uc3m | Universidad Carlos III de Madrid

MASTER IN BIG DATA ANALYTICS

2023-2024

Master's Final Project

'Calibration in deep neural networks through probabilistic generative models"

Javier Méndez García-Brioles

Tutor

Pablo Martínez Olmos

Online, 9 of July

ABSTRACT

In this project, we provide an in-depth analysis of machine learning models, specifically focusing on neural networks, and explore questions related to the predictions of these models. Our main goal is to confirm that without external calibration, a neural network's confidence in its predictions cannot be trusted, making external calibration essential for achieving the best possible models.

Confidence calibration, which involves predicting probability estimates that accurately represent the true likelihood of correctness, is a critical aspect of classification models in various applications. Building on the foundational work of [1], this thesis explores the calibration of modern neural networks, addressing the observed deficiencies in the calibration of contemporary models compared to those from a decade ago. Additionally, we compare these models once they are calibrated.

Through extensive experimentation, we examine the impact of factors such as network depth, width, weight decay, and Batch Normalization on the calibration of neural networks. We conduct a thorough evaluation of various post-processing calibration methods, including temperature scaling, on state-of-the-art architectures using image and document classification datasets.

Our analysis not only provides deeper insights into the learning processes of neural networks but also offers practical guidelines for improving model calibration. We found that temperature scaling, a single-parameter variant of Platt Scaling, is particularly effective in most scenarios. This research contributes to the development of more reliable and interpretable machine learning models, enhancing their applicability in real-world settings where calibrated confidence estimates are crucial.

With these results clarified, we will validate the most important calibration methods from the referenced paper for some of the provided data. Additionally, we will expand the scope to include external calibration methods primarily from this study [2], extend the data used to include audio as input, and examine whether the known behavior is comparable to the results in that new case.

Key Words:

- **Confidence Calibration**: The process of adjusting the probability estimates produced by a model to more accurately reflect the true likelihood of correctness.
- **Neural Networks**: Computational models inspired by the human brain, used for various machine learning tasks, particularly in classification and prediction.
- **Depth and Width of Networks**: Refers to the number of layers (depth) and the number of units within each layer (width) in a neural network architecture.

- **Weight Decay**: A regularization technique used to prevent overfitting by penalizing large weights in the neural network.
- **Batch Normalization**: A technique to improve the training of deep neural networks by normalizing the inputs of each layer.
- **Post-Processing Calibration Methods**: Techniques applied after the initial training of a model to improve the calibration of its probability estimates.
- **Temperature Scaling**: A simple, yet effective post-processing calibration method that adjusts the confidence of a neural network by scaling its logits with a single parameter.
- **Platt Scaling**: A method for transforming the outputs of a classification model into probability estimates, often used in support vector machines but adapted here for neural networks.
- **Audio Data**: Sound information used as input for training and evaluating machine learning models, particularly relevant in the expanded scope of this research.
- **Real-World Applications**: Practical scenarios in which calibrated confidence estimates from neural networks are crucial for decision-making and reliability.
- **Evaluation Metrics**: Standards or criteria used to assess the performance and calibration of neural network models.
- **Machine Learning**: A field of artificial intelligence focused on building systems that learn from data to make predictions or decisions.
- **Expected Calibration Error (ECE)**: A metric that quantifies the calibration of a model by comparing predicted probabilities with actual outcomes over several intervals.
- **Maximum Calibration Error (MCE)**: A metric that measures the largest deviation between predicted probabilities and actual outcomes, assessing the worst-case calibration error.
- **Negative Log Likelihood (NLL)**: A performance measure used in probabilistic models, indicating how well the predicted probabilities match the observed data.
- **Test-Driven Development (TDD):** A software development approach where tests are written before the actual code implementation. It aims to ensure that the code meets requirements and remains maintainable and bug-free throughout development.
- **Loss Function:** A function used to quantify the difference between predicted values and actual values in machine learning and optimization tasks. It serves as a measure of model performance during training.

- **Bayesian Model:** A statistical model that uses Bayesian inference for probabilistic reasoning. It updates beliefs about parameters based on prior knowledge and observed data, providing a framework for uncertainty quantification.
- **Ensemble:** A technique in machine learning where multiple models are combined to improve predictive performance. It leverages the diversity of different models to produce more robust and accurate predictions.
- **Convolutional Neural Networks (CNNs):** Deep learning models specifically designed for processing grid-like structured data, such as images or sequences. They use convolutional layers to automatically learn spatial hierarchies of features.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Motivation

Background

In recent years, machine learning models, particularly neural networks, have seen tremendous advancements and widespread application across various domains. This includes generative AI for image, text, and video generation. While this is not the main focus of our study, it can be simplified to classification and checked with calibration methods. In pure classification, there has also been significant progress, such as object detection in images and complex prediction methods. Despite their impressive performance, one crucial aspect that remains a challenge is the calibration of these models. Calibration refers to the alignment between predicted probabilities and the true likelihood of those predictions being correct. Accurate confidence estimates are essential for many applications, including medical diagnostics, autonomous driving, and financial forecasting, where decisions based on these predictions can have significant consequences. For models with similar accuracy, it is always preferable to have higher confidence, as it allows for more accurate decision making regarding the certainty of the results.

Problem Statement

Modern neural networks, unlike those from a decade ago, often exhibit poor calibration, meaning that their predicted probabilities do not accurately reflect the true likelihood of correctness. This phenomenon, while not fully understood, has been shown in previous studies that many of the optimizations in modern neural networks; such as model capacity, normalization, and regularization enhance accuracy but contribute to poor calibration.

This miscalibration can lead to overconfident or underconfident predictions, which are particularly problematic in high stakes environments. Therefore, it is crucial to develop and validate methods to improve the calibration of neural networks to ensure their reliability and trustworthiness.

Significance

Addressing the calibration problem is vital for enhancing the practical utility of neural networks in real world applications. Well calibrated models can provide more reliable confidence estimates, enabling better decision making processes. For instance, in medical diagnosis, a well calibrated model can help clinicians understand the confidence level of a diagnosis, leading to better patient outcomes. If the model is not confident in its prediction, clinicians might be more inclined to rely on their judgment, which is only possible if they can trust the model's confidence level. Similarly, in autonomous driving, accurate confidence estimates can improve safety by ensuring that decisions made by the vehicle's AI are based on reliable information, defaulting to other factors otherwise. Solving the calibration issue can significantly increase the trust and adoption of neural networks in critical areas and ensure that these models are trusted by the public. This is important because trust in AI has been in decline since the explosion of generative models, as evidenced by studies [3] [4]

Research Questions

This research aims to explore the following specific questions:

1. What are the primary factors contributing to the poor calibration of modern neural networks?
2. Which post processing calibration methods are most effective in improving the calibration of neural networks?
3. How do these calibration methods perform across different types of data, including image, document, and audio datasets?
4. Are the results from the calibration techniques proposed in previous studies replicable?
5. Do the calibration techniques proposed in previous studies maintain the same level of improvement when applied to audio datasets?

1.2. Objectives

Primary Goal

The primary goal of this research is to investigate and validate the effectiveness of various post-processing calibration methods in improving the calibration of modern neural networks. This involves replicating the results of previous studies [1] to verify their findings and ensure consistency across different experimental setups. The goal is to confirm that without external calibration, the confidence of modern neural networks in their predictions cannot be fully trusted.

Secondary Goals

1. Extend the scope of calibration methods by incorporating techniques from additional sources, primarily from recent research [2], to test their effectiveness.
2. Assess the performance of calibration methods on new types of data, specifically audio datasets, to determine if the known behavior is consistent across various data types.
3. Identify the key factors that contribute to poor calibration in modern neural networks, such as model capacity, normalization techniques, and regularization methods, to replicate the results from [1].

