Uncertainty Analysis in Earthquake Prediction using Deep Learning Methods for Improved Risk Management

MSc Research Project
Data Analytics

Aleksandra Kalisz
Student ID: x21118876

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Aleksandra Kalisz
Student ID:	x21118876
Programme:	Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Dr. Catherine Mulwa
Submission Due Date:	12/08/2024
Project Title:	Uncertainty Analysis in Earthquake Prediction using Deep Learning Methods for Improved Risk Management
Word Count:	6672
Page Count:	25

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Aleksandra Kalisz
Date:	15th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Uncertainty Analysis in Earthquake Prediction using Deep Learning Methods for Improved Risk Management

Aleksandra Kalisz
x21118876

Abstract

Predicting earthquake magnitude remains a challenging aspect of natural disaster management. This study explores deep learning techniques to improve earthquake predictions, with a focus on uncertainty in seismic forecasting. The experiment includes Bayesian Long Short-Term Memory (LSTM), Bayesian Convolutional Neural Networks (CNN), Bayesian Temporal Convolutional Networks (TCN), and a Hybrid Bayesian CNN/LSTM model integrated with the Monte Carlo Dropout Method to enhance the reliability of the predictions by effectively quantifying uncertainty. All models underwent a training process of 1000 epochs. The Adam optimizer and Stochastic Variational Inference (SVI) were used to adjust parameters and control uncertainty, improving the learning process. The models were evaluated using various metrics such as Standard Deviation, Uncertainty Estimate, MAE, RMSE and R2 to assess their accuracy and uncertainty. The results indicated that the Bayesian LSTM model was the most effective, delivering the precise forecast while maintaining well-calculated uncertainty, with the lowest Mean Absolute Error (MAE) of 0.0322 and Root Mean Squared Error (RMSE) of 0.0453 and an R-squared score of 0.9903 indicating that it accounted for nearly all the variability in data. Also, the hybrid Bayesian CNN/LSTM model performed well, showing a good balance between accuracy and uncertainty. This research highlights the importance of deep learning methods and their potential in helping to manage the risk of natural disasters, saving lives, and reducing economic losses caused by earthquakes.

1 Introduction

Every year, thousands of people are affected by earthquakes, tsunamis, and landslides caused by the Earth's crust movement. These natural disasters can strike without warning causing significant damage to buildings, roads, bridges, and landscapes as well as it can lead to economic losses and displacement of hundreds of people. The danger and unpredictability of these events make it very challenging for emergency services to plan the response effectively and to allocate resources prioritising the most affected areas. Despite advances in seismology, there is continuous uncertainty around predicting the time, location, and magnitude of earthquakes. However, in recent years, the advancement of technology and especially deep learning methods showed promise in improving forecasting abilities. Deep learning models are able to analyse large amounts of data while using historical data and transfer learning to see the patterns which traditional approaches would miss. This research aims to explore the application of deep learning approaches in earthquake magnitude prediction, focusing on its uncertainty to anticipate seismic events and save lives and to prevent economic losses.

1.1 Background and Motivation

Earthquakes can trigger several catastrophic effects including the development of tsunamis where massive waves are created by sudden displacement of water can flood coastal areas, resulting in loss of life and damage to infrastructure. Moreover, tectonic plate movement can cause a volcanic eruption and cause dangerous gas and magma to come to the surface, necessitating the quick evacuation of the affected area. Other dangers induced by earthquakes are landslides, collapse of structures, fires, and gas explosions which further increase the destruction of populations and ecosystems.

Traditional methods for seismic forecasting and risk management depend on historical data and physical models, which are valuable but can have significant limits regarding data availability, speed, and precision McGuire (1995). In recent years, deep learning techniques have become an attractive choice for further developments in seismology to minimise the impact of earthquakes on people's lives and infrastructure, as well as help in disaster management. Deep learning as a type of AI achieved great results in several sectors, including natural language processing and image identificationLeCun et al. (2015). However, the application of deep learning, particularly for regulating and minimising uncertainty, is still an underexplored area as the unpredictability of earthquakes makes them uniquely difficult to study.

This study aims to address the gap in the field of uncertainty analysis for earthquake magnitude prediction using Bayesian Neural Networks and Monte Carlo Dropout Method applied to four deep learning models: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Temporal Convolutional Network (TCN) and Hybrid CNN/LSTM Model, which are particularly effective in earthquake magnitude predictions due to their ability to handle complex, multidimensional data and capture complex patterns. The model's performances and metrics will be analysed and compared to determine their effectiveness and suitability for different risk management objectives.

1.2 Research Questions

RQ: To what extent do deep learning techniques effectively quantify and reduce uncertainty in earthquake magnitude prediction models, thus improving risk management and preparation for disasters?

SQ 1: What advantages and limitations arise when using deep learning methods for predicting earthquake magnitudes, and how can we address these issues?

SQ 2: What is the best method to address uncertainty in earthquake strength prediction?

1.3 Research Objectives and Contributions

The primary objective of this research is to improve risk management strategies by developing, evaluating and comparing deep learning models that provide good prediction accuracy and reduce uncertainty. This experiment will identify the model's advantages and weaknesses as well as proposed methodologies to increase its effectiveness in real-world scenarios, as well as address additional objectives such as evaluating the computational efficiency of the model, analysing uncertainty calculations, and identifying key predictive features.

The structure of this report will follow: Section 2: Review of the relevant literature on earthquake prediction using deep learning techniques to identify gaps in current knowledge. Section 3: Methodology and the research procedure. Section 4: Design Specification. Section 5: Implementation and evaluation discussing findings, advantages, and limitations of the models. Section 6: Conclusion of the technical report and feature work.

No	Objectives	Description
1	Improve Risk Management Strategies	Develop, evaluate, and compare deep learning models.
2	Evaluate Model Performance and Accuracy	Assess and compare the performance of deep learning models.
3	Analyse Uncertainty Quantification Techniques	Examine and implement techniques for quantifying uncertainty in deep learning models.
4	Assess Computational Efficiency and Scalability	Analyse the computational resources needed for training and deploying the models.

Table 1: Objectives

2 Literature Review on Uncertainty Analysis in Seismology

This section is dedicated to Uncertainty analysis and machine learning approaches in earthquake predictions. It follows the evolution of earthquake prediction methods from first probabilistic methods to machine learning integration and finally deep learning models. Over the years, the dynamic advancement of machine learning has drastically transformed the vision of earthquake magnitude assessment and risk management associated with it. Traditionally, earthquake predictions heavily relied on expert judgements and statistical models, often resulting in limitations in accuracy and reliability. However, the introduction of new and advanced computational techniques as well as the arrival of machine learning methods shifted these approaches toward more data-driven ones.

2.1 Evolution of Uncertainty Analysis Using Deep Learning

The evolution of deep learning (Figure 1) of methods for uncertainty calculation in earthquake prediction has progressed from classical statistical methods to more complex deep learning techniques. The Bayesian Neural Networks (BNNs) were introduced by Neal (2012). They created a foundation which incorporates the Bayesian concept into Neuron Network models, their method treated weights as distributions to provide probabilistic predictions. Also, Breiman (1996)demonstrated the Ensemble Method on Bagging predictors, which showed the effectiveness of using multiple models to capture uncertainty. The arrival of the Monte Carlo Dropout Method described in the work of

Gal and Ghahramani (2016) transformed the field by providing a practical method to estimate model uncertainty. Afterwards, improvements included the Deep Gaussian Process (DGPs) presented in the work of Damianou and Lawrence (2013), which integrated the flexibility of deep learning and Gaussian Process capabilities. In 2015 Blundell et al. (2015) further improved uncertainty estimation in Neural networks, introducing Bayesian Deep Learning (BDL). The latest developments like Evidential Deep Learning by Sensoy et al. (2018) and Uncertainty-aware Neural Networks of Kendall and Gal (2017)provide a strong framework for implementing aleatoric and epistemic uncertainties throughout deep learning models, improving seismic events analysis and dependability.

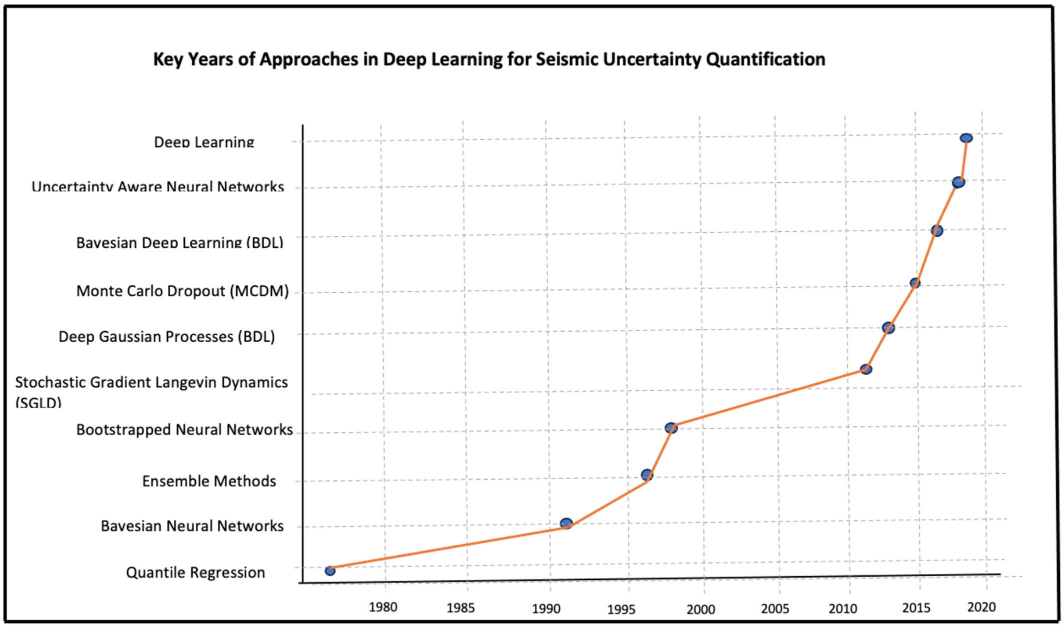

Figure 1: Evolution of Uncertainty Analysis Graph.

