UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Informatics

Application of probabilistic modeling to predict road pavement deterioration

End of Degree Project

Bachelor's Degree in Data Science

AUTHOR: Viana Llovell, Sergio

Tutor: Morillas Gómez, Samuel

Experimental director: Oliveira Dos Santos, Joao Miguel

ACADEMIC YEAR: 2023/2024

Resumen

Un problema importante en el campo de la Ingeniería Civil se refiere al mantenimiento de los pavimentos en la red de transporte. El deterioro de los pavimentos puede medirse de diferentes formas, siendo una de ellas el denominado Índice de Rugosidad Internacional (IRI). Con el fin de planificar de forma óptima las estrategias de mantenimiento de pavimentos, la predicción de la evolución del IRI a lo largo del tiempo sería altamente beneficiosa. En este proyecto proponemos utilizar modelos probabilísticos, concretamente Redes Neuronales Bayesianas, para aproximarnos a este objetivo. Para ello, utilizaremos una base de datos de registros de IRI de la red de carreteras de los Países Bajos que registra la evolución del IRI a lo largo de los años. Esta base de datos, una vez tratada, combinada con datos de tráfico y climatológicos, será la base para los modelos de ML que se entrenarán.

Palabras clave: IRI, Red Neuronal Bayesiana, red de carreteras de los Países Bajos.

Abstract

An important problem in the Civil Engineering field concerns the maintenance of pavement in the transportation network. Pavement deterioration can be measured in diverse ways, being one of them the so-called International Roughness Index (IRI). In order to optimally plan pavement maintenance strategies, the prediction of IRI evolution over time would be highly beneficial. In this project we propose to use probabilistic models, specifically Bayesian Neural Networks, to approach this goal. To do so, we will use a database of IRI records from the Netherlands road network that register the evolution of IRI over the years. This database when treated, combined with traffic and climatological data will be the basis for the ML models that will be trained.

Keywords: IRI, Bayesian Neural Network, Netherlands road network.

Index

Illustration Index

1. Introduction

As an essential component of transportation engineering, pavement performance assessment has a direct impact on road network upkeep, safety, and serviceability. Long-term pavement life, effective resource allocation, and increased user happiness are all guaranteed by efficient pavement management. Understanding and forecasting pavement roughness, which has an immediate influence on ride quality, fuel economy, and vehicle wear and tear, is essential to this assessment. The International Roughness Index (IRI), one of the several indices used to quantify pavement roughness, is the most extensively used worldwide [1].

Developed in the 1980s by the World Bank, the International Roughness Index (IRI) quantifies pavement surface irregularities by measuring the vertical displacement of a vehicle's suspension system over a standardized distance [1]. Expressed in meters per kilometer (m/km), the IRI provides a numerical value representing the roughness of the pavement, which is essential for [1]:

1. **Performance Monitoring**: Continuously assessing the deterioration of pavement over time to plan timely maintenance.
2. **Maintenance and Rehabilitation Planning**: Identifying critical sections of road networks that require intervention.
3. **Budget Allocation**: Prioritizing funding based on the condition of the pavement, ensuring efficient use of resources.
4. **Quality Control**: Evaluating the effectiveness of construction practices by comparing post-construction IRI values against predefined standards.

The IRI is measured in meters per kilometer (m/km) or inches per mile (in/mi), and the values typically fall within certain ranges that correlate with the condition of the pavement. The following table shows a classification per range values:

Measure	Sealed road				
Pavement condition	Very Poor	Poor	Fair	Good	Very Good
International Roughness Index (IRI)	8+	6-7	4-5	3	0-2

Figure 1: IRI condition rating scale for sealed roads.

Source: National Association of Australian State Road Authorities.

In order to help maintaining the quality of the road transportation system, this project has been created to investigate whether is possible to improve the forecasting of the IRI using ML techniques, specifically with probabilistic methods. To do so, a Bayesian Neural Network (BNN), along a linear regression for explainability purposes, has been trained using an IRI database of the Netherlands roads.

1.1. Motivation

My math teacher Samuel Morillas, along with his colleague Joao Oliveira, a professor and civil engineer at the University of Twente in Enschede, served as the inspiration for my project. After completing my Erasmus program at the University of Twente, I thought this would be a great chance to get in touch with Professor Oliveira and work together.