Measurable Outcomes

The success of this research will be measured through several key outcomes:

1. Calibration Metrics: Improvement in standard calibration metrics such as Expected Calibration Error (ECE), Maximum Calibration Error (MCE), and Negative Log-Likelihood (NLL) will be quantitatively assessed before and after applying calibration methods.
2. Replicability: Successful replication of previous studies results will be verified by achieving similar improvements in calibration metrics on identical datasets and neural network architectures.
3. Generalizability: The effectiveness of calibration methods proven to work for data types like images and tabular data will be evaluated for audio data, with the goal of achieving consistent results across these datasets.
4. Comparison of Methods: Comparative analysis of various calibration methods to identify the most effective techniques for different types of neural network architectures and data modalities.

5. Documentation: The research will produce comprehensive documentation to provide a more in-depth understanding of calibration methods applied to neural networks, facilitating their use in the real world.

1.3. Restrictions

Scope

The research is limited to the investigation and validation of post-processing calibration methods for modern neural networks. This includes replicating results from previous studies [1] [2] and extending these methods to new data types such as audio datasets. The focus is primarily on classification tasks; while generative models and other types of machine learning tasks are acknowledged, they are not the primary focus of this study.

Technical Constraints

Several technical constraints could impact the research:

1. Computational Resources: The research requires significant computational power for training and evaluating the models, which will be limited by the available hardware. This constraint is particularly important when we talk about the image datasets that might be beyond our storage capabilities at this scale.

2. Data Availability: The availability and quality of the different datasets can be a main issue in this study. If we are not able to obtain exactly the same datasets used in the study [1], we won't be able to fairly replicate their results.

3. Software Dependencies: The research relies on specific machine learning libraries in Python that have been updated over the years, possibly making the code used in the study [1] obsolete. This is particularly relevant for temperature scaling [5], where the original code might need adjustments.

Time and Resources

The research is planned to be conducted over a period of 3 months. This time frame includes stages for literature review, experimental setup, data collection, model training, evaluation, and documentation. The resources available include:

1. Hardware: Access to a personal computer with 16GB RAM and capacity for GPU processing.

2. Software: Free-use software within an Anaconda environment with CUDA for parallelization.

3. Time: As per the metric of 1 ECTS credit amounting to approximately 30 hours, with a weight of 6 ECTS credits, a total of 180 hours will be dedicated to the project.

Ethical Considerations

Although this research is not in a highly sensitive domain and all our data is publicly available, we adhere to an ethical code to maintain scientific integrity:

1. Transparency and Reproducibility: Maintaining transparency in the research methodology and providing sufficient details to ensure that the experiments can be replicated by other researchers. This will be addressed by releasing the code used on GitHub for anyone who wishes to review it.

2. Impact of Miscalibration: Considering the ethical implications of deploying poorly calibrated models in high-stakes environments, such as medical diagnostics or autonomous driving, this research aims to advance the focus on calibration over pure blind accuracy optimization.

2. STATE OF THE ART

2.1. Machine Learning

Machine learning (ML) has transformed various fields by enabling computers to learn from data and make predictions. Within ML, neural networks, particularly deep learning models, have seen significant advancements, improving accuracy across diverse applications. Since the surge of generative models has made the public more aware of AI in general, we have the obligation to provide not only the most powerful models we can, but also models that can be trusted. [3]

Algorithms of Interest

The effectiveness of different neural network architectures used in the study [1] has been extensively studied:

- **ResNet (Residual Networks)**: Introduced by He et al. (2015), ResNets use skip connections to address the vanishing gradient problem, allowing for very deep networks. They have shown state-of-the-art performance and are easy to use and evaluate.[6]
- **DAN (Deep Averaging Network)**: This model averages word vectors and passes them through multiple layers. It is particularly used in NLP tasks but requires calibration for reliable probability estimates.[7]
- **CNN**: Convolutional Neural Networks (CNNs) are designed to automatically and adaptively learn spatial hierarchies of features through backpropagation by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers. Widely used in any complex classification tasks, CNNs should also benefit from calibration. [8]

Applications

Let's mention the applications where it is of high importance that a machine learning algorithm is trustworthy:

- **Autonomous Driving**: Neural networks detect objects and make navigation decisions. A calibrated driving model ensures that the vehicle can handle uncertain predictions by relying on additional sensors or deferring decisions in any high-stakes situation.[9]

- **Medical Diagnosis**: Models assist in disease detection and treatment recommendations. Calibrated probabilities help healthcare professionals understand the confidence level of predictions, aiding in better decision-making. [10]

Challenges and Limitations

Despite the advancements in neural networks, several challenges and limitations persist in the field of calibration:

- **Model Complexity**: Modern neural networks, with their increased depth and complexity, tend to be poorly calibrated compared to simpler models. This issue can't be avoided as we continually strive for better models, which are almost always linked to larger models.
- **Lack of Understanding**: The reasons behind the miscalibration of modern networks are not fully understood. While some hypotheses point to training methods and architectural choices, comprehensive explanations are still lacking. In other words, we can't exactly pinpoint why this is the case.
- **Post-Processing Calibration Methods**: While techniques like temperature scaling, Platt scaling, and others have shown promise, their effectiveness can vary across different datasets and model architectures. Ensuring consistent improvements in calibration across different datasets or architectures is still uncertain.
- **Computational Resources**: Implementing and validating post-processing calibration methods require extra computational power that most people might not see as a worthwhile investment.

2.2. Calibration Methods

Calibration is the process that involves fine-tuning the predicted probabilities of a model to align more closely with the actual likelihood of outcomes. This is crucial because models like neural networks often generate probabilities that don't accurately represent how likely events truly are. Calibration ensures that these probabilities are trustworthy and genuinely reflect the model's confidence in its predictions. Here we will see the ways of reaching that calibration.

Main Methods

Several methods are currently used to calibrate neural networks to improve the reliability of their predicted probabilities:

- **Histogram Binning**: This non-parametric technique divides predicted probabilities into bins and adjusts the probabilities within each bin to match the observed frequencies. While simple, it may not scale well with large datasets.
- **Isotonic Regression**: Another non-parametric method, isotonic regression fits a piecewise constant non-decreasing function to the predicted probabilities. It is effective but can lead to overfitting, especially with limited data.
- **Bayesian Binning into Quantiles (BBQ)**: An extension of histogram binning, BBQ uses Bayesian principles to determine the optimal binning and probability adjustment, providing a more robust calibration.
- **Temperature Scaling**: Uses a single scalar parameter $T > 0$ for all classes. Given the logit vector z_i, the new confidence prediction is $\hat{q}_i = \max_k \sigma_{\text{SM}}(z_i/T)^{(k)}$. The parameter T is optimized with respect to the NLL on the validation set. Temperature scaling does not affect the model's accuracy but adjusts the confidence estimates.
- **Vector Scaling**: An extension of temperature scaling that adjusts each logit by a different parameter, offering more flexibility at the cost of increased complexity.
- **Matrix Scaling**: Further extends vector scaling by using a matrix to transform the logits, allowing for even more flexible adjustments but also requiring more data and computation.

More invasive methods

This methods can't just be plastered at the end, when a model is already trained, they need to be apply during the training of the model.

