 * \pi * (x + \epsilon)) + \epsilon,$$

where $\epsilon \sim N(0, 0.02)$.

The data is generated as illustrated in Figure 2.5. The training data consists of a range of x: $(0, 0.5)$. To evaluate the model, the test set includes 3000 x points ranging from -0.2 to 1. The purpose of this approach is to observe how the model predicts samples that are not in the training set; and assumes that the model cannot fit the curve at locations x outside the training set.

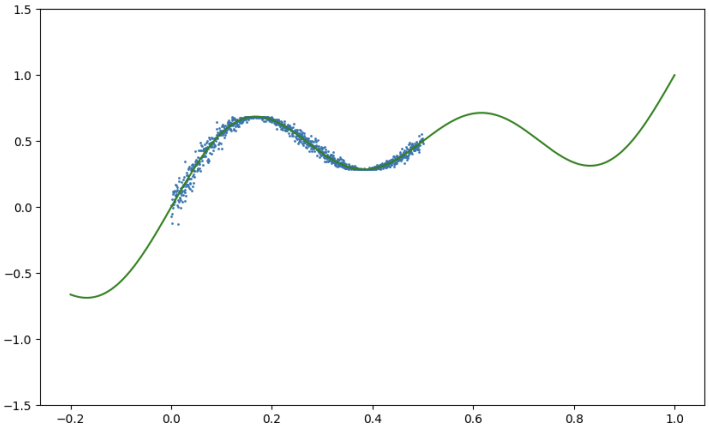

Figure 2.5: Simulated Data

Result

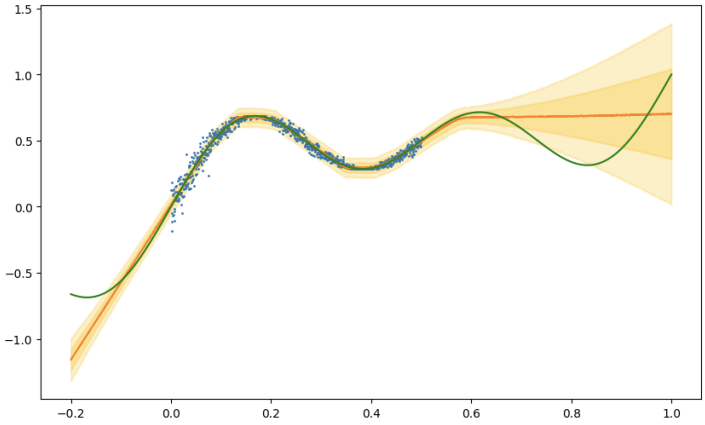

Figure 2.6: Result: Predictions of the Model

The model used is a simple Bayesian neural network with 2 hidden layers. As can be seen in the figure, in addition to the red line being the expectation of y, the model also provides a distribution for each of those points, with the inner yellow boundary being the $mean \pm std$ and the outer being $mean \pm 2 * std$ range. In intervals where there is no

data, the model represents uncertainty by giving a very wide distribution, such as at two tails (figure 2.6).

2.5.2 Classification

Dataset & Problem

The MNIST dataset is a collection of 70000 grayscale images of handwritten digits that are commonly used for training various image processing systems. There are also various modified versions of the MNIST dataset, such as Fashion-MNIST (images of clothing items) and EMNIST (Extended MNIST, which includes letters as well as digits). In this section, we use MNIST in the training step and both MNIST and Fashion-MNIST to evaluate and visualize the performance of the model.

Our problem with the MNIST data set is to classify that handwritten number image into labels from 0 to 9. Most standard implementations of neural networks achieve an accuracy of $98 \sim 99$ % in correctly classifying the handwritten digits. This proves that the model can recognize almost any number given. However, if we feed the model another image (such as an image of clothes or shoes), the model will still return a label in the numbers 0 to 9. This happens because the model does not have enough knowledge about the inputs put into the model. In this case, we would expect the model not to make a confident prediction. For most deep learning models, it is possible to set up a probabilistic version of the model, including Bayesian neural networks. Since they can estimate this uncertainty, we will try to use a Bayesian CNN model and compare the results with a traditional deep-learning model.

Result

First, we will look at the results of a CNN model on the problem of predicting labels of hand-written digits. Each item is paired with a bar chart that represents the model's confidence in its classification (Figure 2.7).

The model has relatively high accuracy with an accuracy of approximately 0.97. However, it can be seen that the model is very confident in the predicted results. The bar charts represent the softmax function value corresponding to each predicted label. Softmax values of predicted labels almost reach 0.9 for all cases, including wrong predictions (Figure 2.8).

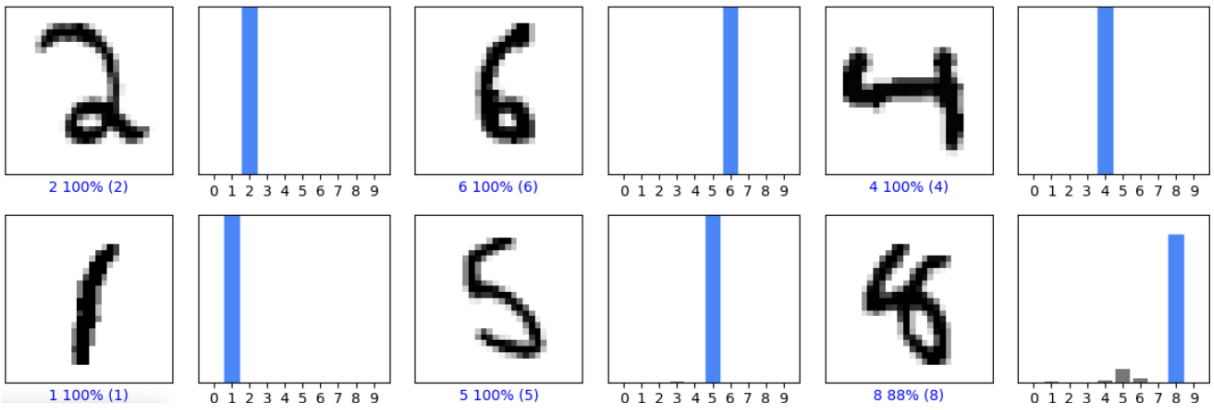

Figure 2.7: MNIST: CNN Prediction

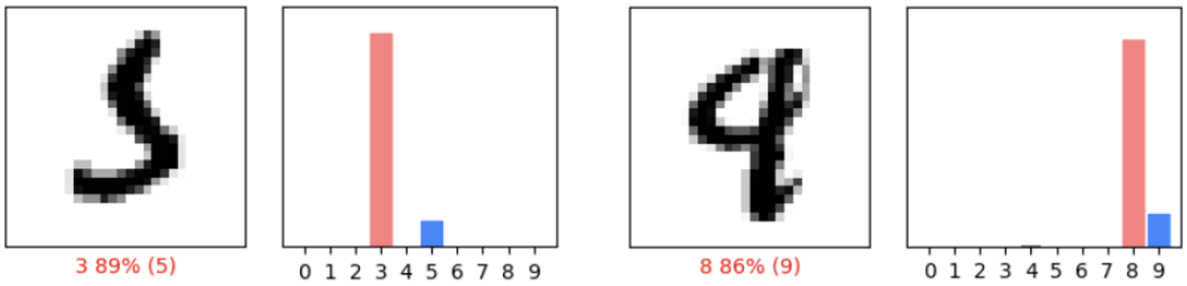

Figure 2.8: Wrong Predictions

When applying the above model to the Fashion MNIST set, the situation seems worse. Despite the high confidence scores shown, with about 98 percent of the test items having the softmax equal 100%, it is noted that all classifications are incorrect. In figure 2.9, the model predicts that the images are single digits, while in reality, they contain a boot, dress, t-shirt, pullover and sneaker.