2.2 An Investigation of Bayesian Neural Networks and Monte Carlo Dropout Method.

In 1996 Cramer et al. (1996) used the approach of the Monte Carlo Dropout Method to estimate uncertainties in seismic hazards for three counties in California, using different parameters' recurrence rate, ground motion attenuation relations, and magnitude frequency distribution. The study showed 95% confidence in uncertainty in shaking level. This study demonstrated the importance of parameter variability when estimating seismic hazards in highly populated areas. Also, the work of Baker and Cornell (2008) focused on studying how different uncertainties affect predictions of damage caused by earthquakes. Researchers introduced PEER framework, suggesting models for correlations where there were no empirical data, as well as including Monte Carlo simulation to more restricting closed-form solutions. The method was computationally intensive and did not produce desired results. The study focused on the uncertainty of the whole model was conducted by Bradley (2009), in performance-based assessment using Monte Carlo simulations demonstrating how uncertainties can influence seismic hazard estimates. The main advantage of this approach is its ability to include a wide range of epistemic uncertainties in hazard models. The method is reliable on seismic assessment but resilient to extensive data. In 2022 Wang et al. (2022) expanded the field of applying Bayesian Convolutional Networks (BCNN) to predict seismic responses to address the uncertainty in seismic events. Their BCNN model achieved NMSE for acceleration: 0.00192% NMSE for displacement: 0.00505%, standard deviation for acceleration: 0.0117%, and Standard deviation for displacement: 0.0510%. The model performed very well with very low NMSE values suggesting high accuracy. Authors came across many limitations of this method like, challenges in maintaining efficiency handling large amounts of high-quality data, as the model depends on it. The model was difficult to interpret, and it can be difficult to identify biases in it. Lastly, implementing this model to real-time scenarios can be challenging due to the time required for model training and interference. Gal and Ghahramani (2016) presented the idea of estimating uncertainty in neural networks by using dropout as a Bayesian approximation. Their method generates different network configurations, which capture epistemic uncertainty. This method which takes uncertainty into account has been used wildly in image classification but also has been adapted to time series data and is often used in earthquake magnitude predictions by capturing inherent uncertainties. However, the authors found this method computationally intensive, as it needs multiple forward passes to estimate uncertainty. Temporal Convolutional Networks (TCN) are particularly suitable for modelling sequential data, such as seismic waveforms. Mousavi and Beroza (2019) used Bayesian TCN with Monte Carlo Dropout to address inherent uncertainties in earthquake prediction. Their research showed that despite the extensive computational resources, Bayesian technique significantly improved the accuracy and reliability of the seismic forecast. Uncertainties of probabilistic seismic hazards were examined by Farhadi and Mousavi (2016) by treating the parameters as random variables rather than fixed values. This time the Monte Carlo simulation was applied to hazard values like levels of risk of potential earthquake. The method proved significant variations of outcomes in applying the method to "hazard" values, compared to traditional methods like fault behaviours and earthquake magnitude. The research found that depending on how the seismic waves decrease, the strength of the hazard estimates can differ. While this method can lead to realistic outcomes, the introduction of multiple uncertainties can lead to many outcomes, and it can be difficult to communicate it to the stakeholders. Also, this process requires continuous validation against new information, especially in real-time situations. Bergen et al. (2019) also presented the use of the Bayesian CNN for processing seismic images, which captured uncertainties in earthquake-prone regions. Their method improved hazard assessment but faced challenges in scaling with large datasets. Overall, in earthquake forecasting using time series data, Bayesian LSTM showed big potential.

Further developing the Bayesian approach, a significant work by Wang and Takada (2009) presented a Bayesian updating framework for predicting seismic ground motion, which improves accuracy by continuous integration and merging data specific to the site. Their method involved linear and non-linear regression models and adjusting the current attenuation (reduction) formula which resulted in the improved performance with metrics: Mean Squared Error (MSE) of 0.049, Root Mean Squared Error (RMSE) of 0.221, and Mean Absolute Error (MAE) of 0.167. This approach improved local predictions but relied greatly on data quality and involved great computational complexity. In another study Moustra et al. (2011) investigated the application of Artificial Neural Networks for earthquake prediction in Greece, they achieved 80.55% accuracy using time series magnitude data, but only 58.02% for events bigger than 5.5 magnitude. When Seismic Electric Signals (SES) were used, the accuracy improved significantly to 84.01% for magnitude prediction and 92.96% for both magnitude and time delay prediction. Yagi et al. (2014) used Bayesian modelling to analyse the Iquique earthquake sequence, which occurred off the coast of Chile. Research provided accrued fault parameters estimation, like the main shock slide, with 95% credibility. The model also provided strong probabilistic predictions for aftershocks with 60 – 70% accuracy and proved to be effective in improving seismic hazard assessment thanks to the detailed fault parameters. The authors came across the problems with data dependency and discovered that this model requires large amounts of high-quality data, additionally, the model easily overfits due to complex and noisy datasets.

In conclusion, the studies on Bayesian Neural Networks and the Monte Carlo Dropout Method in seismic prediction on uncertainty show its high effectiveness in the improvement of uncertainty estimation. Bayesian approaches demonstrate continuous increases in performance through the integration and incorporation of prior data into new information, making it an adaptable method suitable for a variety of seismic assessments. Also, the Monte Carlo Dropout Method enhances prediction models, effectively improving models' reliability across different seismic scenarios, making it a key instrument in accurate risk assessment. While the methods significantly update accuracy in uncertainty, often they require extensive computational resources and high-quality data.

3 Scientific Methodology and Data Pre-Processing

3.1 Introduction

Based on the literature review and research objectives, this chapter explains the methodology used to measure uncertainty in earthquake strength prediction using deep learning models for improved risk management. The experiment will use data which is publicly available, and the link is specified in the data collection section. To answer the research questions, a process flow diagram is established as shown in Figure 2. The process includes several steps, including data collection, data pre-processing and transformation, model architecture, model training procedure and evaluation metrics.

3.2 Research Procedure and Methodology

The project methodology integrated the Cross-Industry Standard Process for Data Mining (CRISP-DM). The method incorporates the business element which is necessary to understand the project objectives and provides a structured framework for planning, executing, and managing the project.

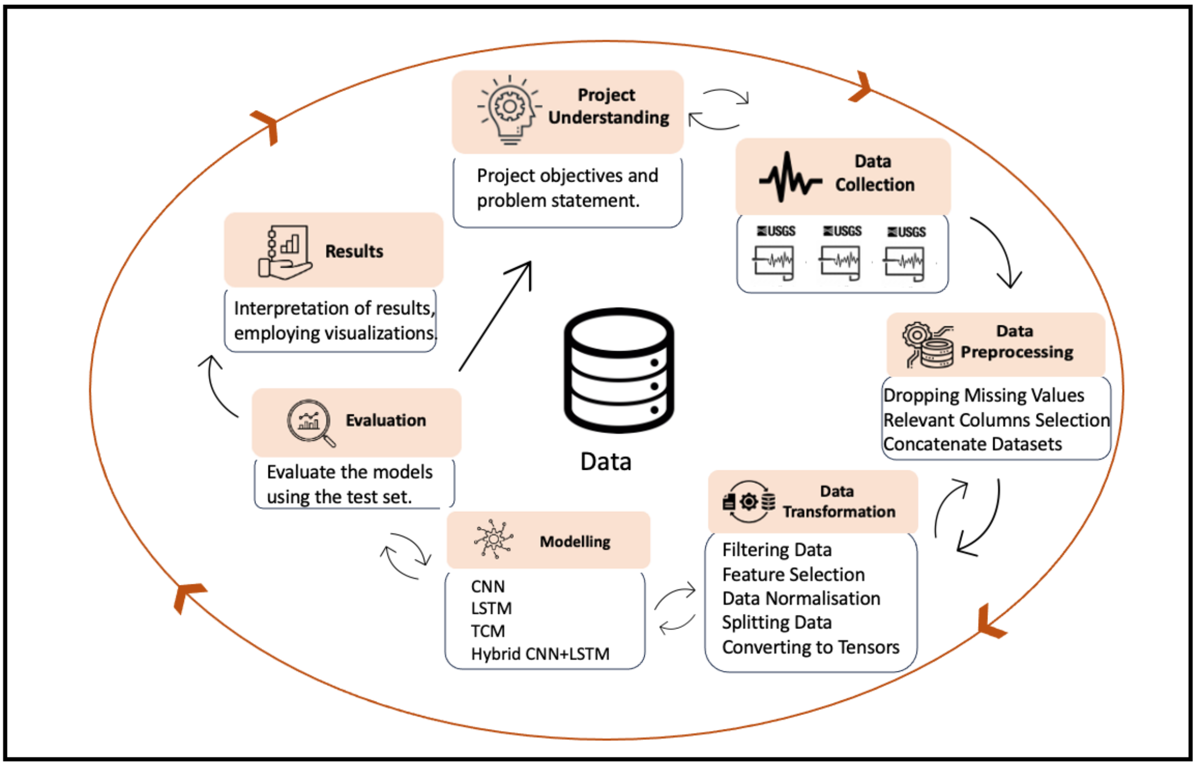

Figure 2: Earthquake Uncertainty Analysis Methodology.

4 Design Specification

This research project uses two-tier architecture to analyse earthquake predictions and its uncertainty. The essential component of the project implementation is shown in Figure 3. Tier 1 of the design consists of the stakeholder including the emergency respond team, aid workers, seismologists, and emergency services, who will be provided with the outcome of the experiment from the previous layer. The business analytics layer is a part of the Tier 2, where the data is collected from The United States Geological Survey Website and pre-processed using Python. Data is split and passed to be trained on the models. Finally, the models are tested and evaluated.