- **Deep Ensemble Methods**: Combining predictions from multiple models trained with different initializations or subsets of data to improve calibration. The diversity among the models helps to reduce miscalibration.[11]
- **Bayesian Neural Networks**: Incorporating Bayesian principles into neural networks to provide calibrated uncertainty estimates. These models naturally output probabilistic predictions that are better calibrated.[12]
- **Calibrated Loss Functions**: Modifying the loss function used during training to directly optimize for calibration metrics like Expected Calibration Error (ECE) in addition to accuracy, leading to better-calibrated models.[13]
- **Self-Calibration Techniques**: Methods where the model internally adjusts its confidence estimates during training, using techniques such as self-distillation, where the model learns from its own softened outputs.[14]
- **Two-Stage Training (TST)**: Involves training the model initially and then freezing some parameters and fine-tuning the remaining parameters specifically for calibration. This method can be highly effective and is easy to implement but is quite resource-intensive.[2]

Algorithm 1 Two-Stage Training (TST)

1: Initialize DNN M with parameters $\{\beta, \phi\}$
2: **Stage 1:** Train M with cross-entropy loss on D_{train} until convergence or early stopping
3: Freeze parameters β of M
4: Re-initialize FC layers of M with parameters $\{\theta, \nu\}$
5: **Stage 2:** Train $\{\theta, \nu\}$ of M with cross-entropy loss on D_{train} until convergence

Best Practices

Best practices for calibration are continually evolving as the field progresses:

- **Evaluation Metrics**: Standard metrics for evaluating calibration include Expected Calibration Error (ECE) and Negative Log-Likelihood (NLL). These metrics should be routinely reported alongside accuracy in model evaluation.
- **Calibration Layers**: Implementing simple calibration layers such as temperature scaling as a post-processing step in the model pipeline ensures that the final probabilities are more reliable without significantly altering the model architecture.
- **Post-Processing Calibration Methods**: While ensuring a model's focus on calibration can be challenging, implementing post-processing calibration methods remains the easiest way to achieve reliable calibration.

3. ANALYSIS

3.1. Description

System Overview

The system under development is designed to enhance the calibration of neural network models to improve the reliability of their predicted probabilities. This system aims to integrate the discussed calibration techniques into a simple workflow that can be easily applied to various neural network architectures and datasets, incentivizing the expansion of the field.

Components

The system is comprised of the following main components:

- **Data Preprocessing Module**: This module handles all data-related tasks, including cleaning, normalization, and augmentation of the input data as needed to ensure it is standardized for training and calibration.
- **Neural Network Module**: The core component where various neural network architectures are defined and trained on different input data.
- **Calibration Module**: Implements different calibration techniques such as Temperature Scaling or Two-Stage Training to adjust the network's output probabilities.
- **Evaluation Module**: Evaluates the performance of the models, both before and after calibration, using metrics such as Expected Calibration Error (ECE). It also tracks the accuracies of the model to check if it decreases with calibration.
- **Testing Module**: As we will be following a Test Driven Development (TDD) approach [15], it is very important to have a testing module to maintain a high test coverage and ensure high code quality.
- **Result Module**: This module generates visual representations of the model's performance, including all relevant statistics and graphs. It helps us intuitively understand the calibration effectiveness and overall performance of the neural network models.

Pipeline

The pipeline shows how data flows through the system in the following manner:

1. **Input Data Collection**: Raw data is collected from various sources and fed into the Data Preprocessing Module.
2. **Data Preprocessing**: The data is cleaned, normalized, and augmented as necessary before being fed to the Neural Network Module.
3. **Model Training**: Preprocessed data is used to train the selected neural network within the Neural Network Module. Depending on the input data, we should be consistent with the study [1]. After training, the model is sent to the Evaluation Module to get the uncalibrated results, and a copy of the model is sent to the Calibration Module to apply the necessary calibration.
4. **Calibration**: The Calibration Module takes the uncalibrated probabilities from the trained model and applies the selected calibration techniques. The calibrated model is then sent to the Evaluation Module to be evaluated and compared.
5. **Evaluation**: The calibrated probabilities are evaluated using the Evaluation Module to assess performance and reliability, and the results are sent to the Result Visualization Module.
6. **Result Visualization**: Visual representations of the model's performance are generated as the final output of our pipeline.

3.2. Requirements

Functional Requirements

ID	Name	Description
FR1	Heterogeneous Input	The system must accept raw input data from various sources and formats (image, tabular data, and audio).
FR2	Preprocessing	The system must preprocess raw input data to ensure it is suitable for training. This may include cleaning, normalization, and augmentation of data.
FR3	Implement Different Models	The system must support the implementation and training of a wide variety of neural network models in accordance with the study [1].
FR4	Model Training	The system must train the implemented neural network models on the preprocessed data.
FR5	Create Calibration Methods	The system must implement various calibration techniques to adjust the predicted probabilities of the models.
FR6	Apply Calibration Methods	The system must apply the selected calibration techniques to the trained models.
FR7	Create Evaluation Metrics	The system must define and implement standard metrics for evaluating model performance and calibration.
FR8	Evaluate Models	The system must evaluate the performance of both calibrated and uncalibrated models using the defined metrics.
FR9	Model Comparison	The system must compare the performance of all different models and calibration techniques for each dataset.
FR10	Show Results	The system must provide the results of the evaluation.

TABLE 3.1. FUNCTIONAL REQUIREMENTS

Non-Functional Requirements

ID	Name	Description
NFR1	Performance	The system should perform data preprocessing, training, calibration, and evaluation within a reasonable time frame.
NFR2	Processing Power	The system should be able to be executed on a simple personal computer, regardless of the number of large datasets and complex neural networks provided.
NFR3	Storage	The system should not have storage problems on a simple personal computer.
NFR4	Reliability	The system should produce consistent and reliable results across different runs.
NFR5	Maintainability	The system should be easy to maintain and update with new calibration techniques and model architectures.

TABLE 3.2. NON-FUNCTIONAL REQUIREMENTS

Risk Assessment

Here we will discuss in depth the risks that might compromise the results of the study:

- **Data Risks**: Difficulty in acquiring the data used by the study[1] might lead us to datasets with poor quality or insufficient data that could affect the reliability of the study.

  Mitigation: As the study[1] has many different databases, we can choose to use only those that are good for our project and discard any that are difficult to obtain.

- **Model Complexity Risks**: As we are using somewhat complex models, this might lead to increased computational requirements that are not at our disposal.

  Mitigation: In such a case, we will exclude the study on images, as that is the most computationally intensive part of the study.

- **Calibration Technique Risks**: Some calibration techniques might not perform well on certain datasets or model architectures.

  Mitigation: Provide a variety of calibration methods to ensure replication of some of the results of the study[1].

- **Incongruency Risks**: Any misinterpretation of the calibration methods or the databases provided might cause our results to vary from those in the study[1].

  Mitigation: Be careful throughout the process pipeline.

Feasibility Study

Once we know the risks of the operation, we can conduct a feasibility study to determine the viability of the project:

- **Technical Feasibility**: Existing scientific literature indicates that the project leverages existing neural network frameworks and calibration techniques, making it technically feasible. The required computational resources are available and scalable.
- **Computational Feasibility**: As our project will be developed on a personal computer, the computational power is limited. Thus, it is not feasible to replicate the full study. This is due to the large size of image datasets and models, so we should exclude them in favor of focusing on tabular data.
- **Time Feasibility**: A complete check of the entire study is not only infeasible but also not the best use of our time. Therefore, we will reduce the scope of the full check of the study[1] to allow time for extensions in calibration methods and audio databases.

Overall, the project seems feasible and holds the potential to emphasize the importance of the reliability of neural network predictions achieved through external calibration techniques. However, we will need to drop the checking of the results for the image datasets due to insufficient computational power, as well as most calibration methods of the study due to time constraints. This should not be an issue as the main focus of the study we want to check is Temperature Scaling.

4. DESIGN

4.1. Data

Data Collection

The process of data collection involves gathering raw data from various sources to ensure a diverse and comprehensive dataset for training and evaluation.

- **Sources:** Data will be collected from the following sources: TODO fill the sources with the URLs

Source	Type of Data	Description
Kaggle	Image, Tabular	Public datasets covering a variety of domains, often used in competitions.
Python libraries	Tabular	A collection of databases implemented as a Python library.
Google Dataset Search	Image, Tabular, Audio	A tool for finding datasets stored across the web, covering various data formats and topics.