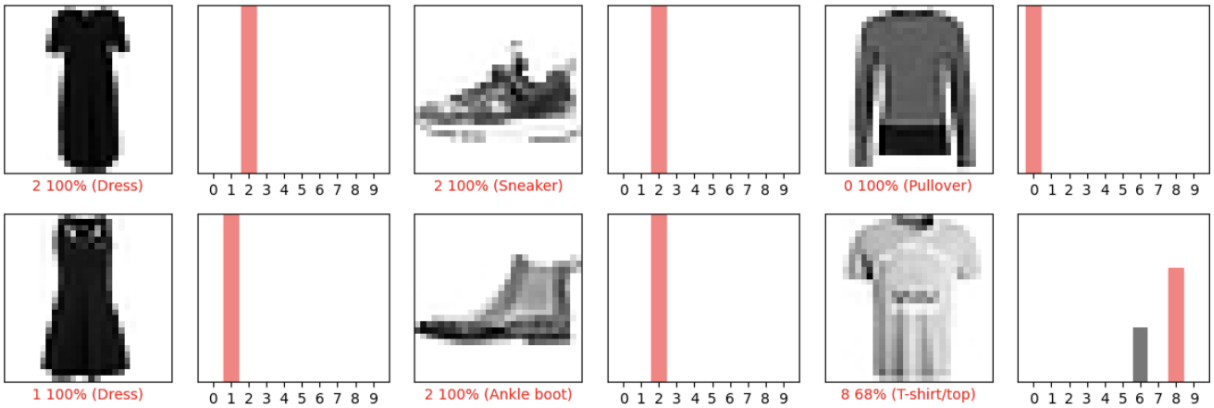

Figure 2.9: Fashion MNIST: CNN Prediction

Thus, using a point-estimate prediction model is not enough. Due to this problem, we move to another solution: Bayesian CNN.

Bayesian CNN addresses model uncertainty with noisy data and untrained data. Unlike traditional CNN, the softmax values are more evenly distributed across the labels, reflecting a more cautious approach. The model captures the uncertainty in its predictions, which is valuable for understanding the model's confidence and reliability.

In the two cases following, it predicts the digits "2" and "9" with less confidence compared to the NN (Note that it is still enough to make a decision if we still want to choose a single point estimate, but that is not the most important thing). Moreover, the BNN also shows a spread of predictions over other digits, indicating some uncertainty.

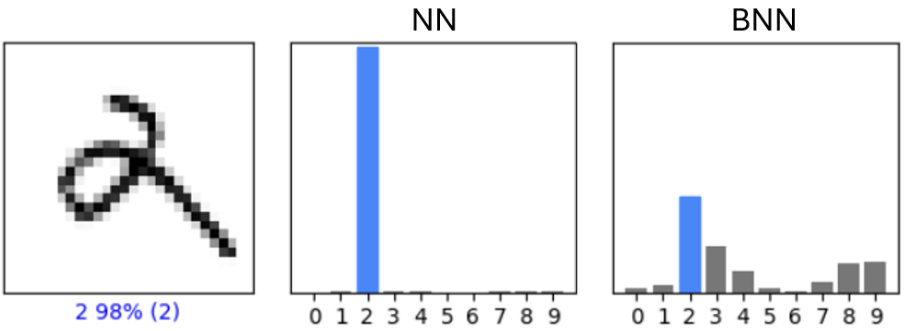

Figure 2.10: Case: 2

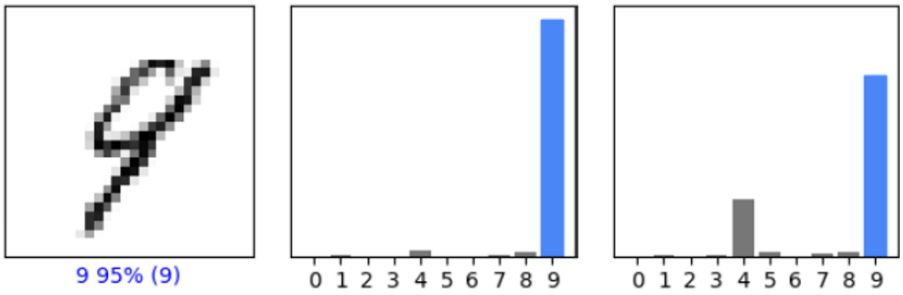

Figure 2.11: Case: 9

These are two cases that were misclassified when using the traditional neural network. By using BNN, they have been classified correctly (Figure 2.12).

Returning to the Fashion-MNIST dataset, the bar charts show a spread of predictions across different classes (Figure 2.13). The model no longer makes confident predictions. The spread of predictions and low confidence levels prove the uncertainty of the model about its classifications. This also implies that if we use it with a certain threshold for decision making, we need to consider certain risks.

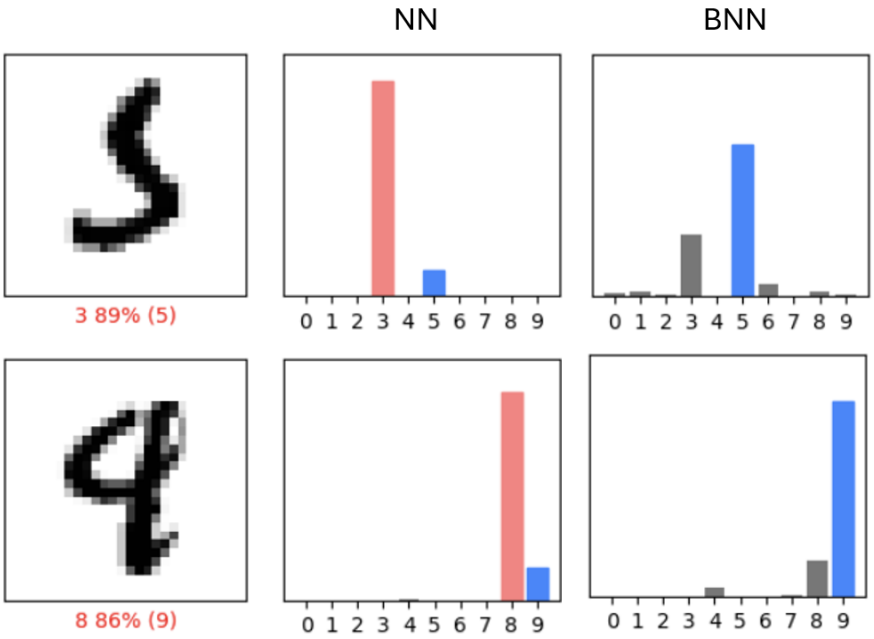

Figure 2.12: From Wrong to True Cases

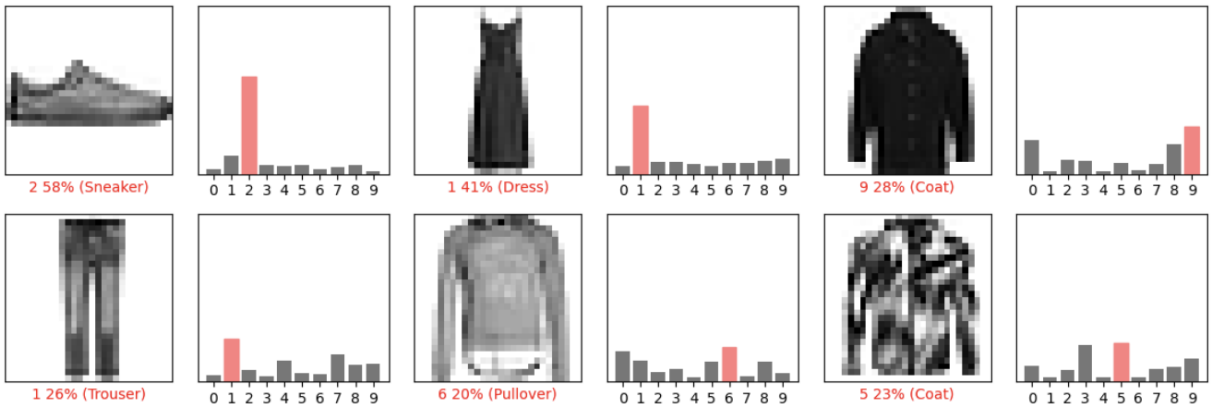

Figure 2.13: Fashion MNIST: Bayesian CNN Prediction

Chapter 2 presents an overview of Deep Learning, Uncertainty, Bayesian Learning and Bayesian Neural Networks. In terms of Bayesian Networks, we have mentioned the main idea, formula, and focus on Variational Inference, which is one of the most popular methods. Finally, to illustrate the advantages of this model, some examples have been introduced. In the next chapter, we introduce the dataset used to predict insurance claim costs and explain the pipeline for processing data and models.

Chapter 3

APPLICATION: INSURANCE CLAIM COST

3.1 Topic

Calculating insurance claim cost is a daily task of insurers. Besides, predicting the ultimate incurred claim cost based on provided data is crucial for them to manage their financial risks effectively and make informed decisions about reserves and pricing.