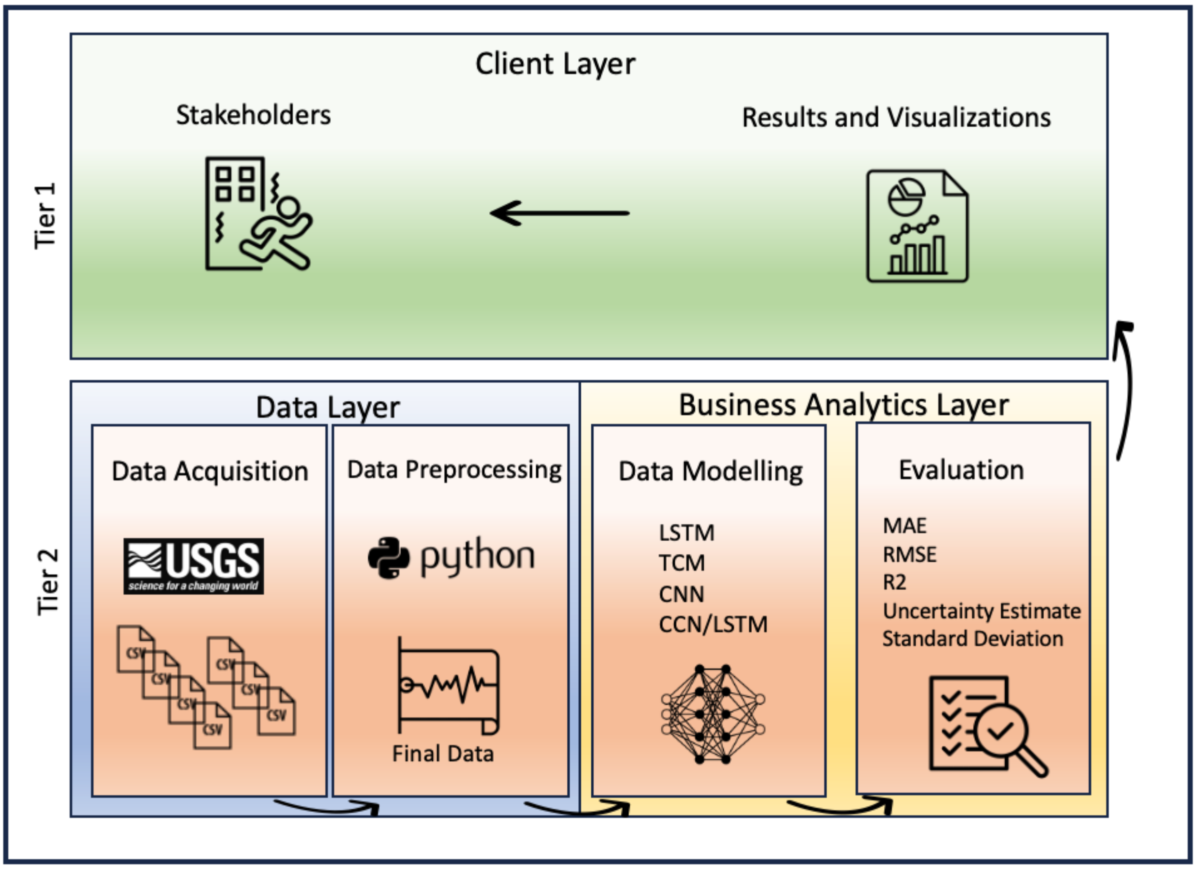

Figure 3: Design Specification.

5 Implementation, Evaluation, and Results of Uncertainty Analysis using Deep Learning

5.1 Introduction

In this section, the models used for analysing uncertainty will be described. The detailed description of collecting and pre-processing data will be presented, as well as model implementation, evaluation, and results. Different models will be compared, based on performance metrics, to determine the most effective model for predicting seismic motion.

5.2 Equipment and Tools

The study uses several tools and software for the development of the models and analysis:

Software: Python, using Jupyter Notebook and Microsoft Excel to visually examine CSV files with data. PyTorch and Pyro libraries in Python were used for model implementation, NumPy for numerical operations and Pandas for data manipulation, Matplotlib for data visualisation, scikit-learn for statistical analysis and evaluation metrics. Hardware: In this study, a machine with minimum 2.6 GHz Dual-Core Intel Core i5 or higher must be used to handle computational demands.

5.3 Data Collection and Pre-processing

5.3.1 Data Collection

The data for this experiment is sourced from the United States Geological Survey Website and includes 102,228 observations collected over 20 years. The downloaded historical data is in CSV format and focuses on the Alaskan region, known for its high seismic activity and complex tectonic environment. Alaska is situated on the North American tectonic plate but is influenced by the Pacific tectonic plate, which folds under it along the southern coast, leading to significant tectonic and volcanic activity. According to Lanzano et al. (2016), it is essential to focus on data from one region as it guarantees consistency in seismic and geological characteristics and it can improve the model's accuracy. Since seismic events vary between locations, a model trained on inconsistent data might not function well. It is preferable to use local data, which can unlock patterns and tendencies of the entire specific region. Data was stored on a local drive as CSV files.

5.3.2 Data Preprocessing

The first step in data pre-processing was to filter out all earthquakes below magnitude 4 on the Richter scale, as events below that threshold are not significant enough to be felt by people or lead to infrastructure damage.

Real-world data is frequently noisy, inconsistent, and incomplete, this project required a number of steps to prepare the data before training the models to ensure reliability, and accuracy in the later phase. That included choosing only relevant columns, which are: Latitude — a measure of north-south position of an earthquake epicentre. Longitude — a measure of east-west position of the earthquake epicentre. Geospatial coordinates are selected to provide precise location of the earthquake and how close it was to the tectonic plate border and fault lines. Depth — is a variable to indicate how deep below the earth surface the earthquake occurred. The effect on the surface and ground shaking and how seismic wave travels is determined by depth; the shallower the depth, the more ground shaking occurs, and seismic waves don't travel far. Magnitude — is a variable which measures the energy generated during an earthquake. This variable is measured, and described, in the Richter scale. It is a critical measure to assess the potential impact of the earthquake on people and infrastructure.

Another crucial step in data pre-processing is dropping missing values to handle incomplete datasets. This approach helps to ensure that predictions made by deep learning models are reliable and accurate. However, it is important to take into a count the effects of this approach, as significant reduction in data can decrease predictive ability. While this is an effective method, it should be used carefully and with consideration of data characteristics.

https://earthquake.usgs.gov/earthquakes/search/

5.3.3 Data Transformation

One of the most important parts of data transformation is data normalisation, as this step ensures that no particular characteristics dominates in the study because of its dimensions. Data normalisation is essential for many reasons. Firstly, it places features with diverse measures in equal position and the algorithm can handle them evenly. Secondly, it allows quicker progress to the best solution, as it enhances performance of machine learning. Thirdly, by not allowing data features with greater ranges to influence the learning process, it allows each characteristic to contribute equally to the final model and reduces bias. Lastly, normalised data lead to better and more reliable model performance by reducing variability and ensuring that the model performs effectively when applied to unseen data. This performance is highlighted in work by García et al. (2015) where authors offer a broad insight into these methods of data pre-processing. Ultimately, the final dataset was created of a total of 6850 observations to train, validate and test deep learning models (Figure 4).

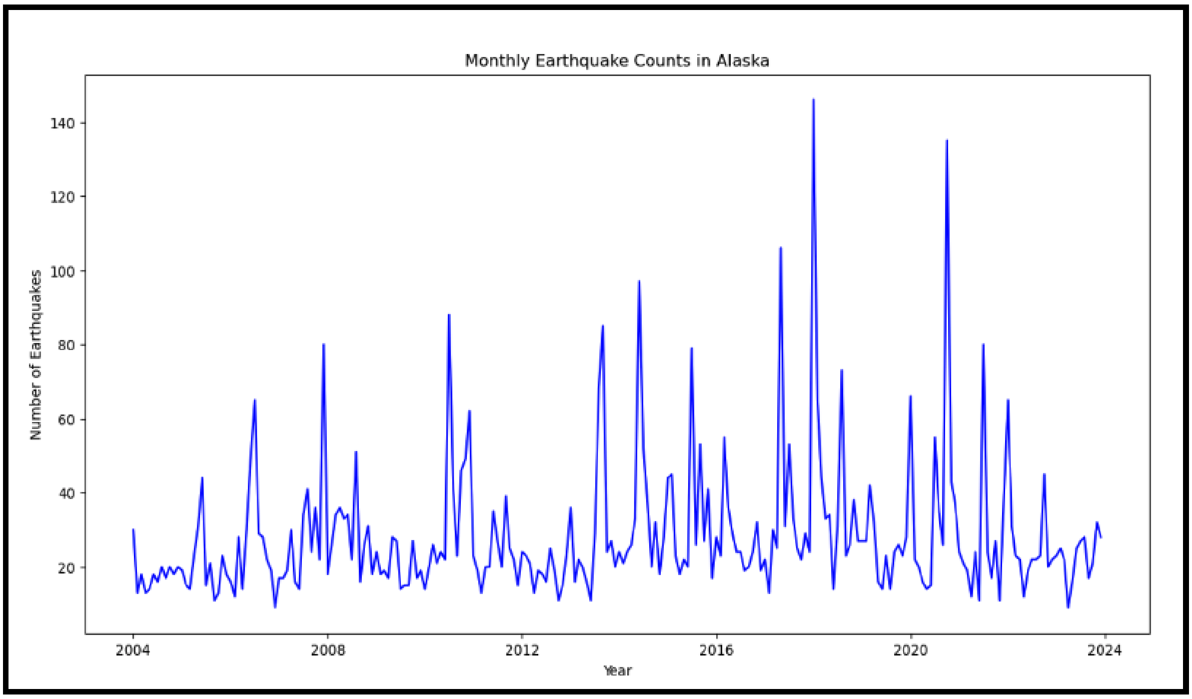

Figure 4: Earthquake Frequency Histogram.

```
Total number of data entries before pre-processing: 102282
Total entries in training set: 4110
Total entries in validation set: 1370
Total entries in testing set: 1370
```

Figure 5: Number of Observations after Pre-processing.

After completing data pre-processing, the data was divided into three subsets for training, validation, and test in (Fig 5) using 60/20/20 technique as this approach helps to create reliable and effective models. The training set is composed of 4110 entries. In the training part, the model learns patterns and correlations between the target variable (magnitude) and input variables (latitude, longitude, depth), assigning magnitude to it.

The next 20% of the data which is 1370 entries is used for validation. This practice allows adjusting its settings and stops the model from memorising the training set. It increases its capacity to handle new data that has never been seen by assessing how well it performs on validation during training. This way of splitting data not only ensures that the model gets enough data to learn efficiently, but also keeps enough, sufficient data to validate and test its results. Once the models been trained and validated, is tested using last 20% of the data (1370 entries). This step gives an unbiased view of how well the model performs and predict the earthquake magnitude using new data.

5.4 Model Development and Training

The approach taken for developing and training earthquake prediction models included developing an array of specific models designed to detect and predict the strength of seismic activity and uncertainty of its predictions. Four primary models were chosen: Long Short-Term Memory (LSTM) network, Convolutional Neural Networks (CNN), the Temporal Convolutional Networks (TCN), as well as Hybrid model of CNN/LSTM. All the models were selected for their strengths in handling sequential data and their ability to evaluate temporal dependencies. Each of these models will incorporate the Bayesian Method and the Monte Carlo Dropout Method for calculation of uncertainty.

5.4.1 Bayesian Long Short-Term Memory Networks with Monte Carlo Dropout Method

Long Short-Term memory networks are designed to address the challenges of traditional RNN, especially the vanishing gradient issue, which makes it difficult to learn long term dependencies in RNN models. LSTM networks are formed by interconnected memory cells, where each one of them contains three gates: the input gate, the forget gate, and the output gate. The purpose of these gate is to regulate the flow of information through the memory cells, allowing the network to support long term information update (Figure 6).

• The "forget gate "decides which details from the previous memory cell should be removed. This gate outputs a value between 0 and 1 for each number of the memory cell, meaning that the forget gate decides how much of the previous information should be discarded and how much should be kept. • The "input gate" decides which new details should be stored in the memory cell. This part involves two phases: first, a sigmoid layer decides which value to update and second, a 'tanh' layer creates an array of new potential values, which might be included in the memory cell.