TABLE 4.1. DATA SOURCES

- **Data Types:** The datasets will include the following types of data:

Data Type	Format	Description
Image	JPEG, PNG	Visual data used for tasks such as classification, object detection, and segmentation.
Tabular	CSV, txt	Structured data organized in rows and columns, typically used for classification and regression tasks.
Audio	WAV, MP3	Sound data used for tasks such as speech recognition, classification, and sentiment analysis.

TABLE 4.2. DATA TYPES

- **Datasets:** Now we will specify the dataset that we are going to use:

Datasets	Type	Description
Stanford Cars Dataset	JPEG	The Car dataset used for the study, found in kaggle .[16]
20 News	CSV	It is a dataset that has around 18000 newsgroups posts on 20 topics from the native python library scikit [17] [18]
Reuters	TXT	Is a dataset of text with 90 classes used in a lot of studies.It is a collection of 10,788 documents from the Reuters financial newswire service and found in [19]
UrbanSound8K	WAV	This dataset contains 8732 labeled sound excerpts (<=4s) of urban sounds from 10 classes of diferent common noises of a city, it is from .[20]

TABLE 4.3. DATABASES

Data Preprocessing

In our case, data preprocessing is essential to prepare the raw data for training neural networks due to our diverse and numerous input datasets. The steps are:

- **Data Cleaning:** Remove noise and irrelevant information from the data, including handling missing values and correcting inconsistencies. This is consistent across all our datasets, so it is safe to do without discriminating by data type.
- **Clusterization:** We divide the different datasets by type (tabular, image, audio) for further heterogeneous processing of the data.
- **Normalization:** Standardize the data to a common scale without distorting differences in the range of values. This should be the same for all types of data, but because we have data of different dimensions, we find it easier to do after separation.
- **Data Transformation:** Convert data into a format suitable for model input, such as converting categorical data to numerical values for tabular data, resizing for image data, or feature selection for audio data.

Data Augmentation

Data augmentation techniques will be used to artificially increase the size of the training dataset, which helps in improving the robustness and performance of the models. This is an optional process, but as we use it to bump the accuracy of some models, it is important to mention it:

- **Image Augmentation:** There exist techniques such as rotation, flipping, scaling, and color adjustment but because of our hardware limitations we will not be using them.
- **Audio Augmentation:** In the same sentiment, there exist methods to augment audio data but we will not be using them.
- **Tabular Data Augmentation:** Synthetic data generation methods, in concrete oversampling, will be used to balance classes in tabular datasets.

4.2. Model Creation

Algorithm Selection

The algorithms selected for this project are ResNet and DAN, because they are the ones used by the study [1]. For the additional datasets added to the study, we did consider using pre-trained models like VGGish to achieve the best possible results with minimal computational power, but because this has focus in calibration and not accuracy we developed a simple CNN to have more control over it.

Model Training

The training process follows the Train-Validation-Test split as described in the base study:

- **Training:** The models are trained on the preprocessed datasets using a set of predefined hyperparameters.
- **Validation:** During training, the models are validated using a separate validation set to tune hyperparameters and prevent overfitting.
- **Testing:** After training, the models are evaluated on a test set to assess their generalization performance.

Calibration

The design process for using calibration methods includes:

- **Temperature Scaling:** A post-processing technique applied to the model outputs to adjust the predicted probabilities. Temperature scaling involves multiplying the logits (inputs to the softmax function) by a temperature parameter before applying softmax. This simple method is effective for many models, making the probabilities more reflective of the true likelihoods without altering the model's accuracy.

- **Two-Stage Training:** A more computationally complex method involving additional training stages specifically designed to improve model calibration. In the first stage, the model is trained normally. The second stage focuses on calibration by optimizing a calibration-specific loss function. This method can significantly enhance calibration at the cost of additional training time and computational resources.

- **Bayesian Model:** Bayesian calibration incorporates uncertainty into the model predictions using Bayesian inference. By treating the model parameters as random variables and using prior distributions, this approach provides probabilistic outputs that naturally reflect uncertainty. Bayesian methods can offer more reliable probability estimates but might be a strong restriction in the development of the model.

- **Model with Calibration-Targeted Loss:** This method involves training the model with a loss function specifically designed to improve calibration. The loss function combines the standard cross-entropy loss (loss_ce) with additional terms for expected calibration error (loss_ece) and negative log-likelihood (loss_nll). The combined loss function is given by:

  $$\text{loss} = \text{loss\_ce} + \lambda \cdot \text{loss\_ece} + \lambda \cdot \text{loss\_nll}$$

  where λ is a hyperparameter that balances the contribution of the calibration-specific losses. This direct approach ensures that the model's probability estimates are well-calibrated by design, but we should check if this have any repercussion in the accuracy of the model.

- **Ensembling for Calibration:** Ensembling combines predictions from multiple models to improve a large number of metrics, an in our case in theory it should improve the calibration. Techniques like bagging, boosting, or stacking can be used, but we will be using basic stacking of the output models of our other calibration methods. Ensembling reduces variance and often produces better-calibrated probabilities by averaging over multiple models' outputs, mitigating the individual models' overconfidence.

Model Evaluation

The performance of the models will be evaluated using the following parameters:

- **Accuracy:** The proportion of correct predictions made by the model.
- **Expected Calibration Error (ECE):** Measures the difference between predicted probabilities and actual outcomes.
- **Negative Log-Likelihood (NLL):** A metric that quantifies the uncertainty of probabilistic predictions.

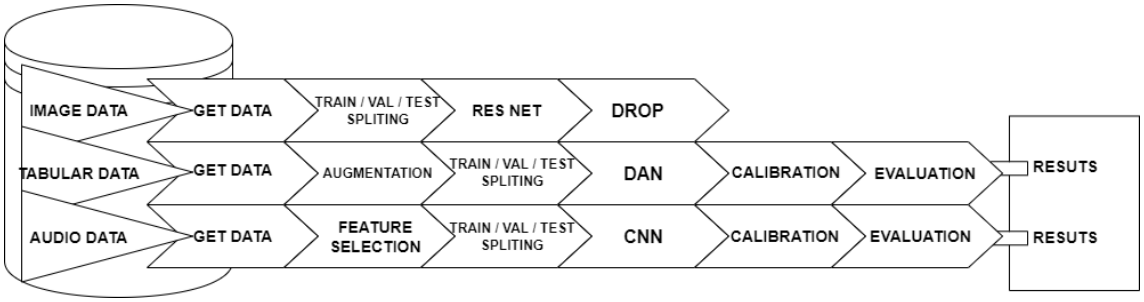

Fig. 4.1. Data pipeline in detail of the model from input to output

4.3. Expansion

Other Calibration Methods

The decision to add other calibration methods, such as Two-Stage Training Calibration Loss, Ensemble methods or Bayesian models, is motivated by the emergence of new calibration methods since the creation of the base study. This serves as a good indicator of how well methods like Temperature Scaling have aged. These new methods provide different approaches to adjusting predicted probabilities and offer alternative ways to achieve calibration.

Other Data Types

To expand the base study, audio data will be used in addition to image and tabular data. The reasons for this addition are:

- **Diversity:** Including audio data increases the diversity of the dataset, allowing the study to explore the effectiveness of calibration methods across unchecked data types to corroborate the effectiveness of these methods.
- **Complexity:** Audio data introduces new challenges and complexities, such as handling temporal dependencies and feature extraction, which can provide insights into the robustness of calibration methods or reveal blind spots that were not found before.