The initial incurred claim cost represents the initial estimate made by the insurer at the onset of a claim. It is the insurer's first assessment of the expected cost of settling the claim. This estimate is typically made soon after the claim is reported and is based on initial information such as the nature of the claim, policy coverage, initial assessment of damages or losses, etc.

Ultimate incurred claim cost refers to the total amount of claims payments that the insurance company ultimately disburses for a particular claim. It includes all payments made throughout the entire lifecycle of the claim, from the initial estimate to the final settlement. Ultimate incurred claim cost reflects the actual total financial impact of the claim on the insurer.

The challenge lies in accurately predicting the final cost early in the claims process using historical data and other relevant features, which can involve sophisticated modeling techniques to account for uncertainties and variability in claims patterns.

3.2 Data Descriptions

The dataset used in this thesis originates from the Actuarial Loss Prediction Competition 2020/21 hosted by The Actuaries Institute of Australia, the Institute and Faculty of Actuaries, and the Singapore Actuarial Society. This competition aims to foster the growth of data analytics skills, particularly among actuaries, by tasking participants with predicting Workers Compensation claims using realistic synthetic data.

The dataset in this competition comprises 90,000 authentic synthetic worker compensation insurance policies involving an accident. Each entry contains demographic and worker-specific details, along with a textual description of the accident. In particular, the dataset is divided into two files: a training set and a test set. In this thesis, we only take the training set into consideration (from here on, when referring to the dataset, it will be understood as the training set file), since the test set does not have an available target column for evaluation.

The dataset contains 15 columns (14 features and 1 target column) with 54000 records. Features include claim number, date, description (a text feature), some demographic fields of workers and some related to their current jobs. The initial incurred claim cost is also provided in order to predict the ultimate one (table 3.1).

Data type	Number	Fields
Categorical	3	Gender, MaritalStatus, PartTimeFullTime
Numerical	8	InitialIncurredCalimsCost, UltimateIncurredClaimCost, HoursWorkedPerWeek, WeeklyWages, DaysWorkedPerWeek, Age, DependentChildren, DependentsOther
Text	1	ClaimDescription
Date time	2	DateTimeOfAccident, DateReported

Table 3.1: Columns in the Dataset (Exclude Claim ID)

3.3 Methodology

3.3.1 Data Preprocessing

Data Cleaning

Firstly we extracted the year, month, day, weekday and hour from datetime of the accident and the report for data analysis. An extra column called DaysReportDelay (Number from accident to report) was created by subtracting the day of the report and the accident. Some cleaning processes included removing redundant and non-contributing features from the dataset have been done. Missing/ unknown values from Gender, MaritalStatus have been filled with the mode of the variables. We also remove some outliers from the dataset.

Feature Extraction

The pipeline for Feature Extraction is as in Figure 3.1. We will use this dataset to feed the models in the next part.

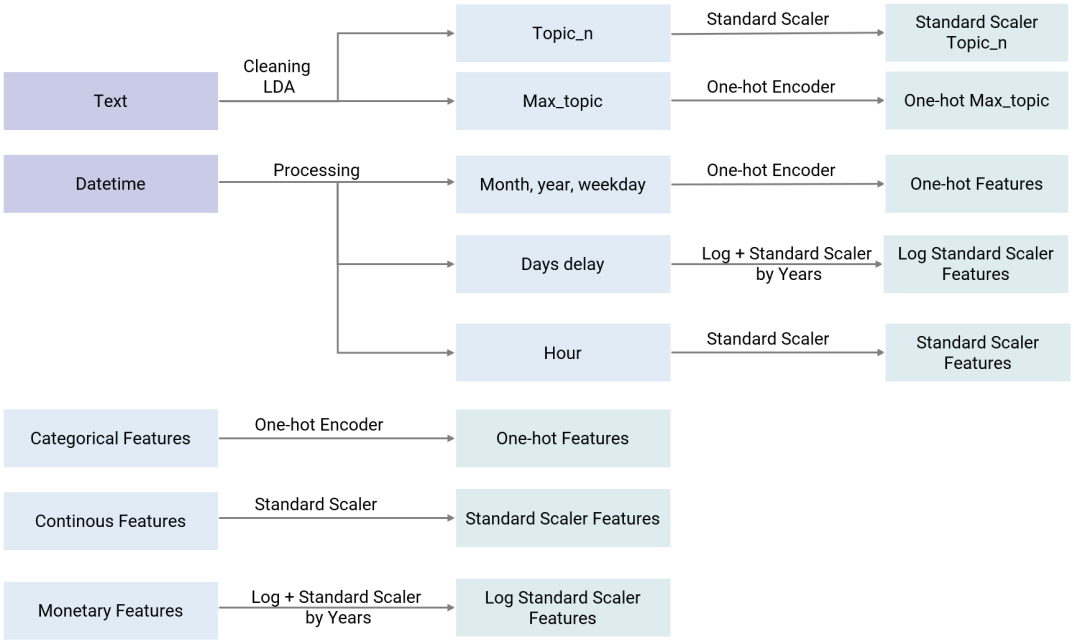

Figure 3.1: Pipeline for Feature Extraction

We consider some discrete fields such as DependentChildren and DependentsOther as categorical features due to their nature and distribution. Categorical features were

label-encoded for the purpose of representation, while we applied the standard scaler to the continuous ones. We grouped features related to money by years to avoid the impact of inflation, then used a standard scaler to standardize the log of continuous features. Transforming monetary features in data analysis is crucial to handle extreme values and account for changes over time.

The text feature - Claim Description was analyzed then removed stop words and applied a vectorizer. Some techniques used concluding *Latent Dirichlet Allocation* (LDA) to extract topics in textual corpora automatically. After that, we have n topics that are the most popular and k max_topic columns which are categorical variables. Each observation has the max_topic$_i$ value being one of the n founded topics from LDA.

The LDA model is a generative model class that allows defining a set of imaginary topics where each topic will be represented by a set of words. The goal of LDA is to map all documents to corresponding topics so that the words in each document represent those imaginary topics. It was introduced by Blei et al. (2001).

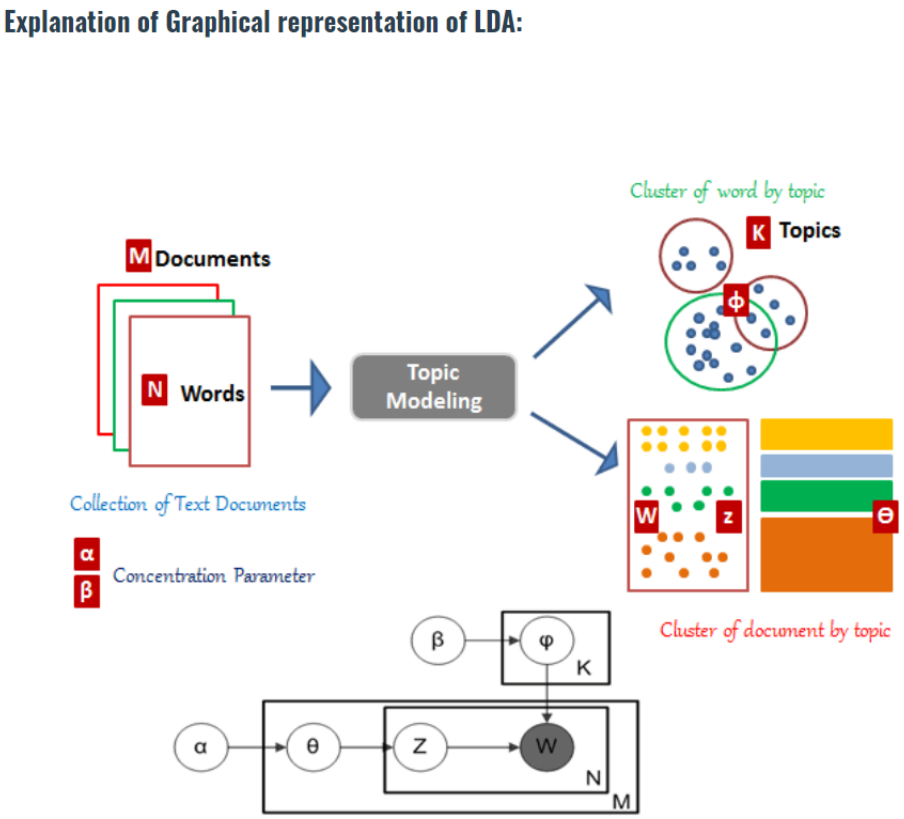

Figure 3.2: LDA Model
(Source: analyticsvidhya.com)

The model uses Dirichlet distributions to capture the distribution of topics within documents and the distribution of words within topics.