At this stage, the old memory cell is updated to the new memory. This process is based on connecting together old and new candidates of memory cells, which are weighted by the output of the "forget' and "input" gates Yu et al. (2019).

• The "output gate" decides the output of the LSTM cell according to the updated memory cell. The memory cell is passed through 'tanh' layer and multiplied by the output of the sigmoid gate.

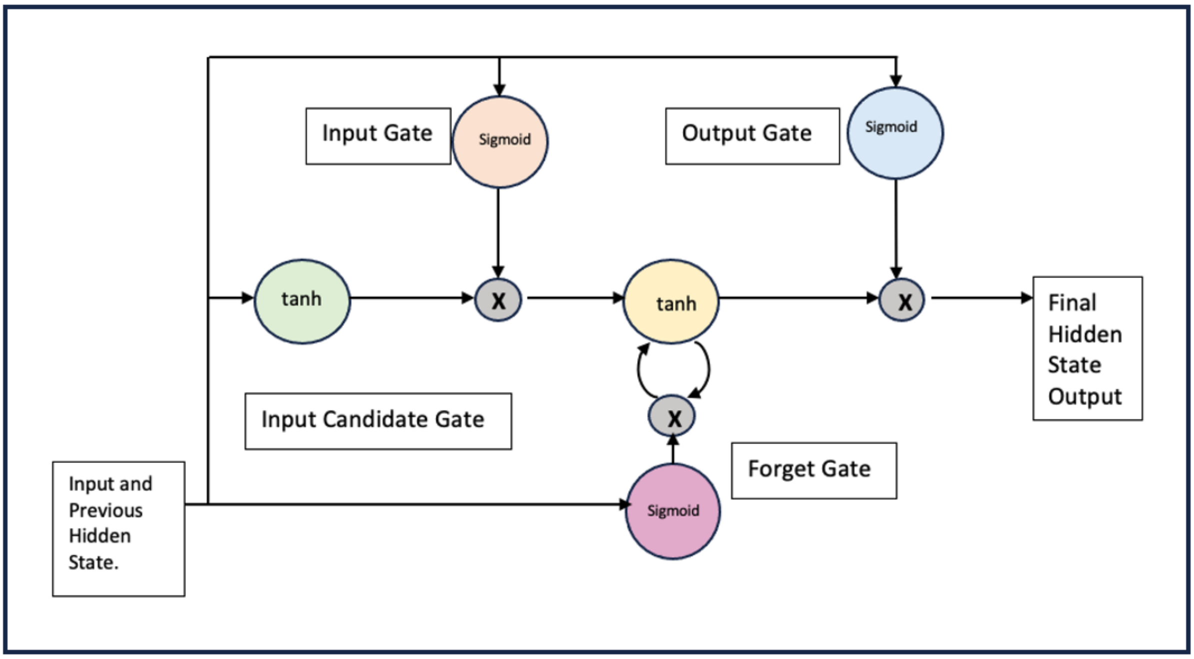

Figure 6: LSTM Model.

In the experiment, the Bayesian LSTM model was created using 'pnn.PyroModule' which configures the LSTM layer with weights and biases using probabilistic programming. This is acheived by using 'pynn.PyroSample' to add to these parameters. Also, the model includes a fully connected layer with similar probabilistic weights and biases, plus a dropout layer to prevent overfitting. The 'forward' definition passes input through the LSTM and applies a dropout and outputs to the predicted mean and observation noise represented by a sampled noise parameter. The 'train and evaluate' function configures the model using 'AutoNormal' guide and the Adam Optimizer inside the Stochastic Variational Inference (SVI) framework, and uses 'Trace_ELBOstop' to minimize loss during training. The model is trained over 1000 iterations, with training and validation losses recorded and plotted. The test set is used to develop predictions using Pyro's 'predictive' class, by calculating the mean and standard deviation of predictions. Also, the model performance was evaluated using Mean Absolute Error (MAE), R-squared Score (R^2), Root Mean Squared Error (RMSE), Uncertainty Estimate (Standard Error), and Standard Deviation.

Model Training: The Bayesian Long Short-Term Memory (LSTM) model with Monte Carlo Dropout was trained over 1000 epochs, demonstrating major improvements in loss metrics, and achieving high evaluation scores. The training process showed a consistent decline in both training and validation losses. Initially, the training loss started at 32218.44, with a validation loss of 24278.97 then dropped to 3619.52 at epoch 400. However, at Epoch 700 the training loss has decreased further to negative values of -588.86, indicating overfitting or potential data inconsistency. The trend continued and at Epoch 900, the training loss was -2031.54 and the validation loss slightly negative at -8.19.

Model Evaluation: Over the course of the 1000 iteration the model showed consistent improvement in performance with training loss decreasing and validation loss reduced indicating strong learning ability (Fig 7). However, the training loss became negative which can indicate slight possible overfitting, possibly indicating that the model picked up noise instead of patterns. Additionally, plotting predictions vs true magnitudes of the earthquake confirms the model's accuracy with the points of predicted and true values closely aligning with true values (Figure 8), which also could be due to overfitting. The evaluation metrics (Table 2) show that the model performed very well. The low Uncertainty Estimate and Standard deviation represent stable and dependable predictions and the Mean Absolute Error and Root Mean Squared Error are small, indicating high accuracy. The R2 Score shows that the model explains 99.03% of the variability in data.

Metrics	**Value**
Uncertainty Estimate (Standard Error)	0.1452
Standard Deviation	0.0453
Mean Absolute Error (MAE)	0.0322
R-squared Score (R^2)	0.9903
Root Mean Squared Error	0.0453

Table 2: Bayesian LSTM Performance Metrics

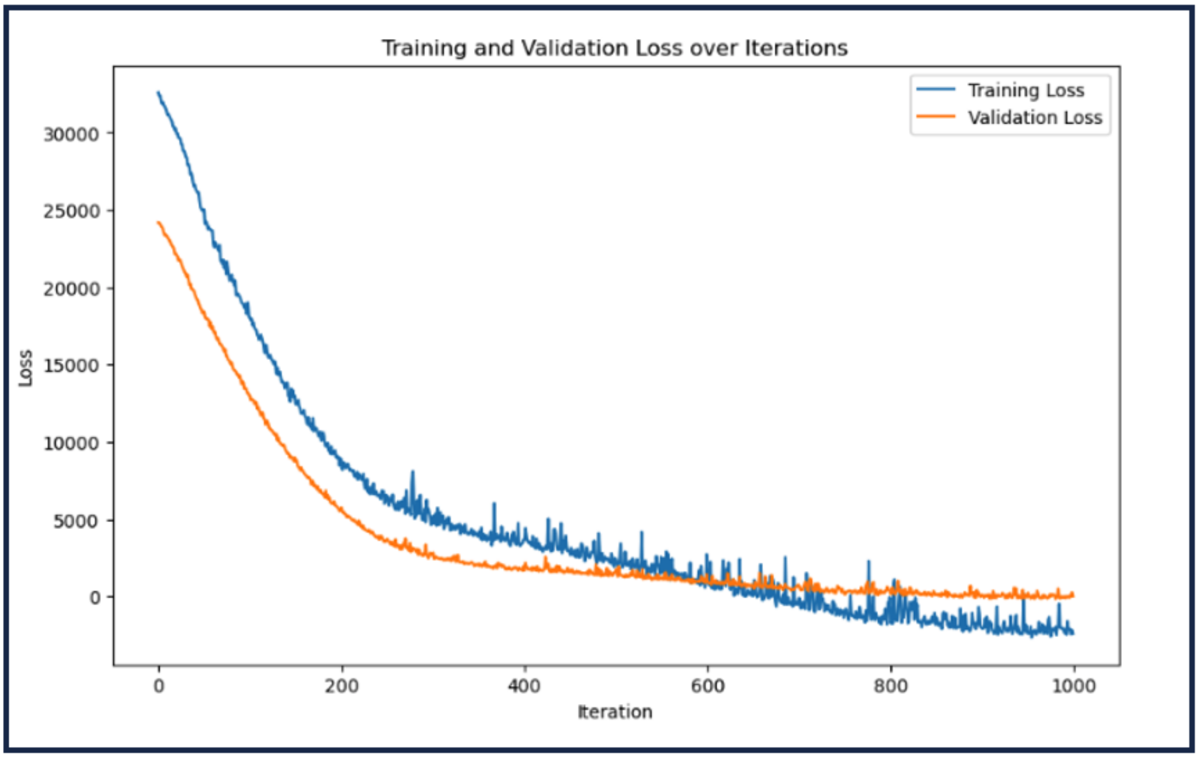

Figure 7: Bayesian LSTMT raining and Validation Loss over Iterations .

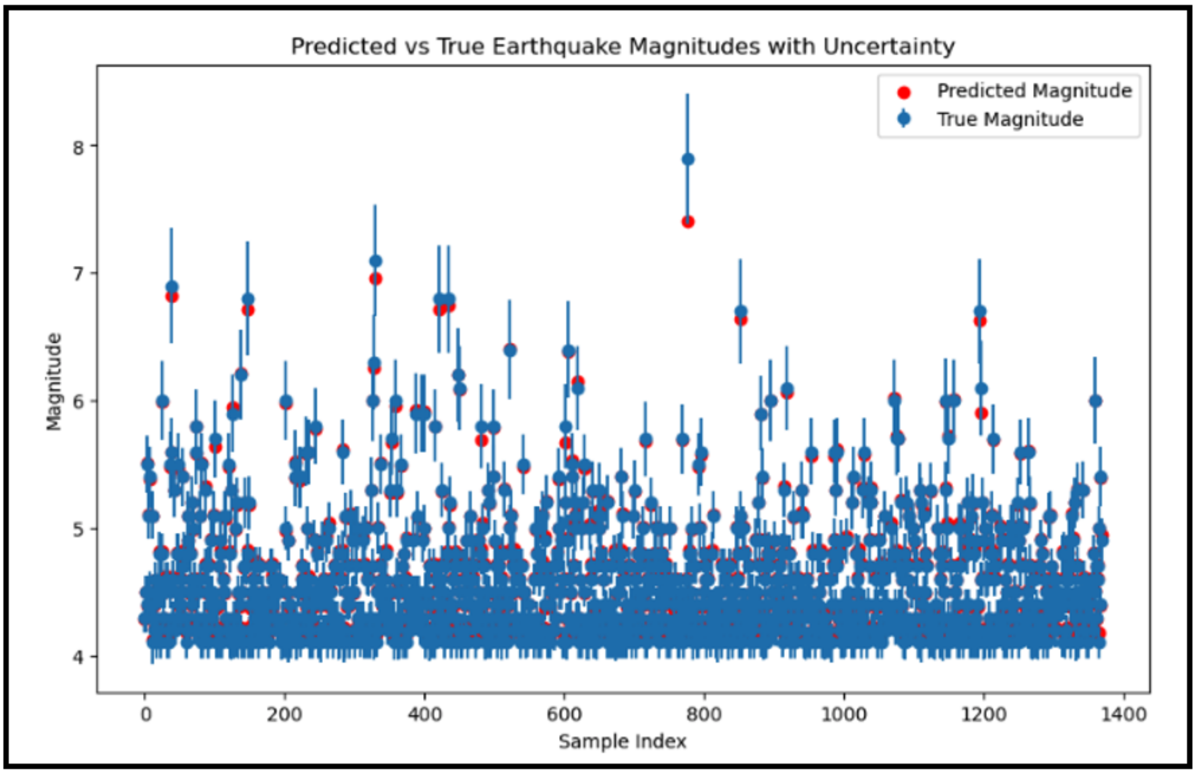

Figure 8: Bayesian LSTM Predicted vs True Magnitudes of the Earthquake.