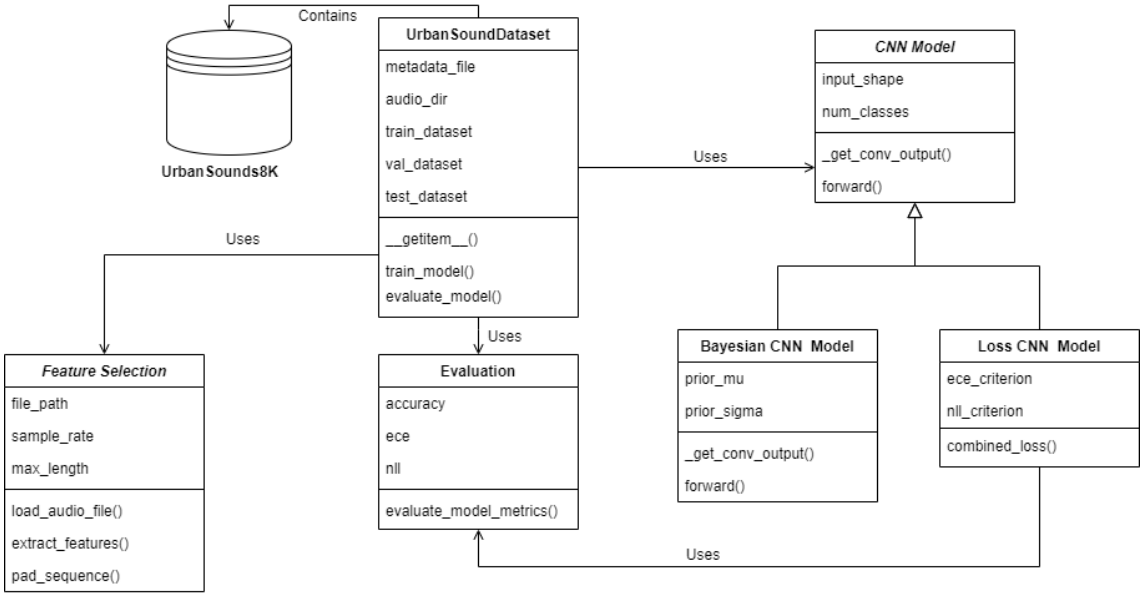

Fig. 4.2. Graph of the classes dedicated to audio

5. IMPLEMENTATION

5.1. System Architecture

Overview

The system architecture is designed to support the calibration of neural network models by integrating various modules that handle data preprocessing, model training, calibration, evaluation, and result visualization. The architecture ensures modularity, scalability, and ease of integration, facilitating the addition of new datasets, models, and calibration techniques.

Modules

The system is divided into the following main modules:

- **Data Preprocessing Module:** This module handles the cleaning, normalization, augmentation, and transformation of raw data, preparing it for training and calibration.
- **Neural Network Module:** This module defines and trains various neural network architectures on the preprocessed data. It includes the implementation of ResNet, DAN, and CNN for audio data.
- **Calibration Module:** This module implements calibration techniques such as Temperature Scaling and Two-Stage Training to adjust the predicted probabilities of the trained models.
- **Evaluation Module:** This module evaluates the performance of both calibrated and uncalibrated models using metrics such as Accuracy, Expected Calibration Error (ECE), and Negative Log-Likelihood (NLL).
- **Result Module:** This module generates visual representations of the model's performance, helping to intuitively understand the effectiveness of the calibration techniques and overall model performance.
- **Testing Module:** As the project follows Test-Driven Development (TDD)[15], this module ensures high test coverage and code quality through comprehensive testing.

This modularization is represented in the Component Diagram shown in the next image:

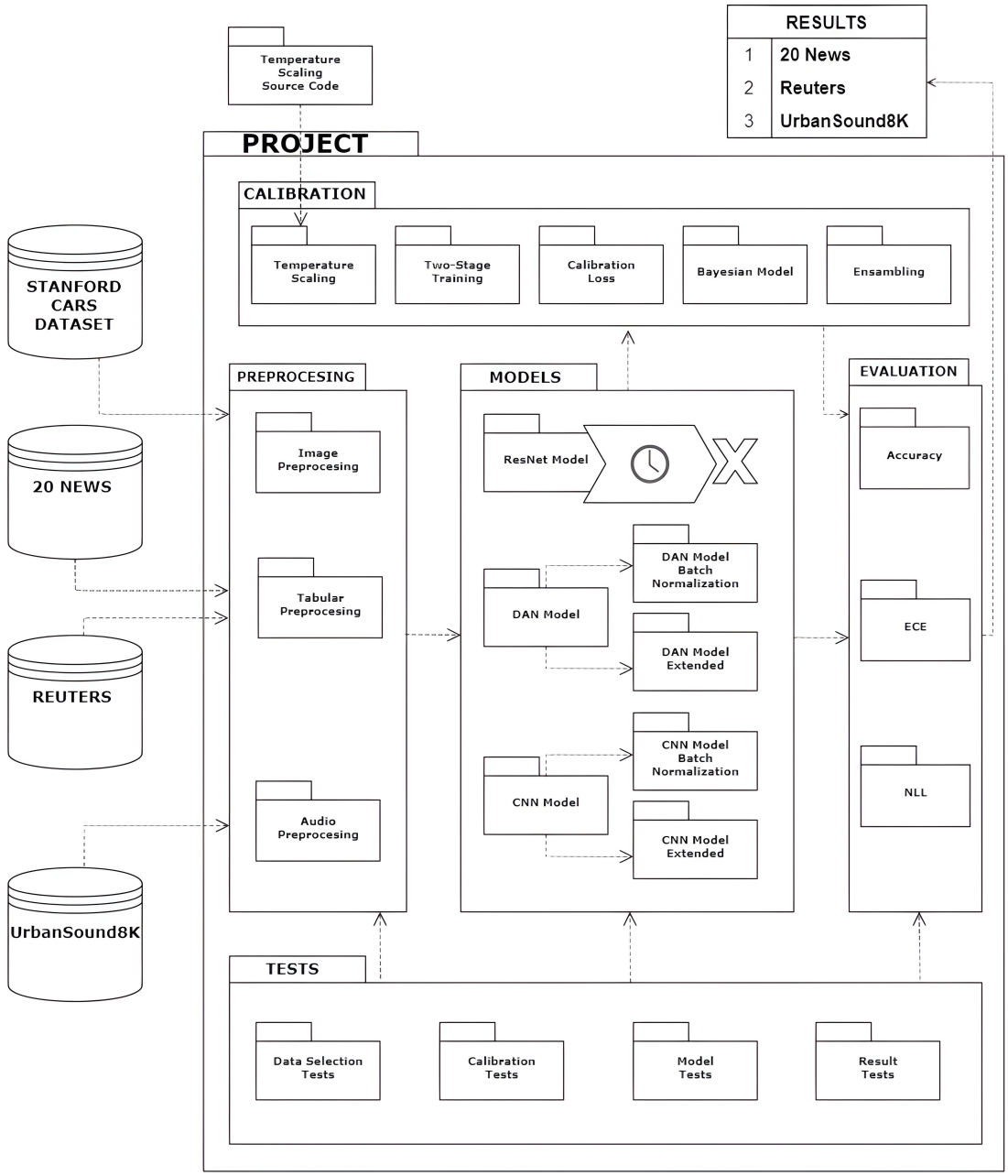

Fig. 5.1. Component Diagram

Workflow

The workflow of the modules is structured to ensure consistent data flow and functionality:

- Data collected from various sources is first processed by the **Data Preprocessing Module**.
- The preprocessed data is then fed into the **Neural Network Module** for model training.
- Trained models are evaluated by the **Evaluation Module** to get baseline performance metrics.
- The trained models are also sent to the **Calibration Module**, where calibration techniques are applied.
- The calibrated models are evaluated again by the **Evaluation Module**.
- Finally, the results are visualized by the **Result Module** for analysis and interpretation.
- Throughout the development process, the **Testing Module** ensures the integrity and quality of the code.

5.2. Development Process

Version Control

The project uses GitHub for version control, managed via command line. This system allows for effective collaboration, tracking of changes, and maintaining different versions of the project. Key practices include:

- Regular commits with descriptive messages.
- Branching for new features and bug fixes.
- Pull requests and code reviews to ensure code quality.
- Merging approved changes into the main branch.