3.3.2 Modeling

In this thesis, we try three models to predict claim cost, including a traditional neural network (NN), Bayesian neural network (BNN) and generalized linear model (GLM). In the first step, we use a simple traditional neural network to predict the ultimate insurance claim cost. The model consists of 4 hidden dense layers followed by each dense layer being a dropout layer with a dropout rate of 0.5 and an output layer (figure 3.3).

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 155)	3720
dropout (Dropout)	(None, 155)	0
dense_1 (Dense)	(None, 155)	24180
dropout_1 (Dropout)	(None, 155)	0
dense_2 (Dense)	(None, 155)	24180
dropout_2 (Dropout)	(None, 155)	0
dense_3 (Dense)	(None, 310)	48360
dropout_3 (Dropout)	(None, 310)	0
dense_4 (Dense)	(None, 1)	311

Figure 3.3: Architecture of the Traditional Neural Network

The activation function used is ReLU and the optimizer is Nadam. Some techniques used in the model to avoid overfitting are Early stopping, which allows the model to stop training when a monitored metric has stopped improving and ReduceLROnPlateau to reduce learning rate when a metric has stopped improving.

Then to measure model uncertainty, we run a BNN. This model has the same architecture as the traditional version, but we replace Dense layers with DenseFlipout layers.

Finally, we use a GLM, a traditional model popular in the field of insurance cost prediction, to compare with BNN.

Generalized linear models are the flexible generalization of ordinary linear regression that allows for the dependent variable to have a non-normal distribution. GLMs are

used for modeling a wide variety of data types and are a fundamental tool in statistical analysis.

Generalized linear models allow the average to depend on the explanatory variable through a link function and the return value is an element of a set of distributions called Exponential Family (e.g., Normal, Poisson, Binomial). The model in the thesis uses the Tweedie family as an assumption. According to Nelder and Wedderburn (1972), each GLM has three components:

- Random component specifies the distribution of the dependent variable (Y).
- Systematic component (linear predictor) describes the explanatory variables in the model through their linear combination.
- Link function (η or $g(\mu)$) establishes the connection between the random and systematic components. It specifies how the expected value of the dependent variable is related to the linear predictors.

$$g(\mu_i) = \eta_i$$

Examples of link functions are identity, log, reciprocal, logit, and probit. We also assume that the transformed mean adheres to a linear model, expressed as

$$\eta_i = x_i^{'}\beta$$

Because the link function is one-to-one, it can be inverted to obtain:

$$\mu_i = g^{-1}(x_i^{'}\beta)$$

3.3.3 Evaluation

To compare point estimates across models, we use the RMSE and MAE metrics.

$$RMSE = \sqrt{\frac{1}{N}\sum_{1}^{N}(\text{Cost}_{true} - \text{Cost}_{pred})^2}$$

$$MAE = \sqrt{\frac{1}{N}\sum_{1}^{N}|\text{Cost}_{true} - \text{Cost}_{pred}|}$$

In addition, we select from the test set a sample of 10 observations, of which five observations are outliers and the other are typical for the dataset. Visualizing predicted results of observations helps make comparisons easier to understand.

Chapter 3 defines the topic that was concerned in the thesis, and presents the description of the dataset and the framework that we used to process data, train and evaluate models. The next chapter will emphasize some insights from the dataset and the results of the three models.

Chapter 4

RESULTS

4.1 Data Analysis

In this part, we conduct exploratory data analysis (EDA) on the dataset to uncover insights and comprehend the underlying patterns, trends, and data characteristics.

We divided the dataset into training (80%) and test set (20%). In 10800 observations of the testset, we choose a 10-observation sample to visualize later.

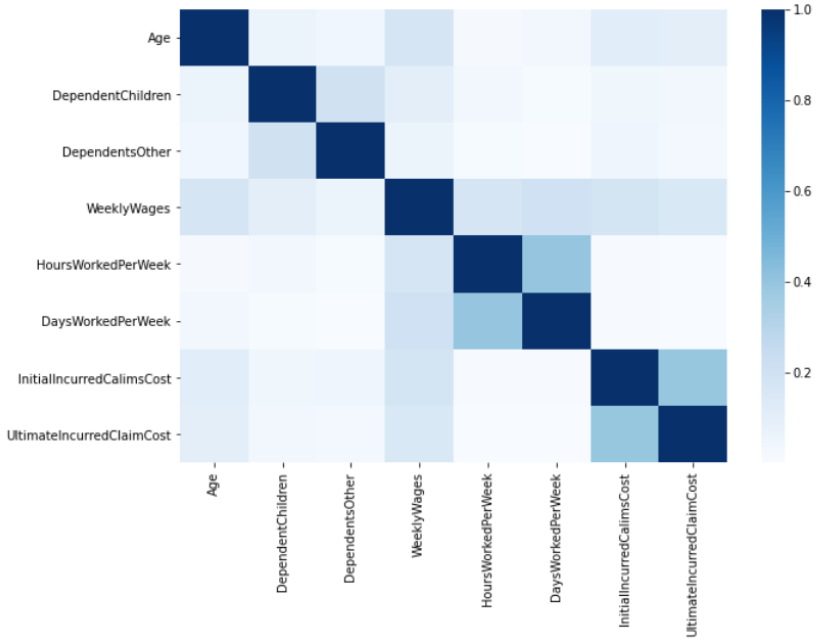

Figure 4.1: Correlation Heatmap along the Features

The heatmap presented is a correlation matrix showing the relationships between

various features in the dataset. Looking at the heatmap of the correlation, it can be seen that the two monetary fields have the highest correlation, about 0.4, suggesting that initial incurred claim costs are predictive of ultimate incurred claim costs.