5.4.2 Bayesian Convolutional Neural Networks (CNN) with Monte Carlo Dropout

Convolutional Neural Networks (CNNs) were designed to extract hierarchical patterns from spatial data like images, but it was successfully modified and adapted to time data series. The model captures complex patterns by applying convolutional filters that are very effective in detecting features across various positions and dimensions. CNN is using pooling and sharing of parameters to achieve translation consistency and minimise computational complexity. This allows users to manage big datasets and improve the accuracy in different predictions. CNNs are built of several essential elements, including, convolutional layer, pooling layers, and fully connected layers, each with a distinct function (Figure 9).

• Convolutional Layer are the fundamental components of CNNs. They apply a series of filters to the input data, transforming them and producing a feature map. In the time series data, each filter slides over the data and learns spatial features to identify patterns. This process helps CNN automatically detect significant trends and patterns. • Pooling layers decrees the computational load and help control overfitting. For the time series, data pooling operations are used to preserve the clearest features. • Fully Connected Layers are positioned at the end of the network. They create the final output, taking the filtered features from convolutional and pooling layers and transforming them into outputs of classification and regression results.

CNNs models for time series data are very effective thanks to their ability to learn complex temporal dependencies and spatial hierarchies without manual feature extraction Zhao et al. (2017).

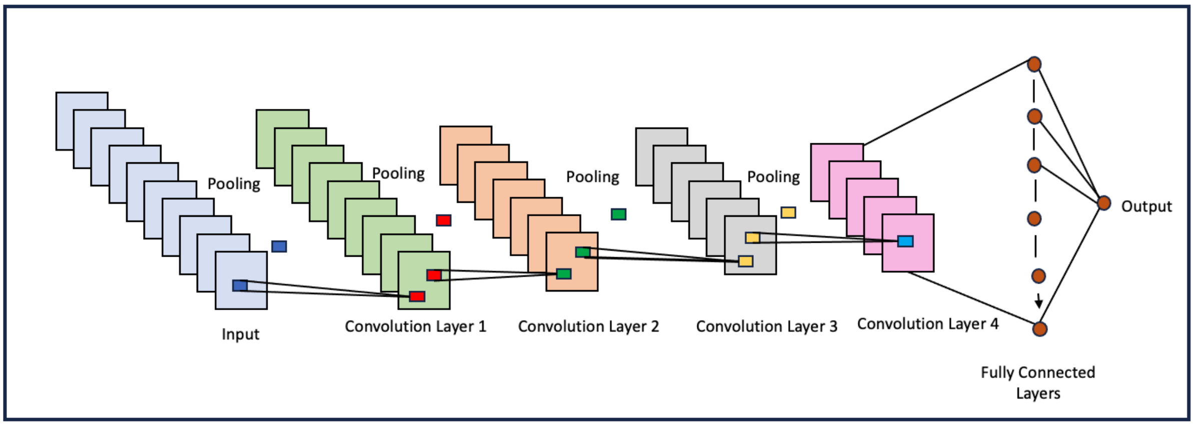

Figure 9: CNN Model.

In this project, a Bayesian CNN model was implemented using `pnn.PyroModule` by incorporating probabilistic programming. Firstly, the input data passes through a series of convolutional layers, each containing probabilistic weights and biases specified using `pnn.PyroSample`, which assigns a normal distribution to these dimensions. After each convolutional layer, a max pooling layer (`pool1` and `pool2`) is applied to decrease the spatial dimensions of feature maps. For model regularization and to allow uncertainty calculations during inference, a dropout is added after pooling layers to randomly deactivate neurons. The flattened output from the convolutional layers is passed through a fully connected layer (`fc`) with probabilistic weights, followed by a second dropout to ensure model stability. The final layer, outputting data (`fc_out`), generates the predicted mean of the data and the levels of noise across different data points, which are sampled by the parameter `sigma`. At the end, the Monte Carlo Dropout is employed using the `predictive` class with multiple samples to estimate the posterior distribution of the estimates. The model was trained and evaluated using the `train_and_evaluate` function with Stochastic Variational Inference (SVI). This method monitors the learning and validation process and is designed to combine stochastic optimization with variational inference to handle large datasets. The model's performance was evaluated using Mean Absolute Error (MAE), R-squared Score (R^2), and Root Mean Squared Error (RMSE).

Model Evaluation: The results of the model show that the Bayesian Convolutional Neural Network (CNNs) with the Monte Carlo Dropout was trained over 1000 epochs and shows reduction in training and validation losses, indicating some learning ability. (Figure 10). Starting at a high loss initially of 29543.11 and validation loss of 21454.61. By iteration 500 the training loss decreased further and reached 7 7354.12 with a validation loss of 4583. By iteration 900 the training loss further decreased to 4774.12, with validation loss of 2899.73 showing improvement. Despite reductions observed over iterations, the model's ability to perform on unseen data remains low, as shown in evaluation metrics (Table 3). The high MAE of 0.4008 and RMSE of 0.4910 indicate lack of precision in predictions. Furthermore, the negative R2 score means the model explains very little of variability, indicating poor fit. Also, this score might suggest that the model captures underlying patterns of the data. There are significant discrepancies between true and predicted magnitudes of the earthquake (Figure 11), meaning that the model is not predicting outcomes accurately. The low Uncertainty Estimate of 0.6334 and Standard Deviation of 0.0615 show stable but not accurate predictions.

Metric	Value
Uncertainty Estimate (Standard Error)	0.6334
Standard Deviation	0.0615
Mean Absolute Error	0.4008
Root Mean Squared Error	0.4910
R-squared Score	-0.1423

Table 3: Performance Metrics

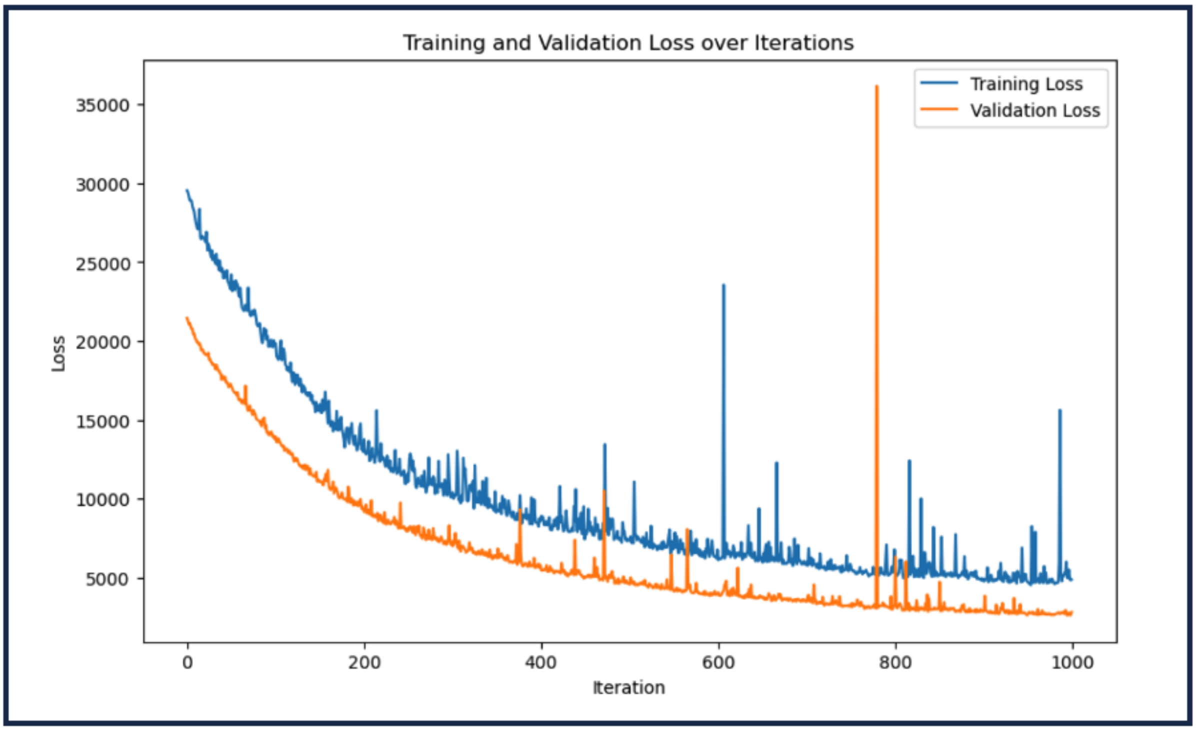

Figure 10: Bayesian CNN training and Validation Loss over Iterations.