Testing

The development follows a Test-Driven Development (TDD) approach [15], using Spyder IDE for coding and testing. The testing framework ensures that all code is tested thoroughly before integration. Key aspects include:

- Writing tests before implementing features.
- Using unit tests to test individual components and functions.
- Employing integration tests to ensure that different modules work together correctly.
- Continuous testing during development to catch and fix bugs early.
- Ensuring high test coverage to maintain code reliability and performance.

Even though we do this, we will not focus on maximum coverage as that would be overkill.

6. RESULTS

Here, we present the results of our study. We performed the entire pipeline on four different datasets, but we only obtained results for three. The dataset Stanford Cars Dataset, involving images, was too computationally intense for our hardware. Therefore, we employed the control calibration method, temperature scaling, to test and compare it with the other new calibration methods.

We will dispense with histogram binning, isotonic calibration, and most other calibration methods from the main study, except for temperature scaling. This decision is based on the findings of the study by Guo et al. (2017) [1], which highlight the superior performance of temperature scaling compared to other calibration methods.

For brevity and compactness, we will omit these other calibration methods in favor of our own methods to test against temperature scaling.

6.1. Table Accuracy

In this section, we examine the accuracy of our models. A perfectly calibrated model is useless if it compromises the accuracy of the predictions. It's worth noting that we decided to exclude the Bayesian model for audio data since we were unable to achieve performance sufficient for practical use.

Dataset	Base Model	Uncalibrated	Temp. Scaling	TST	Calibration Loss	Bayesian	Ensemble
Tabular Data							
20 News (Base)	DAN 3 256	89.32%	89.32%	89.32%	**89.45%**	80,63%	87.6%
20 News (Batch normal)	DAN 3 256	**90.77%**	**90.77%**	90.53%	90.65%	76.86%	87.92%
20 News (+ Width)	DAN 3 1024	**89.4%**	**89.4%**	89.29%	88.87%	70.95%	85.58%
20 News (+ Depth)	DAN 6 256	70%	70%	72.56%	**74.19%**	62.73%	69.89%
Reuters (Base)	DAN 3 256	35.66%	35.66%	35.35%	**39.63%**	38.51%	36.96%
Reuters (Batch normal)	DAN 3 256	34.82%	34.82%	35.13%	36.2%	**37.98%**	35.79%
Reuters (+ Width)	DAN 3 1024	35.22%	35.22%	**36.69%**	34.02%	35.75%	35.38%
Reuters (+ Depth)	DAN 6 256	**41.14%**	**41.14%**	40.65%	39.89%	39.85%	40.5%
Audio Data							
UrbanSound8K (Base)	CNN 5 128	51.48%	51.48%	56.50%	**65%**	12.74%	59.45%
UrbanSound8K (Batch normal)	CNN 5 128	70.06%	70.06%	75%	**78.13%**	—	73.06%
UrbanSound8K (+ Width)	CNN 5 256	52.49%	52.49%	**59.76%**	46%	—	49.6%
UrbanSound8K (+ Depth)	CNN 7 128	71.49%	71.49%	77.68%	**80.65%**	—	75.63%

TABLE 6.1. TEST ACCURACY (%)

This table has results on NLP datasets and an audio dataset before calibration and with various calibration methods. The first number following a model's name denotes the network depth, and the second number the maximum layer width. Accuracy with temperature scaling is exactly the same as uncalibrated.

Here, we observe that temperature scaling remains effective in maintaining high accuracy since it does not alter the accuracy of the uncalibrated model by definition. Additionally, we notice that our model with the modified loss function surprisingly yields very good accuracies.

6.2. Table ECE

Here we see our main metric to compare how efficient is each method in their job. And as we can see in the table in this case temperature Scaling seems to be outperformed by Calibration Loss and Two Stage Training.

Dataset	Base Model	Uncalibrated	Temp. Scaling	TST	Calibration Loss	Bayesian	Ensemble
Tabular Data							
20 News (Base)	DAN 3 256	1.7%	1.66%	2.3%	**0.94%**	2.49%	1.81%
20 News (Batch normal)	DAN 3 256	2%	1.96%	**1.68%**	1.76%	2.6%	2%
20 News (+ Width)	DAN 3 1024	3.5%	2.67%	**1.96%**	3.68%	4.21%	3.2%
20 News (+ Depth)	DAN 6 256	18.6%	15.6%	15.32%	**15.11%**	17.73%	16.64%
Reuters (Base)	DAN 3 256	17.71%	15.99%	20.69%	**11.56%**	12.69%	15.72%
Reuters (Batch normal)	DAN 3 256	18.8%	15.75%	21.7%	21.08%	**13.12%**	18.09%
Reuters (+ Width)	DAN 3 1024	19.01%	**15.84%**	25.61%	19.63%	16.71%	19.36%
Reuters (+ Depth)	DAN 6 256	9.63%	8.72%	11.22%	**8.3%**	9.21%	9.46%
Audio Data							
UrbanSound8K (Base)	CNN 5 128	6.5%	6.1%	**4.6%**	5%	9.5%	5.55%
UrbanSound8K (Batch normal)	CNN 5 128	3.3%	3.2%	2.56%	**2.32%**	—	2.9%
UrbanSound8K (+ Width)	CNN 5 256	8.2%	**6%**	8.7%	11.5%	—	7.6%
UrbanSound8K (+ Depth)	CNN 7 128	5%	**2.8%**	3.2%	4.7%	—	4.76%

TABLE 6.2. ECE (%) (WITH M = 15 BINS)

This table has results on NLP datasets and an audio dataset before calibration and with various calibration methods. The first number following a model's name denotes the network depth, and the second number the maximum layer width.

6.3. Table NLL

Here, for our second calibration method, we see that the results from the last table were not a fluke, as Calibration Loss consistently outperforms the other calibration models.

Dataset	Base Model	Uncalibrated	Temp. Scaling	TST	Calibration Loss	Bayesian	Ensemble
Tabular Data							
20 News (Base)	DAN 3 256	0.418	0.417	0.3871	**0.3770**	0.7065	0.4611
20 News (Batch normal)	DAN 3 256	0.346	0.3441	0.3571	**0.3379**	0.9212	0.4216
20 News (+ Width)	DAN 3 1024	0.408	0.392	**0.3736**	0.4062	1.2796	0.5825
20 News (+ Depth)	DAN 6 256	**1.244**	1.416	1.3417	1.4236	1.9989	1.473
Reuters (Base)	DAN 3 256	3.3446	3.1267	3.7411	**2.7613**	2.9955	3.261
Reuters (Batch normal)	DAN 3 256	3.8241	3.4636	4.5102	**2.5451**	3.444	3.549
Reuters (+ Width)	DAN 3 1024	3.6111	**3.3233**	4.426	3,5627	3.4822	3.621
Reuters (+ Depth)	DAN 6 256	2.573	**2.5227**	2.7709	2.5337	2.5371	2.557
Audio Data							
UrbanSound8K (Base)	CNN 5 128	1.593	1.587	1.2891	**1.0892**	5.722	1.325
UrbanSound8K (Batch normal)	CNN 5 128	0.821	0.821	0.7326	**0.6557**	—	0.7652
UrbanSound8K (+ Width)	CNN 5 256	1.472	1.44	**1.2550**	1.6702	—	1.593
UrbanSound8K (+ Depth)	CNN 7 128	0.879	0.875	0.6760	**0.6381**	—	0.821

TABLE 6.3. NLL

This table has results on NLP datasets and an audio dataset before calibration and with various calibration methods. The first number following a model's name denotes the network depth, and the second number the maximum layer width. To summarize, NLL roughly follows the trends of ECE.