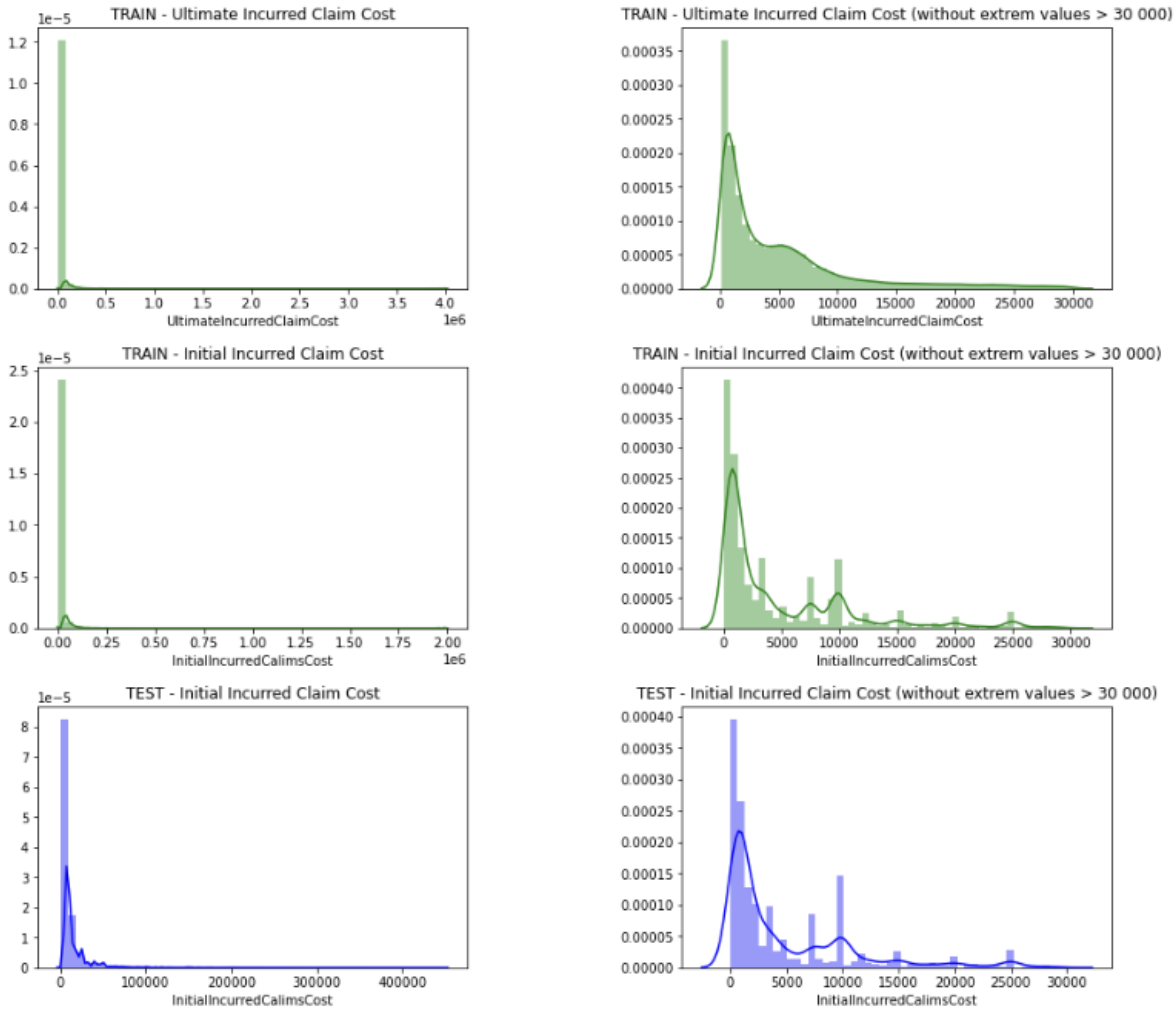

Figure 4.2: Distribution of Ultimate and Initial Claim Cost

The graphs provide a comparative analysis of the distribution of the Ultimate Incurred Claim Cost and Initial Incurred Claim Cost in both the training and test datasets, with and without extreme values (greater than 30,000). The histograms illustrate that both the ultimate and initial incurred claim costs are highly skewed to the right, with most values concentrated near zero and a few extremely high values stretching up to around 4 million.

Outliers

The insurance industry has very special cases where the initial estimate is very far from the actual amount paid. This dataset also contains such outliers. There are some cases that have the largest ultimate cost in the training set, more than 800,000, but the initial cost is less than 50,000. We will separate these cases from the training set and not include them in the fitting process.

Null values

The data in the dataset is quite complete. The only field having null values is Marital Status, with 23 null values in the training set and 6 in the test set.

Text: Claim Description

The bar charts display the most frequent words found in claim descriptions for both the training and test datasets. Both datasets share the same top words: "right", "left", "strain", "lower", "finger", "lifting", "hand", "struck", "shoulder", and "fell".

In both datasets, "right" and "left" are the most frequent words, indicating that these terms are commonly used to describe the side of the body affected in claims. Words related to body parts ("finger", "hand", "shoulder") and actions or conditions ("strain", "lifting", "struck", "fell") are prevalent, reflecting typical components of claim descriptions.

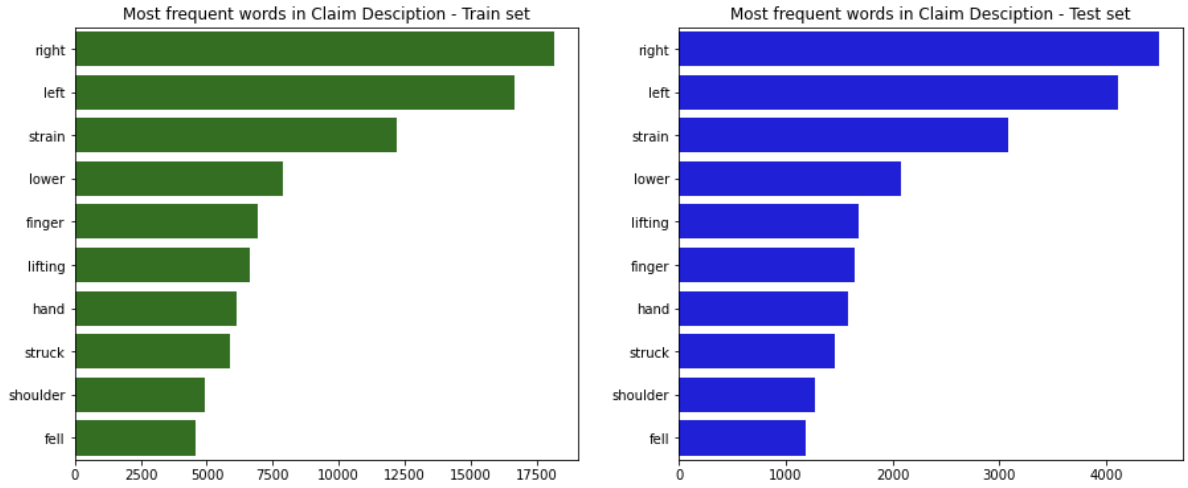

Figure 4.3: Frequency of the 10 Most Frequent Words

The frequencies are higher in the training set compared to the test set, which is due to the larger size of the training dataset. Despite the differences in absolute frequencies, the relative order of word frequencies remains consistent between the two sets, suggesting similar patterns in the descriptions of claims across both datasets.

Because it is life insurance, the top three words ("left", "right", "strain") are very frequent and do not affect the insurance amount. We add them to the list of English stopwords to remove when transforming the text.

Cleaning

This preprocessing step prepares the data for further analysis and modeling by:

- Extracting useful time-related features from date columns.
- Calculating delays and checking for same-year reporting.
- Clipping extreme values to handle outliers.
- Handling missing or inconsistent data values. Null values in Marital Status are replaced by "S" - single, which is more popular in the dataset. We also replace "Unknown" values of the feature Gender with "Men".
- Ensuring logical consistency for certain features: We mark invalid HoursWorkedPerWeek values that exceed the possible maximum (168 hours in a week).

Feature Engineering

Words in text feature are tokenized, converted to lowercase, and lemmatized. We remove stopwords and filter out words that occur in less than 10 documents, or more than 30% of the documents. LDA is used to extract topics from the claim description.

We fit a transformer including Log and Standard Scaler by years. The two box plots below compare the distribution of initial costs associated with accidents over different years.

The left plot shows the distribution of ultimate claim costs for accidents from 1988 to 2005, with each year represented by a separate box plot. The costs are capped at 100,000 to focus on smaller claims. The distribution is highly skewed, with numerous outliers and increasing median costs over the years, particularly after 2000. This indicates that raw monetary values are influenced by extreme outliers and possibly inflation.

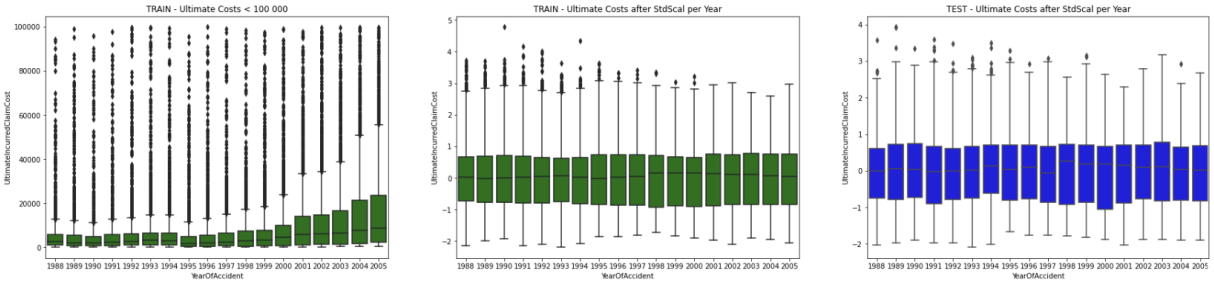

Figure 4.4: Ultimate Cost Boxplot Before and After Transformation

The right plot presents the same data after applying a log transformation and standardizing by year to account for inflation. The log transformation reduces the skewness of the data, compressing the range of values and mitigating the impact of outliers. Standardizing by year normalizes the values, allowing for consistent comparisons across different years. As a result, the distribution of claim costs is more uniform and stable over time.

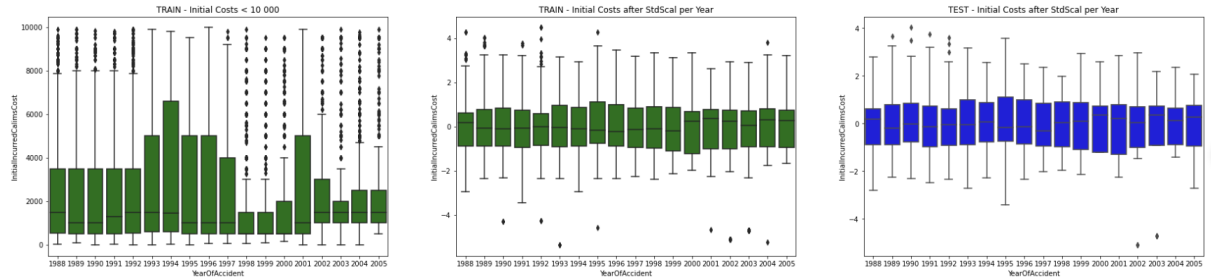

Figure 4.5: Initial Cost Boxplot Before and After Transformation

When applying the same transformation to feature initial cost, we have similar results. The standardization process applied to the training data generalizes well to the test data. Although outliers remain, their presence is reduced compared to the raw data. This consistency indicates the robustness of the transformation in preparing the data for modeling.