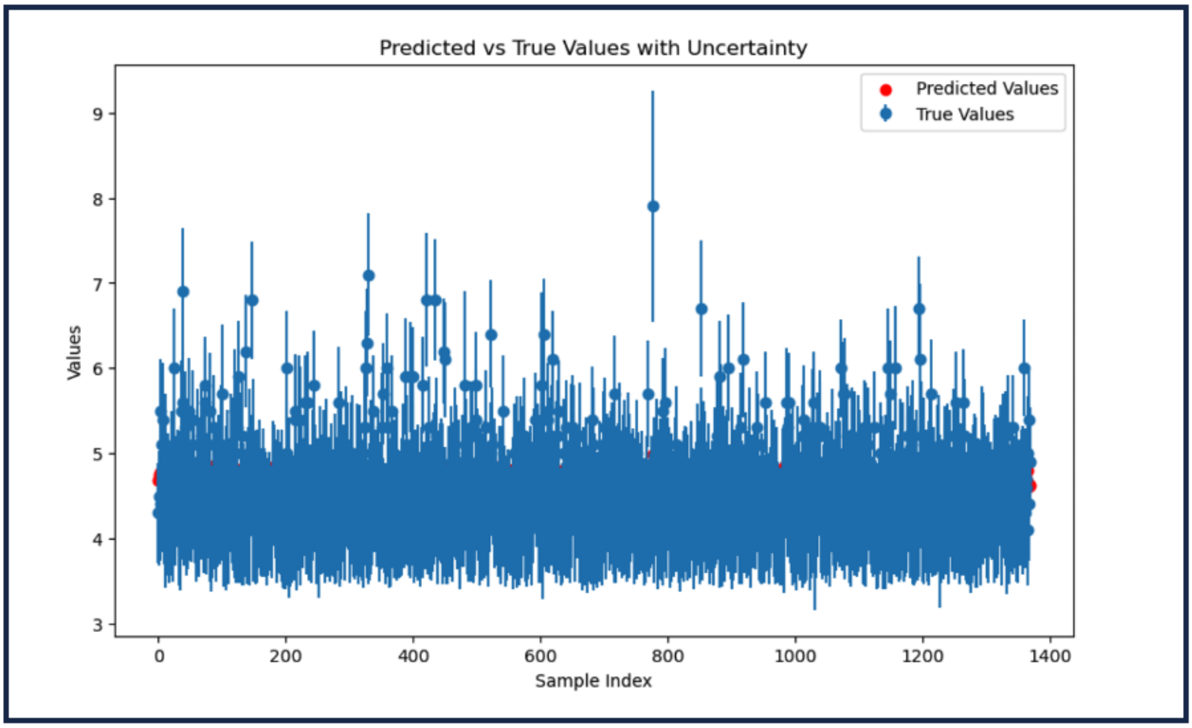

Figure 11: Bayesian CNN Predicted vs True Magnitudes of the Earthquake.

5.4.3 Bayesian Temporal Convolutional Networks (TCN)

Temporal Convolutional Networks are designed to handle sequence and time series data, and they succeed at time series forecasting as they can model long-term dependencies within data. The model (Figure 12) uses casual convolutions which guarantee the predictions at a specific moment depending on the information from the past and present, as well as dilated convolutions which allow the network to cover wider timescale with fewer layers. Bayesian method improves this design by adding the ability to estimate the uncertainty in predictions. This method enables the model to learn a distribution of possible weights instead of fixed weights, which means that for each weight in the network, there is a range of values that the model can choose from. Finally, this approach helps to regulate the model by reducing the risk of overfitting.

The models process starts with an input layer where the data is fed to the network. The data is processed by the convolutional layer which consist of casual convolutions to make sure that the predictions are based on past and present information. It uses dilated convolutions to capture long term dependencies using gaps in data points and helps to increase model's response Zargar (2021). The network then employs the Bayesian inference to learn the distribution of weights which can vary, and during predictions the model samples these weight multiple times, resulting in multiple predictions for each timestamp. During the inference and training process, the Monte Carlo Dropout is applied by performing multiple forward passes on the same input data with different dropout sequences. These predictions are gathered to produce the final predicted value and calculate uncertainty, which are displayed by the output layer.

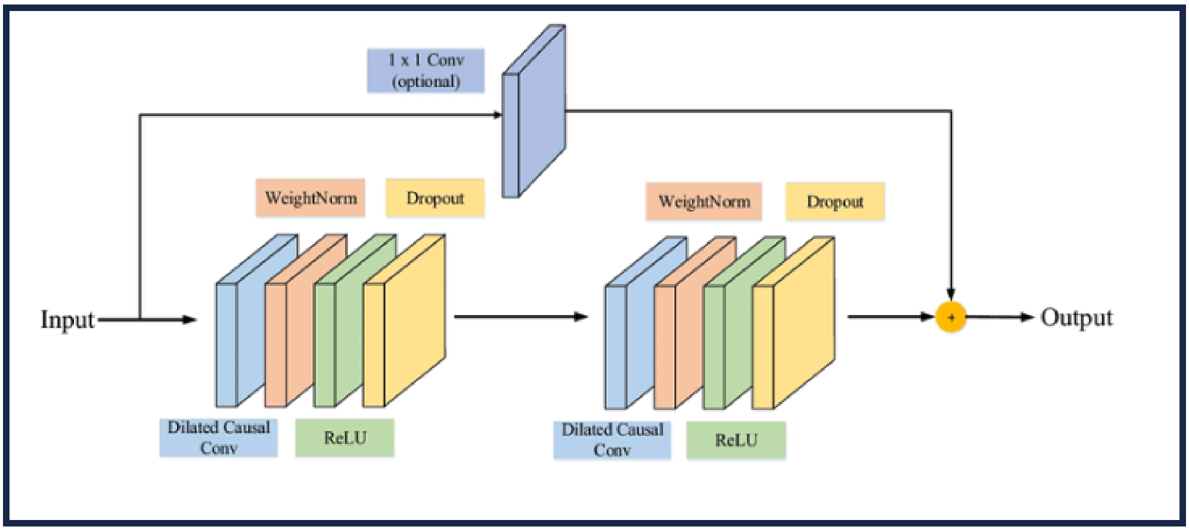

Figure 12: Simple TCN Model.

Model Training: The training process in the implemented model showed initial loss 16816.67 and improved over time, the loss decreases across iterations dropping to 1233.49 by iteration 900, Similarly the validation loss decreases from 8693.94 to 700.36 indicating that model is learning well and also generalising well to unseen data (Figure 13).

Model Evaluation: Uncertainty estimate showed by standard error of 0.3371 suggest a reasonable level of confidence in predictions. Also, the model's performance metrics display MAE of 0.1228 indicating low prediction error as well as R2 score of 0.8799 suggests that the model explains about 87.99% of variance in the data. The Root Mean Squared Error (RMSE) of 0.1592 represents the average of predictions errors (Table 4). The predicted earthquake magnitudes closely align with true magnitudes, showing the model's accuracy (Figure 14). Overall, the model is making accurate predictions with good level of certainty.

Metrics	**Value**
Uncertainty Estimate (Standard Error)	0.3371
Standard Deviation	0.0956
Mean Absolute Error (MAE)	0.1228
Root Mean Squared Error (RMSE)	0.8799
R-squared Score (R^2)	0.1592

Table 4: Bayesian TCN Performance Metrics

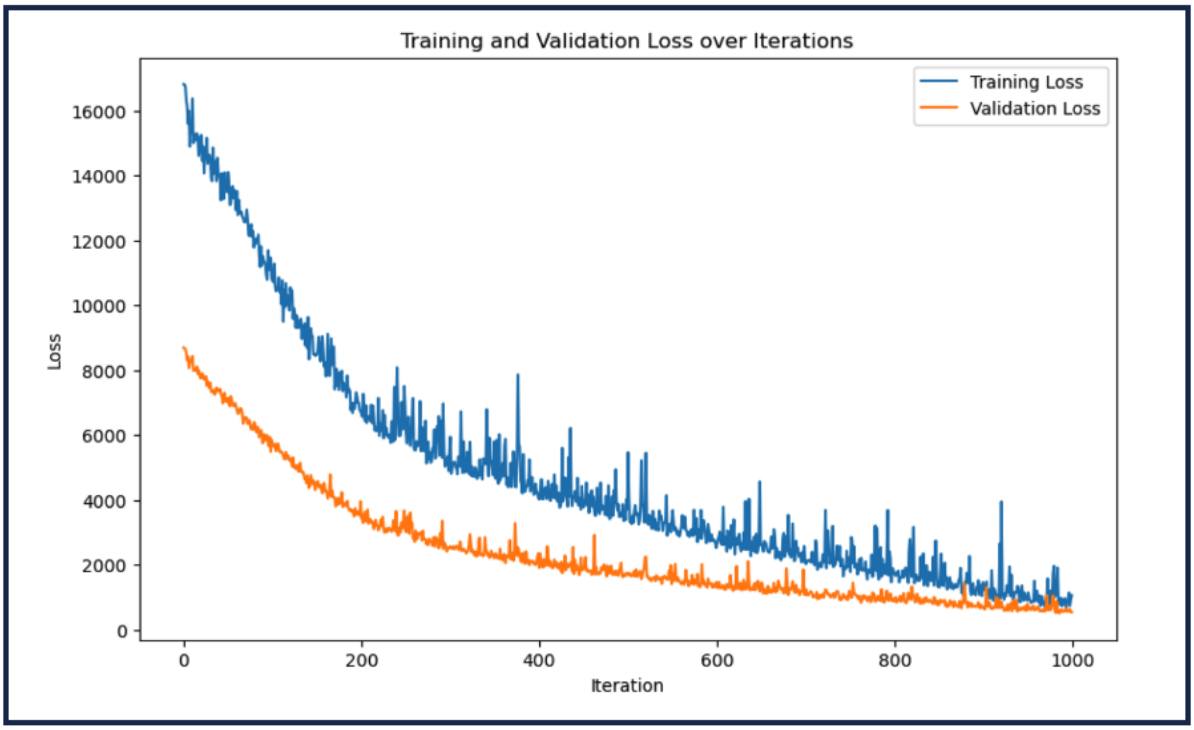

Figure 13: Bayesian TCN Training and Validation over Iterations.