6.4. Interpretation of Results

Within the computational limitations we have faced, we can draw some conclusions based on the interpretation of our results:

- Confirmation of Previous Study Findings: Temperature scaling indeed appears effective across various applications, providing moderate calibration improvement at minimal cost.
- Confirmation of Modern Model Findings: Our tables for each dataset feature different models adjusting key parameters that contribute to poor calibration in modern models. Generally, it holds true that these parameters significantly impact calibration, often beyond full correction by any calibration method. However, for mediocre base models like Reuters(+Depth) or UrbanSound8K(Batch normal), adjustments still yield benefits.
- Unsuccessful Alternatives: Despite proposing alternative calibration methods alongside proven temperature scaling, our results are mixed. Bayesian models theoretically ideal for calibration struggle practically, being overly restrictive. The Ensemble method, incorporating other methods, fails to reach its potential, likely affected by poor Bayesian results.
- Successful Alternatives: Two Stage Training and the custom Loss function model emerge as straightforward yet effective calibration alternatives, achieving results comparable to or better than temperature scaling. Particularly surprising is Calibration Loss, which excels in calibration without sacrificing accuracy, contrary to theoretical expectations.

7. PLANNING

7.1. Development Tools

Software Tools

The following software tools are used for development:

- **Python:** The primary programming language for implementing models, data preprocessing, and evaluation.
- **TensorFlow/Keras:** Used for building and training neural network models.
- **NumPy and Pandas:** Essential for data manipulation and analysis.
- **Jupyter Notebook:** For interactive coding, testing, and documentation.
- **Spyder IDE:** Used for development and testing with Test-Driven Development (TDD).
- **Git:** Version control system used via command line.
- **GitHub:** For repository hosting, collaboration, and version management. https://github.com/javiman555/TFM-Data-Science

Hardware Requirements

The project requires the following hardware:

- **High-Performance CPU:** Necessary for general computation and data processing tasks when the models are too big to handle in the GPU.
- **GPU:** Required for efficient training of deep learning models and powered by CUDA.
- **RAM:** 16 GB of RAM for handling large datasets and models.
- **Storage:** Sufficient SSD storage for datasets, models, and software. Some internal cleaning needed to be done for this to be the case.

Testing Frameworks

The following testing frameworks are utilized:

- **spyder-unittest:** Python's built-in testing framework used for unit and integration testing applied to the Spyder IDE.[21]

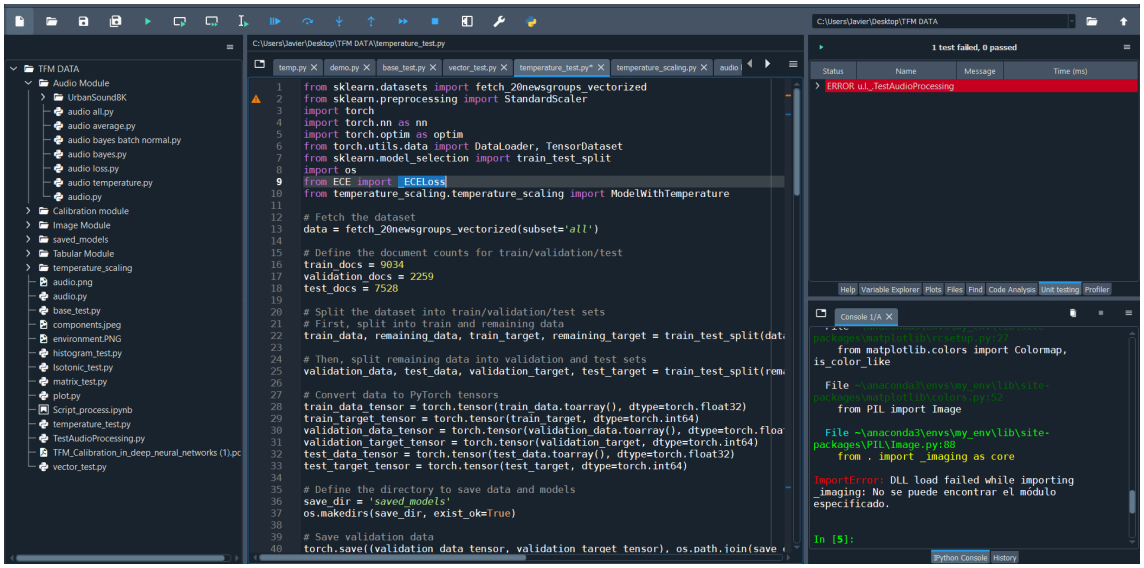

Fig. 7.1. Working environment during development

Project Management Tools

The project management tools used are:

- **Trello:** For task management and tracking project progress.
- **Google Drive:** For external storage and first drafts to development.

7.2. External Checkers

Thesis Checkers

As part of ensuring the quality and accuracy of the thesis, several tools and strategies were employed:

- **Grammar and Spelling Checks:** To maintain high standards of grammar and spelling throughout the document, www.grammarcheck.net was used extensively. This tool helped identify and correct grammatical errors, punctuation issues, and spelling mistakes, ensuring the text is clear and professional.
- **Reference Management:** Proper citation and bibliography formatting were managed using the intrinsic LaTeX external bibliography support. These tools helped in organizing references and ensuring they adhered to the required citation style.
- **Formatting Checkers:** To ensure the document adhered to the required formatting guidelines, tools like LaTeX, specifically Overleaf, were used for document preparation, and various templates were applied to maintain consistency in style and structure.
- **Automated Proofreading:** Online automated tools like ChatGPT were used to conduct a final review of the document, catching any lingering issues that might have been overlooked during manual proofreading.

These tools and methods collectively ensured that this thesis is polished, well-structured, and free of errors.

7.3. Time Spent

Timeline

The timeline of the project is as follows:

- **Month 1:** Project planning and setup, including defining requirements and initial data collection.
- **Month 2:** Model training and calibration, along with continuous integration and testing.
- **Month 3:** Final evaluation, result analysis, and documentation.

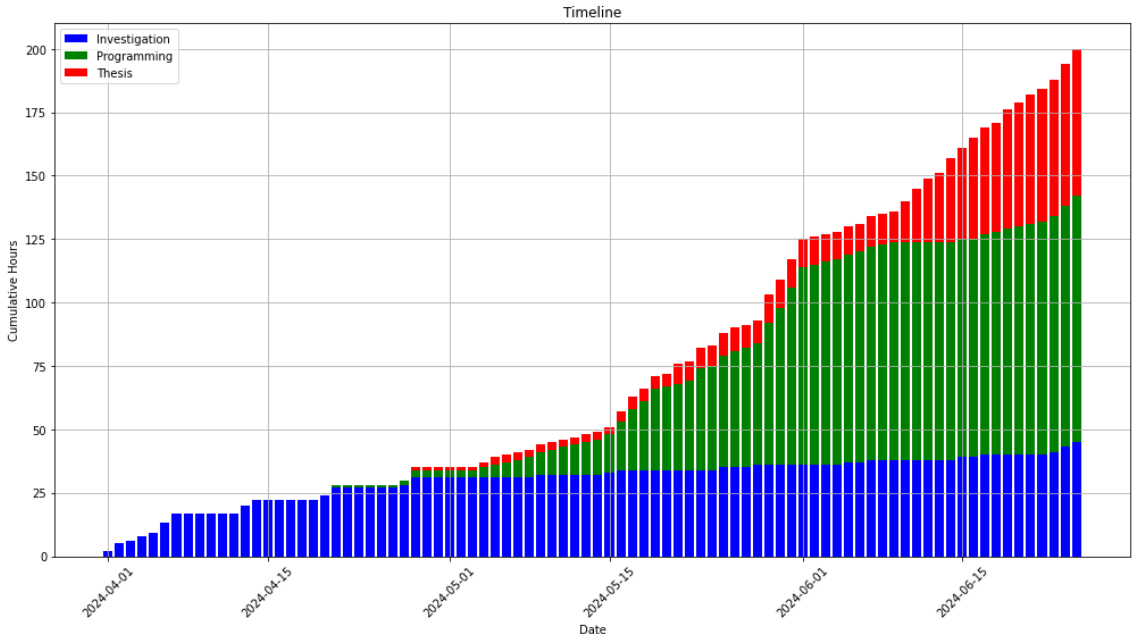

Fig. 7.2. Timeline

Reflection

Reflecting on the research process:

- The iterative development process allowed for continuous improvement and adaptation to new findings.
- Effective use of project management tools facilitated smooth progress and timely completion of milestones.
- The integration of external checkers and adherence to best practices ensured high code quality and reliable results.