4.2 Results

4.2.1 Traditional Neural Networks

We use cross-validation with 5 folds to evaluate the validation set after training.

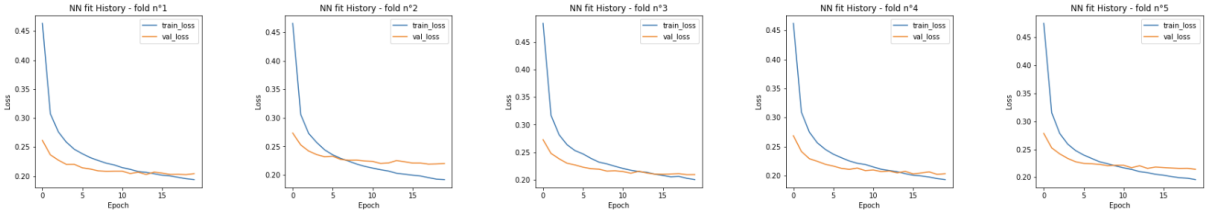

Figure 4.6: Loss of 5-fold Cross Validation - Traditional Neural Networks

It can be seen that the model has converged, and both the train loss and validation loss decrease after 20 epochs. The model is quite stable and can be evaluated on a test set. Mean of RMSE from 5 folds is 23,702. When using this model to predict in the test set, RMSE becomes 24,451.

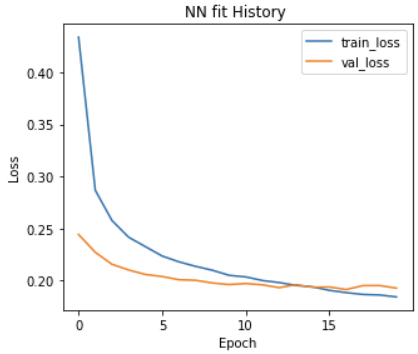

Figure 4.7: Traditional Neural Networks Loss

4.2.2 Bayesian Neural Networks

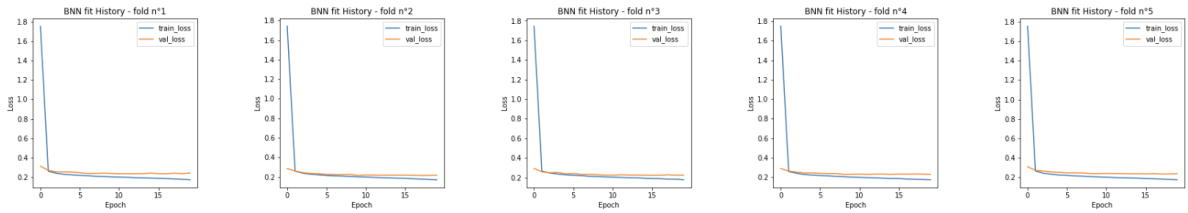

Figure 4.8: Loss of 5-fold Cross Validation - Bayesian Neural Networks

The graphs illustrate the training and validation loss histories for five different folds during the fitting process of a Bayesian Neural Network model. The rapid convergence and low, stable loss values for both training and validation datasets across multiple folds indicate that the model is effectively capturing the underlying patterns in the data while

maintaining good generalization capabilities. When applied to the test set, the model gets a similar result (Figure 4.9).

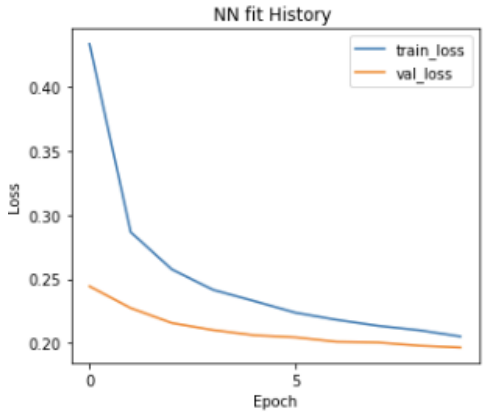

Figure 4.9: Bayesian Neural Networks Loss

4.2.3 Generalized Linear Models

GLM has a much lower running time than the neural network, but the results are quite good. This explains why traditional statistical models still have a certain place in the insurance industry although deep learning models are more and more powerful and popular.

Model	RMSE	MAE
Traditional Neural Network	24450.9	5663.0
Bayesian Neural Network	24215.0	5847.0
Generalized Linear Model	24493.6	5859.1

Table 4.1: Comparison among Three Models

4.2.4 Comparisons

In this step, we predict the value of cost 1000 times by sampling from BNN, then draw a boxplot representing the distribution of cost. The green point is the real ultimate cost, while the red point is the estimated prediction from the traditional neural network.

The model's prediction ability is quite good when almost all selected observations have prediction values within the range of the distribution. The model has shown its

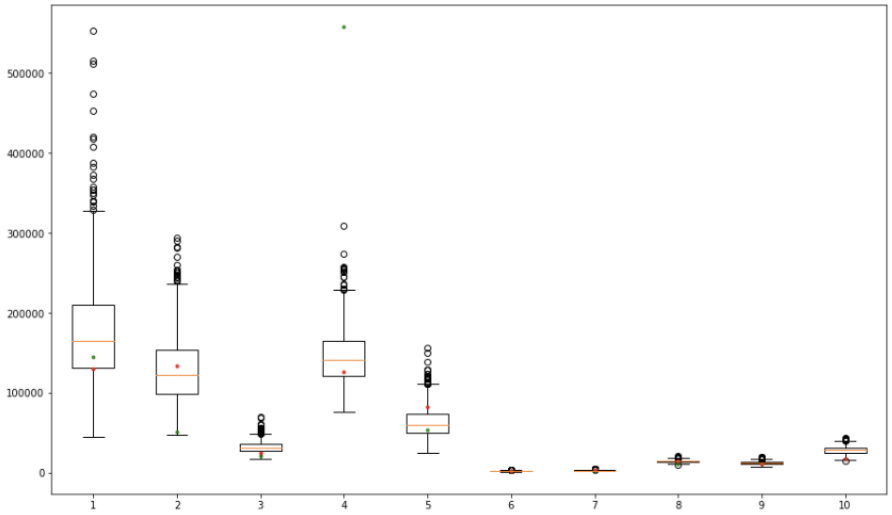

Figure 4.10: Boxplots for the Sample Result

ability to quantify uncertainty. Out-of-distribution observations have a much wider IQR than the remaining observations, with more outliers. The presence of many outliers in observations 1-5 indicates a skewed distribution with some extreme values. Meanwhile, normal inputs have a much smaller range. Observations 6-10 show more consistent data with fewer outliers and lower variance. There are very special cases where the range that BNN predicts cannot be covered (observation 4 in Figure 4.10), however, predicting a wide range proves that the model shows uncertainty with the results.

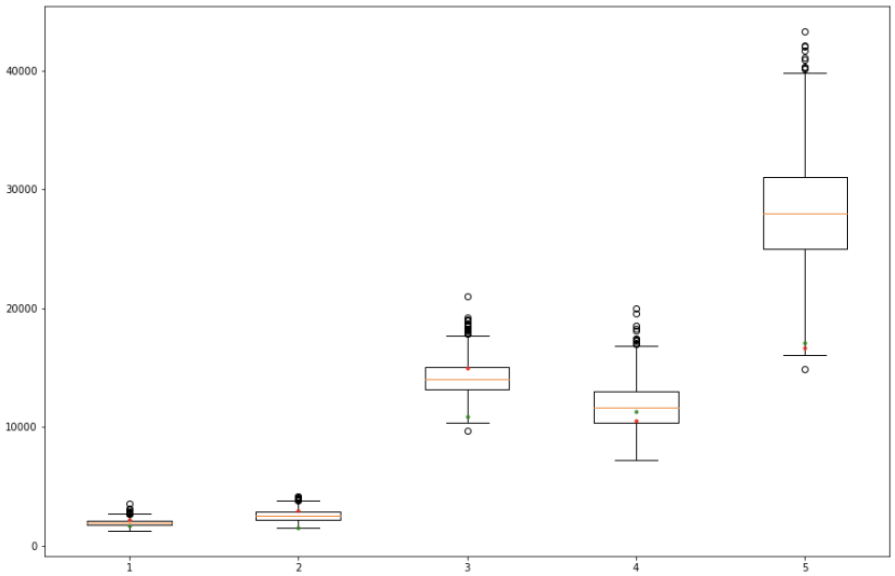

Figure 4.11: Boxplots for the Result of Observation Not Out-of-distribution

The bar chart in Figure 4.12, 4.13 compares the predictive performance of three models: Generalized Linear Model, Bayesian Neural Network, and Neural Network.