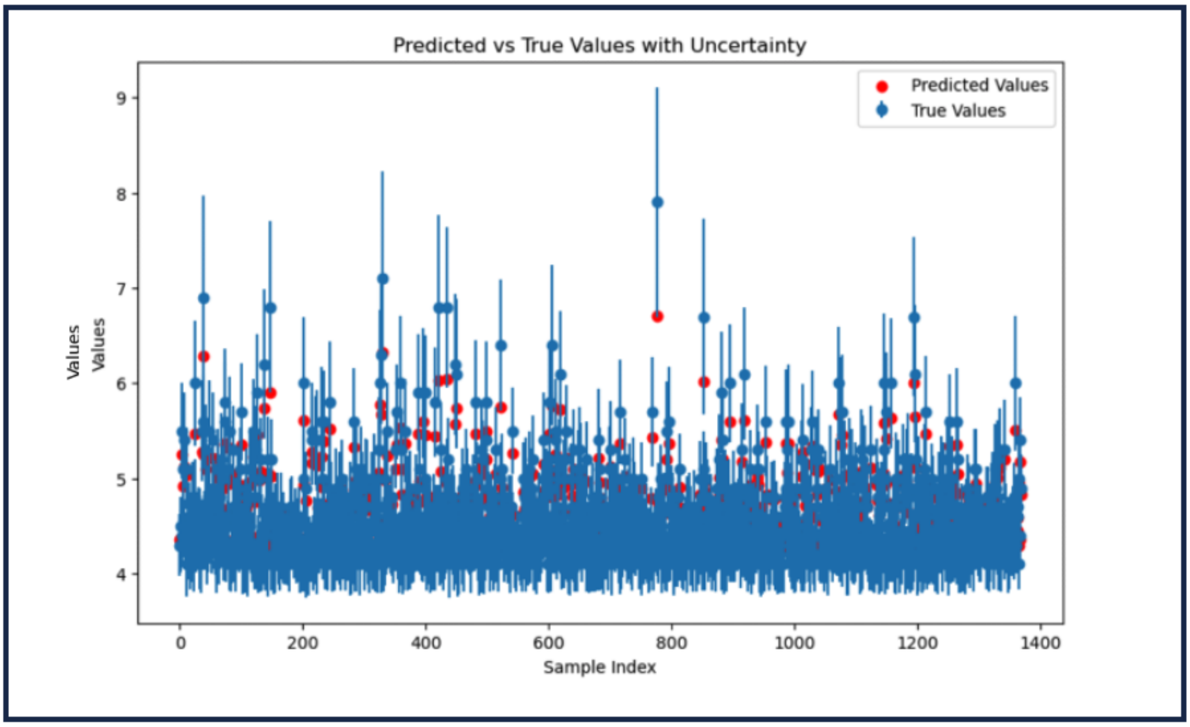

Figure 14: Bayesian TCN Predicted vs True Earthquakes Magnitudes.

5.4.4 Hybrid Bayesian CNN/LSTM Model

The hybrid Convolutional Neural Networks (CNNs) and the Long Short-Term Memory (LSTM) networks model is very efficient in processing time series data. It uses a CNN layer to analyse each time step of data, capturing local patterns with convolutional filters. After, max pooling layer is applied to reduce the size of these features, improving management of data. When the spatial features are extracted, they are flattened to the format that LSTM layers can process. The component of LSTM checks the sequence data, capturing temporal dependencies and long-term patterns. Finally, these learnt features are mapped to produce the final output using suitable activation functions for classification and regression Lin et al. (2017).

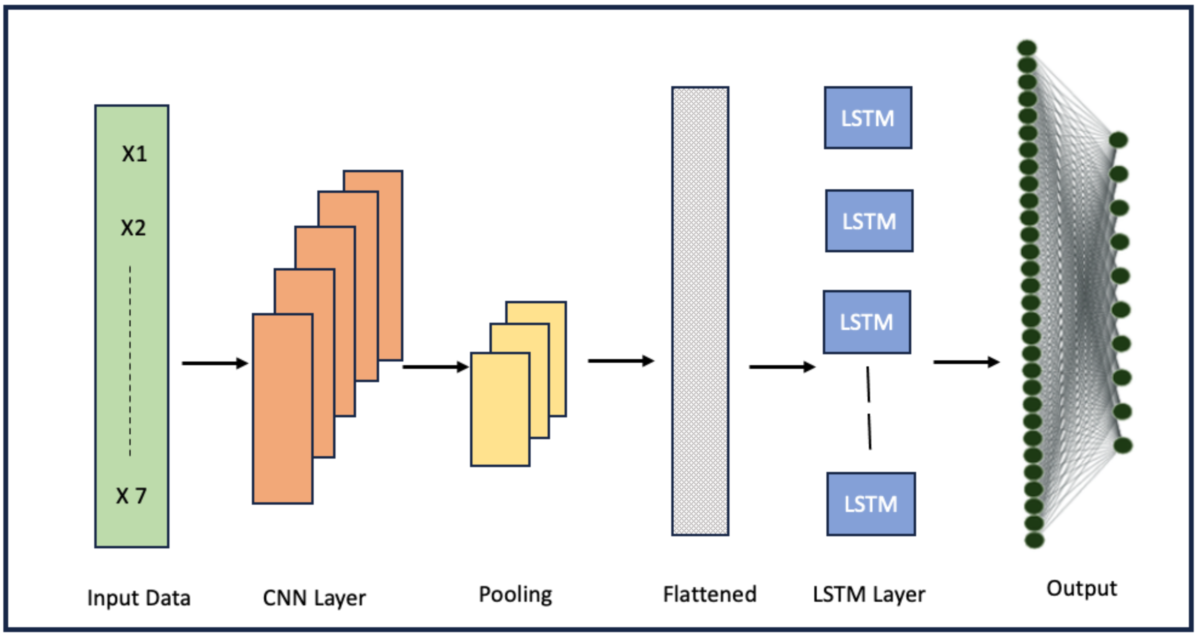

Figure 15: Hybrid CNN/LSTM Model.

Model Training: In the experiment the model starts with two convolutional layers, the first one has 32 filters each with kernel size of 3 and uses probabilistic weights and biases, that means each one of them is treated as a random variable following normal distribution. After, the max pooling is applied to reduce the output size, which helps to also reduce the data while it retains the important information. This operation is followed with a second convolutional layer with 64 filters, also using probabilistic weights and biases for feature extraction. After another max pooling layer is applied to further reduce the outputs dimensions. Following the convolutional layers process, data is reshaped to fit the input format of LSTM Layer. Data is processed through LMST Layer, capturing temporal dependencies using a number of hidden units to encode sequence information. Following the LSTM Layer, the model uses a fully connected layer with probabilistic weights and biases to map the output to hidden dimension. The Monte Carlo dropout layer is applied for uncertainty calculation and the output layer (fully connected layer) produces the final predictions based on probabilistic weights and biases. The model is trained using stochastic variation interference (SVI) that approximates the final distribution of the model's parameters. To minimise the loss function, Adam optimizer is used, and the training loop updates the model's parameters over 1000 iterations, calculating validation loss to monitor performance of unseen data. Once trained, the model is evaluated using the Monte Carlo Dropout to estimate predictive uncertainty, using several forward passes of Pyro's predictive class sampling predictions. Finally, the key metrics are produced. The training results of Hybrid Bayesian CNN/LSTM model show significant reduction in training and validation loss over 1000 iterations, pointing at effective learning. The training loss decreases from 29690.43 to 3609.86 and the validation loss drops from 21328.30 to 1696.30 illustrating model's improved fit (Figure 16).

Model Evaluation: The Mean Absolute Error (MAE) 0.2130 indicates that predictions are close to the true values (Figure 17), while the R2 score of 0.6093 suggests that the model explains 60.93% of the variance in data. The RMSE of 0.2871 shows that predictions shift slightly from true values on average. The Uncertainty Estimates with standard error of 0.4331 and Standard Deviation of 0.0860 indicates a good level of confidence in predictions. Overall, the model performs well, balancing accuracy and uncertainty, but it has potential for future improvements (Table 5).

Metric	**Value**
Uncertainty Estimate (SE)	0.4331
Standard Deviation	0.0860
Mean Absolute Error (MAE)	0.2130
R-squared Score (R^2)	0.6093
Root Mean Squared Error (RMSE)	0.2871

Table 5: Hybrid CMM/LSTM Performance Metrics

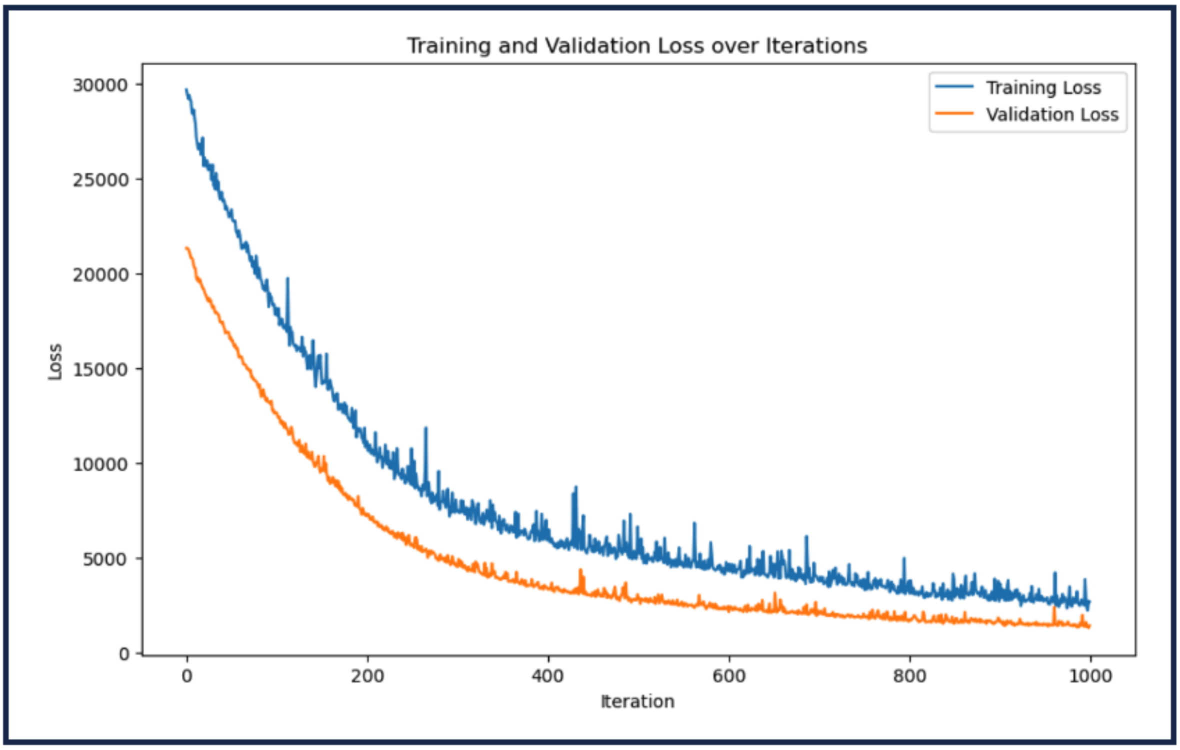

Figure 16: Hybrid Bayesian CNN/LSTM Training and Validation over Iterations.