Lessons Learned

The lessons learned during the research include:

- **Importance of Modular Design:** Designing the system in a modular way made it easier to manage complexity and integrate new components.
- **Value of TDD:** Test-Driven Development significantly improved code quality and reduced bugs. Initially, it may slow down development, but dealing with environment issues early on allowed us to quickly verify everything was working correctly, ultimately saving us a lot of time.
- **Adaptability to New Techniques:** Being open to incorporating new methods and tools enhanced the overall effectiveness of the project. If we hadn't believed in the promise of the calibration loss method and had dismissed it as impossible to work with, we would not have discovered what appears to be a very promising calibration method.

8. CONCLUSION

8.1. Objectives Met

8.1.1. Evaluation

Throughout the project, several challenges arose, primarily due to hardware limitations that made it impossible to confirm results for image datasets. However, this issue did not significantly impact the main objective, as the primary model (temperature scaling) was successfully validated for both tabular and audio data. Additionally, we thoroughly examined how key parameters of modern models affect calibration and their specific impacts. Overall, the project successfully achieved its objectives.

8.1.2. Achievements

In addition to confirming previous studies, the research achieved the following key goals:

- Successful expansion of calibration models to include an audio database.
- Identification of alternative calibration methods that are as effective as, or even better than, temperature scaling.

8.1.3. Challenges Faced

The research encountered several challenges:

- Computational power and storage limitations made it impossible to work with image datasets and challenging to collect results for augmented models (+Width / +Depth) due to extended training times. Some models required over an hour to train under these conditions.
- Issues with Python dependencies and compatibility, exacerbated by the fragile nature of CUDA compatibility with GPUs. A misstep in environment setup could lead to a broken configuration, rendering CUDA-based code inoperable and requiring the creation of a new environment to resume work.

8.2. Future Work

8.2.1. Improvements

The main improvement we should make is to implement cross-validation for calculating calibration metrics. Cross-validation can provide more robust estimates of model performance and calibration across different subsets of data, enhancing the reliability of our findings. This would address any uncertainties we have regarding the outstanding results of the Calibration Loss method. Additionally, we should reevaluate the Ensemble method using only high-performing models as input.

8.2.2. Image Calibration

Currently, we were unable to conduct calibration experiments on image datasets due to hardware limitations. The computational demands for processing image data exceeded the capabilities of our current hardware setup. This limitation prevented us from confirming the applicability of calibration techniques to image datasets within the scope of this study.

8.2.3. Next Steps

Moving forward, the next step for our research should be to focus on methods that integrate calibration directly inside the models. By embedding calibration mechanisms within the model architecture itself, rather than applying them as post-processing steps, models can potentially learn to produce well-calibrated predictions during training. This approach could streamline the calibration process and improve overall model performance in various real-world applications. While we have explored several effective methods with this philosophy, we have yet to investigate the more invasive Self-Calibration Techniques, which involve adding calibration layers to normal models or modifying the training outputs in real time.

BIBLIOGRAPHY

[1] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, *On calibration of modern neural networks*, 2017. arXiv: 1706.04599 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1706.04599.

[2] M. Jordahn and P. M. Olmos, *Decoupling feature extraction and classification layers for calibrated neural networks*, 2024. arXiv: 2405.01196 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2405.01196.

[3] O. Evans et al., *Truthful ai: Developing and governing ai that does not lie*, 2021. arXiv: 2110.06674 [cs.CY]. [Online]. Available: https://arxiv.org/abs/2110.06674.

[4] S. Lin, J. Hilton, and O. Evans, *Truthfulqa: Measuring how models mimic human falsehoods*, 2022. arXiv: 2109.07958 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2109.07958.

[5] G. Pleiss, *Temperature$_s$caling*, Accessed: 2024-06-26, 2017. [Online]. Available: https://github.com/gpleiss/temperature_scaling.

[6] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: 1512.03385 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1512.03385.

[7] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. III, "Deep unordered composition rivals syntactic methods for text classification," Jan. 2015, pp. 1681–1691. DOI: 10.3115/v1/P15-1162.

[8] F.-F. Li, J. Johnson, and S. Yeung, *Cs231n convolutional neural networks for visual recognition*, https://cs231n.github.io/convolutional-networks/, Accessed: 2024-06-26, 2023.

[9] J. Macfarlane and M. Stroila, "Addressing the uncertainties in autonomous driving," *SIGSPATIAL Special*, vol. 8, no. 2, pp. 35–40, Dec. 2016. DOI: 10.1145/3024087.3024092. [Online]. Available: https://doi.org/10.1145/3024087.3024092.

[10] V. E. Staartjes and J. M. Kernbach, "Letter to the editor. importance of calibration assessment in machine learning–based predictive analytics," *Journal of Neurosurgery: Spine SPI*, vol. 32, no. 6, pp. 985–987, 2020. DOI: 10.3171/2019.12.SPINE191503. [Online]. Available: https://thejns.org/spine/view/journals/j-neurosurg-spine/32/6/article-p985.xml.

[11] J. Zhang, B. Kailkhura, and T. Y.-J. Han, "Mix-n-match : Ensemble and compositional methods for uncertainty calibration in deep learning," in *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, Eds., ser. Proceedings of Machine Learning Research, vol. 119, PMLR, 13–18 Jul 2020, pp. 11 117–11 128. [Online]. Available: https://proceedings.mlr.press/v119/zhang20k.html.

[12] B. F. I. David N. DeJong and C. H. Whiteman, "A bayesian approach to calibration," *Journal of Business & Economic Statistics*, vol. 14, no. 1, pp. 1–9, 1996. DOI: 10.1080/07350015.1996.10524625. eprint: https://www.tandfonline.com/doi/pdf/10.1080/07350015.1996.10524625. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/07350015.1996.10524625.

[13] A. S. Mozafari, H. S. Gomes, S. Janny, and C. Gagné, "A new loss function for temperature scaling to have better calibrated deep networks," *arXiv preprint arXiv:1810.11586*, 2018.

[14] J.-J. Liu, Q. Hou, M.-M. Cheng, C. Wang, and J. Feng, "Improving convolutional networks with self-calibrated convolutions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[15] M. M. Moe, "Comparative study of test-driven development tdd, behavior-driven development bdd and acceptance test–driven development atdd," *International Journal of Trend in Scientific Research and Development*, vol. 3, no. 4, pp. 231–234, 2019.

[16] J. Li, *Stanford cars dataset*, 2018. [Online]. Available: https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset/data.

[17] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[18] scikit-learn developers, *The twenty newsgroups dataset*, 2024. [Online]. Available: https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html.

[19] NLTK Contributors, *Reuters corpus*, https://www.kaggle.com/datasets/nltkdata/reuters, Accessed: 2024-06-26, 2024.

[20] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *22nd ACM International Conference on Multimedia (ACM-MM'14)*, Orlando, FL, USA, Nov. 2014, pp. 1041–1044.

[21] Spyder IDE Developers, *Spyder unittest plugin*, https://github.com/spyder-ide/spyder-unittest, Accessed: 2024-06-26, 2024.

[22] Anaconda Software Distribution, *Anaconda*, version 2-2.4.0. [Online]. Available: https://anaconda.com.

[23] Spyder IDE Developers, *Spyder*, Jun. 26, 2024. [Online]. Available: https://www.spyder-ide.org/.