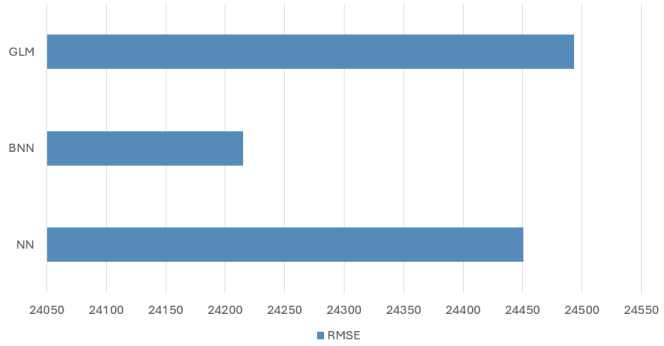

Figure 4.12: Comparing RMSE of Models

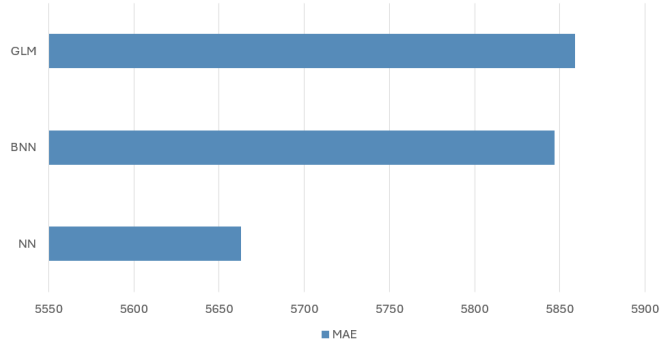

Figure 4.13: Comparing MAE of Models

When using RMSE, BNN has the lowest value since it may better penalize OODs or noisy test points. Traditional deep learning works well if using the MAE metric. The Generalized Linear Model has the highest RMSE and MAE. This indicates that it performs the worst in terms of the average squared error of its predictions and does not excel in minimizing the average magnitude of errors either. However, the difference is small and can be accepted if the point estimate is not preferred to the interpretability of the model.

The neural network and Bayesian neural network require a much longer time to converge for training. The GLM has the fastest training time among the three models as GLMs are generally less complex and require fewer computational resources, making them very efficient to train (Figure 4.14). It is an advantage of traditional statistical models.

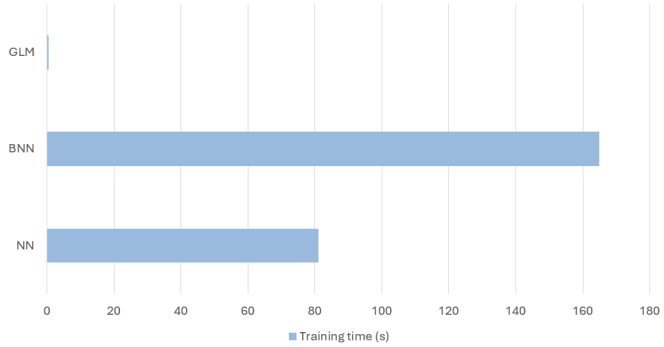

Figure 4.14: Comparing Training Time of Models

Chapter 4 has covered the data analysis and results of each model. It also compares three models: Traditional neural network, BNN and GLM. The better performance belongs to the traditional neural network and BNN, however, BNN can intimate the uncertainty in the model. The result that GLM brings is not as good as the others, but it can be acceptable in case we prefer the lower training time.

The next chapter will conclude all the results and contributions of this thesis, and give some recommendations for future works.

Chapter 5

CONCLUSION AND FUTURE RESEARCH

Conclusion

This thesis has explored the application of Bayesian Neural Networks in measuring uncertainty and their specific use in predicting insurance claim costs.

Through an in-depth review of theoretical foundations, we have established a comprehensive understanding of BNNs, highlighting their unique capability to incorporate uncertainty in predictions through Bayesian inference. The literature review underscored the significance of accurate uncertainty quantification in various fields and set the stage for practical application in the insurance industry.

The implementation of BNNs provided significant insights into their performance relative to traditional neural networks and another statistic model, GLMs. By designing and training BNN models, we demonstrated that BNNs not only deliver accurate predictions but also offer robust uncertainty estimates. These estimates are crucial for decision-making processes where understanding the confidence of predictions can substantially mitigate risks.

Applying BNNs to the prediction of insurance claim costs revealed the practical benefits of this approach. The BNN model, tailored for insurance data, successfully incorporated domain-specific features and provided meaningful predictions along with quantifiable uncertainties. This dual capability enhances the reliability of the model, making it a valuable tool for insurance companies in risk assessment, pricing strategies,

and reserve setting.

In evaluating the impact of data quality and quantity, the thesis highlighted the importance of robust data handling techniques. We proposed methods to address common data challenges, such as missing values, outliers, and inflation problems in monetary variables, ensuring that the BNN model maintained high performance and reliable uncertainty estimates even in less-than-ideal data conditions.

Limitations and Future Research

As for future work, it would be beneficial to tune the model in order to have better performance. Besides, experiments with other approaches to uncertainty quantification are also worth implementing. For example, since the distribution of features and targets in the claim cost problem is complex, conformal prediction or epistemic neural networks, which are the models that do not consider prior distribution, may bring better performance.

Additionally, this thesis does not include the analysis of feature importance. Since neural networks seem like "black boxes", it will be essential to know which features most affect the target field. It can help make predictions more optimal, as well as time-saving. This is an important factor in the finance and insurance industry, where the results should be interpretable.

In conclusion, this thesis has demonstrated that Bayesian Neural Networks are a powerful tool for measuring uncertainty and predicting insurance claim costs. By integrating BNNs into the insurance industry, companies can benefit from more accurate and reliable predictions, ultimately leading to improved risk management and financial stability. Future research should continue to refine these models and explore new applications, ensuring that the full potential of BNNs is realized.

REFERENCES

Abdulkadir, U., & Fernando, A. (2024). A Deep Learning Model for Insurance Claims Predictions [Publisher: Tech Science Press]. *Journal on Artificial Intelligence*, *6*, 71–83. https://doi.org/10.32604/jai.2024.045332

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. https://doi.org/10.1186/s40537-021-00444-8

Angelopoulos, A. N., & Bates, S. (2022). A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification [arXiv:2107.07511 [cs, math, stat]]. https://doi.org/10.48550/arXiv.2107.07511

Back, A., & Keith, W. (2019). Bayesian Neural Networks for Financial Asset Forecasting.

Bjarnadottir, S., Li, Y., & Stewart, M. G. (2019). Chapter Nine - Climate Adaptation for Housing in Hurricane Regions. In E. Bastidas-Arteaga & M. G. Stewar (Eds.), *Climate Adaptation Engineering* (pp. 271–299). Butterworth-Heinemann. https://doi.org/10.1016/B978-0-12-816782-3.00009-7

Blei, D., Ng, A., & Jordan, M. (2001). Latent Dirichlet Allocation. *Advances in Neural Information Processing Systems*, *14*. Retrieved May 26, 2024, from https://papers.nips.cc/paper_files/paper/2001/hash/296472c9542ad4d4788d543508116cbc-Abstract.html

Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight Uncertainty in Neural Networks [arXiv:1505.05424 [cs, stat]]. https://doi.org/10.48550/arXiv.1505.05424

Chandra, R., & He, Y. (2021). Bayesian neural networks for stock price forecasting before and during COVID-19 pandemic [Publisher: Public Library of Science]. *PLOS ONE*, *16*(7), e0253217. https://doi.org/10.1371/journal.pone.0253217

Duerr, O., Sick, B., & Murina, E. (2020). *Probabilistic Deep Learning: With Python, Keras and TensorFlow Probability*. Simon; Schuster.