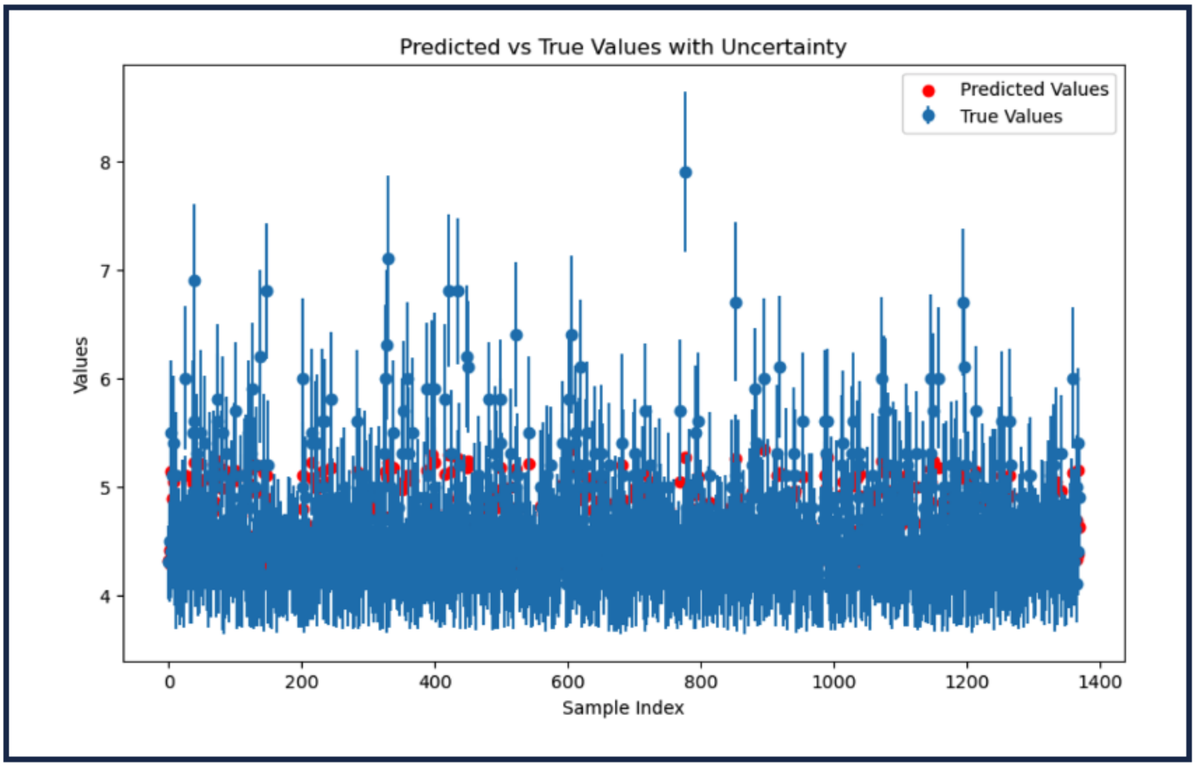

Figure 17: Hybrid Bayesian CNN/LSTM Predictions vs True Values with Uncertainty.

5.5 Conclusion

Based on evaluation metrics (Figure 18), the Bayesian LSTM model stands out as the most effective, it has the lowest uncertainty estimate of 0.1452, and the lowest standard deviation indicating confidence and consistency in its forecast. It has also achieved the lowest Mean Absolute Error (MAE) of 0.0322 and lowest Root Mean Squared Error (RMSE) of 0.0453, showing great predictive accuracy. Furthermore, it has an impressive R-squared score of 0.9903 indicating that it accounts for almost all the variability in the data. However, the Bayesian CNN model has the highest uncertainty estimate of 0.6334 and the largest MAE of 0.4008, suggesting lower degree of confidence and higher average error in predictions. Additionally, its R2 score is negative (-0.1423), which suggests that it is a poor fit. Bayesian TCN model shows the highest standard deviation of 0.095 and low R-squared score of 0.1592 indicating more prediction variability and limited explanatory power. However, it performs better than CNN in terms of MAE and RMSE. The Hybrid Bayesian CNN/LSTM model comes out as a top performer overall because of its high accuracy and minimal uncertainty. Its balanced performance of uncertainty estimate (0.4331) and low RMSE (0.2871) and MAE (0.2130) with solid R-squared score of 0.6093, indicates that the hybrid approach benefits from handling temporal dependencies by LSTM model and the feature extraction abilities of CNNs. In summary, the Bayesian LSTM model presents as the best overall performer due to its high accuracy and low uncertainty and Hybrid approach offers a strong alternative.

Model	Uncertainty Estimate (SE)	Standard Deviation	Mean Absolute Error (MAE)	Root Mean Squared Error (RMSE)	R-squared Score (R2)
Bayesian TCN	0.3371	0.0956	0.1228	0.8799	0.1592
Bayesian CNN	0.6334	0.0615	0.4008	0.4910	-0.1423
Bayesian LSTM	0.1452	0.0453	0.0322	0.0453	0.9903
Hybrid Bayesian CNN/LSTM	0.4331	0.0860	0.2130	0.2871	0.6093

Figure 18: Evaluation Comparison Table.

6 Conclusion and Future Work

This research investigated the use of deep learning techniques for earthquake prediction, focusing on calculating and reducing uncertainty with the goal of improving risk management which can be integrated into disaster emergency planning and early warning system. This study aimed to achieve one of the several objectives by developing deep learning models such as Bayesian Long Short-Term Memory (LSTM), Bayesian Convolutional Neural Networks (CNN), Bayesian Temporal Convolutional Networks (TCN), and a Hybrid Bayesian CNN/LSTM model. All the models were used with the Monte Carlo Dropout Method to calculate uncertainty. Among these models, the Bayesian Long Short-term Memory model outperformed the other models with the lowest Mean Absolute Error (MAE) of 0.0322 and Root Mean Squared Error (RMSE) of 0.0453 and R-squared score of 0.9903 indicating that it accounted for nearly all the variability in data. Also, the hybrid Bayesian CNN/LSTM model performed well and showed good balance between accuracy and uncertainty. The research identified that while the Bayesian models offer improved uncertainty, they also require the significant computational resources. These discoveries are consistent with the objectives of creating a model that decrease uncertainty and increase forecast accuracy, identifying the most effective techniques for earthquake's magnitude prediction. Also, through the experiment it was established that greater computational resources are needed to handle deep learning models and large amount of data, which is essential for accurate predictions. The future work should address integrating more diverse data like satellite images, sensor outputs, and real time data, also creating different hybrid deep learning models has a great potential as demonstrated by experiment. Finally, sharing results and integrating it with the early warning application system can help with preparation for disaster and save lives. Further attention should also be given to technical and regulatory compliance to guarantee correct implementation of these technologies.

Acknowledgment

I would like to express my sincere gratitude to my supervisor Dr Catherine Mulwa for her guidance and support throughout the Research Project. I am grateful for her dedication, expertise, and encouragement in writing this project. Also, I would like to thank my partner, friends, and family for their constant support and understanding. Their support and encouragement were invaluable throughout this process.

References

Baker, J. and Cornell, C. (2008). Uncertainty propagation in probabilistic seismic loss estimation, *Structural Safety* **30**(3): 236–252.

Bergen, K., Johnson, P., de Hoop, M. and Beroza, G. (2019). Machine learning for data-driven discovery in solid earth geoscience, *Science* **363**(6433): eaau0323.

Blundell, C., Cornebise, J., Kavukcuoglu, K. and Wierstra, D. (2015). Weight uncertainty in neural network, *International Conference on Machine Learning*, PMLR, pp. 1613–1622.

Bradley, B. (2009). Seismic hazard epistemic uncertainty in the san francisco bay area and its role in performance-based assessment, *Earthquake Spectra* **25**(4): 733–753.

Breiman, L. (1996). Bagging predictors, *Machine Learning* **24**: 123–140.

Cramer, C., Petersen, M. and Reichle, M. (1996). A monte carlo approach in estimating uncertainty for a seismic hazard assessment of los angeles, ventura, and orange counties, california, *Bulletin of the Seismological Society of America* **86**(6): 1681–1691.

Damianou, A. and Lawrence, N. (2013). Deep gaussian processes, *Artificial Intelligence and Statistics*, PMLR, pp. 207–215.

Farhadi, A. and Mousavi, M. (2016). Consideration of the rupture model uncertainties in the probabilistic seismic hazard analysis, *Soil Dynamics and Earthquake Engineering* **83**: 191–204.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning, *International Conference on Machine Learning*, PMLR, pp. 1050–1059.

García, S., Luengo, J. and Herrera, F. (2015). *Data preprocessing in data mining*, Vol. 72 of *Cham, Switzerland*, Springer International Publishing.

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision?, *Advances in Neural Information Processing Systems* **30**.

Lanzano, G., D'Amico, M., Felicetta, C., Puglia, R., Luzi, L., Pacor, F. and Bindi, D. (2016). Ground-motion prediction equations for region-specific probabilistic seismic-hazard analysis, *Bulletin of the Seismological Society of America* **106**(1): 73–92.

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning, *Nature* **521**(7553): 436–444.

Lin, T., Guo, T. and Aberer, K. (2017). Hybrid neural networks for learning the trend in time series, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pp. 2273–2279.

McGuire, R. (1995). Probabilistic seismic hazard analysis and design earthquakes: closing the loop, *Bulletin of the Seismological Society of America* **85**(5): 1275–1284.

Mousavi, S. and Beroza, G. (2019). Bayesian-deep-learning estimation of earthquake location from single-station observations, *arXiv preprint arXiv:1912.01144* .

Moustra, M., Avraamides, M. I. and Christodoulou, C. (2011). Artificial neural networks for earthquake prediction using time series magnitude data or seismic electric signals, *Expert Systems with Applications* **38**(12): 15032–15039.

Neal, R. (2012). *Bayesian Learning for Neural Networks*, Vol. 118, Springer Science & Business Media.

Sensoy, M., Kaplan, L. and Kandemir, M. (2018). Evidential deep learning to quantify classification uncertainty, *Advances in Neural Information Processing Systems* **31**.

Wang, M. and Takada, T. (2009). A bayesian framework for prediction of seismic ground motion, *Bulletin of the Seismological Society of America* **99**(4): 2348–2364.

Wang, T., Li, H., Noori, M., Ghiasi, R., Kuok, S. and Altabey, W. (2022). Probabilistic seismic response prediction of three-dimensional structures based on bayesian convolutional neural network, *Sensors* **22**(10): 3775.

Yagi, Y., Okuwaki, R., Enescu, B., Hirano, S., Yamagami, Y., Endo, S. and Komoro, T. (2014). Rupture process of the 2014 iquique chile earthquake in relation with the foreshock activity, *Geophysical Research Letters* **41**(12): 4201–4206.

Yu, Y., Si, X., Hu, C. and Zhang, J. (2019). A review of recurrent neural networks: Lstm cells and network architectures, *Neural Computation* **31**(7): 1235–1270.

Zargar, S. (2021). Introduction to sequence learning models: Rnn, lstm, gru, *Technical report*, Department of Mechanical and Aerospace Engineering, North Carolina State University.

Zhao, B., Lu, H., Chen, S., Liu, J. and Wu, D. (2017). Convolutional neural networks for time series classification, *Journal of Systems Engineering and Electronics* **28**(1): 162–169.