Gal, Y. (2016). Uncertainty in Deep Learning. Retrieved May 3, 2024, from https://www.semanticscholar.org/paper/Uncertainty-in-Deep-Learning-Gal/3c623c08329e129e784a5d03

Goundar, S., Prakash, S., Sadal, P., & Bhardwaj, A. (2020). Health Insurance Claim Prediction Using Artificial Neural Networks. *International Journal of System Dynamics Applications*, *9*, 40–57. https://doi.org/10.4018/IJSDA.2020070103

Hauzenberger, N., Huber, F., Klieber, K., & Marcellino, M. (2023). Enhanced Bayesian Neural Networks for Macroeconomics and Finance [arXiv:2211.04752 [econ, stat] version: 3]. Retrieved March 5, 2024, from http://arxiv.org/abs/2211.04752

Hershey, J. R., & Olsen, P. A. (2007). Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models [ISSN: 2379-190X]. *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, *4*, IV–317–IV–320. https://doi.org/10.1109/ICASSP.2007.366913

Jospin, L. V., Buntine, W., Boussaid, F., Laga, H., & Bennamoun, M. (2022). Hands-on Bayesian Neural Networks – a Tutorial for Deep Learning Users [arXiv:2007.06823 [cs, stat]]. *IEEE Computational Intelligence Magazine*, *17*(2), 29–48. https://doi.org/10.1109/MCI.2022.3155327

Kafková, S., & Krivankova, L. (2014). Generalized Linear Models in Vehicle Insurance. *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*, *62*, 383–388. https://doi.org/10.11118/actaun201462020383

Kendall, A., & Gal, Y. (2017). What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *Advances in Neural Information Processing Systems*, *30*. Retrieved May 26, 2024, from https://proceedings.neurips.cc/paper/2017/hash/2650d6089a6d640c5e85b2b88265dc2b-Abstract.html

Khairnar, P., Thiagarajan, P., & Ghosh, S. (2020). *A modified Bayesian Convolutional Neural Network for Breast Histopathology Image Classification and Uncertainty Quantification*. https://doi.org/10.31224/osf.io/5xf8c

Kingma, D. P., & Welling, M. (2022). Auto-Encoding Variational Bayes [arXiv:1312.6114 [cs, stat]]. https://doi.org/10.48550/arXiv.1312.6114

Kiureghian, A. D., & Ditlevsen, O. (2009). Aleatory or epistemic? Does it matter? *Structural Safety*, *31*(2), 105–112. https://doi.org/10.1016/j.strusafe.2008.06.020

MacKay, D. J. C. (1992). A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation*, *4*(3), 448–472. https://doi.org/10.1162/neco.1992.4.3.448

McAllister, R. T., Gal, Y., Kendall, A., van der Wilk, M., Shah, A., Cipolla, R., & Weller, A. (2017). *Concrete Problems for Autonomous Vehicle Safety: Advantages of Bayesian Deep Learning* [ISSN: 1045-0823]. International Joint Conferences on Artificial Intelligence, Inc. Retrieved May 23, 2024, from https://www.repository.cam.ac.uk/handle/1810/266683

Neal, R. M. (1996). Monte Carlo Implementation. In R. M. Neal (Ed.), *Bayesian Learning for Neural Networks* (pp. 55–98). Springer. https://doi.org/10.1007/978-1-4612-0745-0_3

Nelder, J. A., & Wedderburn, R. W. M. (1972). Generalized Linear Models [Publisher: [Royal Statistical Society, Wiley]]. *Journal of the Royal Statistical Society. Series A (General)*, *135*(3), 370–384. https://doi.org/10.2307/2344614

Olsgärde, N. (2021). *Deep Bayesian Neural Networks for Prediction of Insurance Premiums* (Doctoral dissertation). Retrieved May 21, 2024, from https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-304675

Osband, I., Wen, Z., Asghari, S. M., Dwaracherla, V., Ibrahimi, M., Lu, X., & Van Roy, B. (2023). Epistemic Neural Networks [arXiv:2107.08924 [cs, stat] version: 8]. https://doi.org/10.48550/arXiv.2107.08924

Quijano Xacur, O. A., & Garrido, J. (2015). Generalised linear models for aggregate claims: To Tweedie or not? *European Actuarial Journal*, *5*(1), 181–202. https://doi.org/10.1007/s13385-015-0108-5

Sahai, R., Al-Ataby, A., Assi, S., Jayabalan, M., Liatsis, P., Loy, C., Hamid, A., Al-Sudani, S., Alamran, M., & Kolivand, H. (2023). Insurance Risk Prediction Using Machine Learning. https://doi.org/10.1007/978-981-99-0741-0_30

Verdoja, F., & Kyrki, V. (2021). Notes on the Behavior of MC Dropout [arXiv:2008.02627 [cs, stat]]. https://doi.org/10.48550/arXiv.2008.02627

Wen, Y., Vicol, P., Ba, J., Tran, D., & Grosse, R. (2018). Flipout: Efficient Pseudo-Independent Weight Perturbations on Mini-Batches [arXiv:1803.04386 [cs, stat]]. https://doi.org/10.48550/arXiv.1803.04386

Wuthrich, M. V. (2016). Machine Learning in Individual Claims Reserving. https://doi.org/10.2139/ssrn.2867897

Zhu, L., & Laptev, N. (2017). Deep and Confident Prediction for Time Series at Uber [arXiv:1709.01907 [stat]]. *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, 103–110. https://doi.org/10.1109/ICDMW.2017.19

APPENDIX

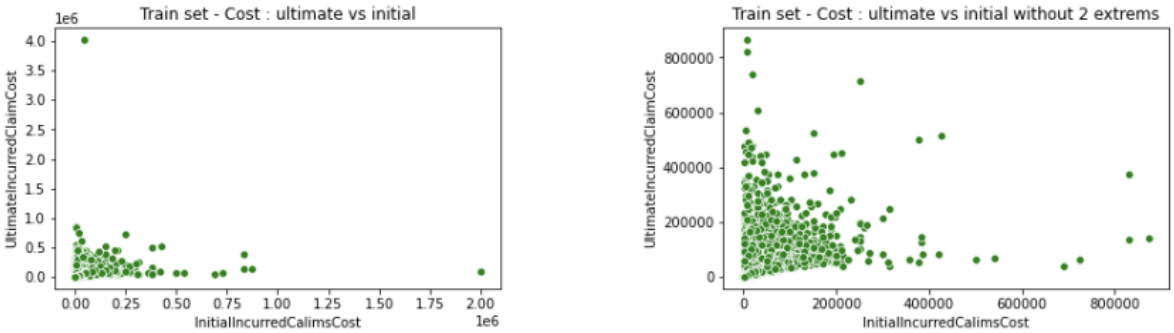

Figure .1: Scatter plot of Initial and Ultimate Cost

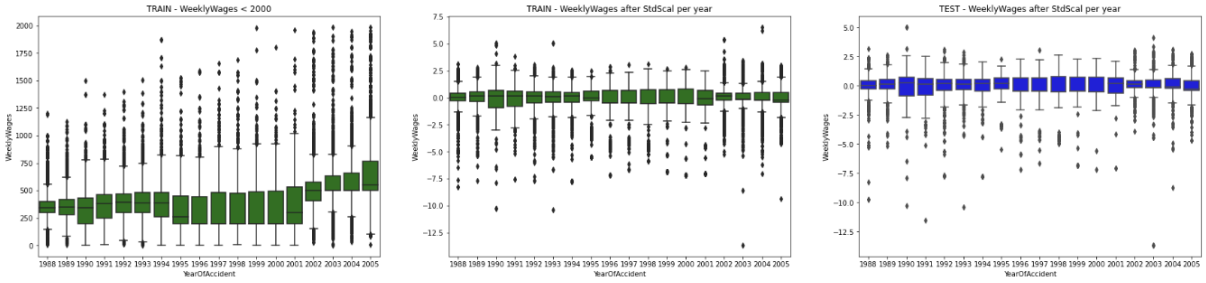

Figure .2: Weekly Wages Boxplot Before and After Transformation

MyNEUThesis.pdf

ORIGINALITY REPORT

5%	5%	1%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.diva-portal.org Internet Source	3%
2	cran.rstudio.com Internet Source	1%
3	ufdcimages.uflib.ufl.edu Internet Source	1%

Exclude quotes	Off	Exclude matches	< 1%
Exclude bibliography	On		