Beyond these coincidences, the project's focus is on machine learning applications, a topic that has interested me throughout my studies at the UPV. The research specifically deals with BNNs, which is a subject I am excited to learn more about because it is a variant of the traditional Neural Network, which we had not studied at this degree. This research presents a good opportunity for me to learn more about probabilistic models and advance my machine learning skills.

1.2. Objectives

This project is centered around three primary objectives, each designed to systematically address the problem of predicting the International Roughness Index (IRI) using machine learning techniques:

The first objective is to create a comprehensive dataset by integrating our data with public datasets, including IRI, KNMI, and INWEVA data. This integrated dataset will serve as the foundation for training machine learning models and will enable the exploration of the relationships between various features and IRI Increase.

The second objective is to develop a probabilistic machine learning model, specifically a Bayesian Neural Network (BNN). The BNN will be trained on the complete dataset to generate predictions of IRI Increase. The performance of the BNN will be rigorously evaluated to determine its effectiveness in capturing the complex, non-linear relationships present in the data.

The third objective is to compare the predictive power of the BNN with that of a simpler Linear Regression (LR) model. Additionally, the LR model will be employed to help explain the behavior of the BNN by revealing underlying linear relationships within the data. While the LR model may not capture non-linear relationships as effectively, it will provide valuable insights into the general data patterns and contribute to a deeper understanding of the factors influencing IRI Increase.

1.3. Methodology

This project follows a systematic approach grounded in the scientific method to address the problem of predicting the International Roughness Index (IRI) for specific road segments. The methodology is organized around the formulation of several hypotheses, the implementation of models to test these hypotheses, and the evaluation of the outcomes against the initial objectives.

The first step in our methodology involved formulating a clear hypothesis: <u>The IRI of a road segment can be predicted accurately by integrating public datasets with features such as weather conditions and pavement quality</u>. This hypothesis guided the development of our machine learning models, including the Bayesian Neural Network (BNN) and linear regression models.

Additional hypotheses were formulated to explore different aspects of the problem:

- **Hypothesis 1**: The integration of multiple datasets (e.g., IRI, KNMI, and INWEVA) can create a comprehensive dataset that effectively represents the factors influencing IRI.
- **Hypothesis 2**: A probabilistic model like the BNN can provide more reliable predictions of IRI Increase compared to a traditional linear regression model.

To test our hypotheses, various public datasets were integrated with the IRI dataset. This included weather conditions from KNMI, and additional environmental data from INWEVA. Python was used extensively for data preprocessing, cleaning, and merging to ensure a complete and usable dataset.

With the dataset prepared, the next step was the model development phase:

- **Bayesian Neural Network (BNN)**: The BNN was developed as our primary model for predicting IRI Increase. The model was trained on the integrated dataset, allowing it to capture complex, non-linear relationships between the features and the IRI.
- **Linear Regression Model**: A simpler linear regression model was also developed to serve as a baseline for comparison and to help explain the behavior of the BNN. This model aimed to highlight any linear relationships within the dataset.

Both models were trained and tested using a train-test split method, with the BNN's performance being the primary focus. The linear regression model provided insights into the factors that had the most significant impact on the BNN's predictions.

Once the models were trained, the hypotheses were tested by evaluating the performance of both the BNN and the linear regression model:

- **Hypothesis 1 Evaluation**: The success of data integration was measured by the completeness and usability of the dataset, as well as the performance of the models trained on this data.
- **Hypothesis 2 Evaluation**: The BNN’s predictive performance was assessed using various metrics (e.g., correlation, mean squared error) to determine its effectiveness in predicting IRI Increase.

The results obtained from these evaluations provided insights into the validity of our hypotheses and the effectiveness of the models developed.

1.4. Structure

The structure of this project will be the following: firstly, the data treatment and the final dataset will be explained, where the reader will understand how the data has been gathered and joined, as well as the meaning of each variable. Then, a detailed explanation of what is a BNN will take place, along with the algorithms it uses to make its predictions. Finally, the results obtained with the models will be shown. These results will gather the explanatory graphics created by the LR and the predictions achieved using the BNN.

2. State of the Art

The prediction and modeling of IRI are vital for proactive pavement management so several approaches have been developed, evolving from empirical methods to advanced machine learning techniques. Initially, empirical models relied heavily on historical data and straightforward statistical methods. Over the years, these models have become more complex, incorporating a broader range of variables and advanced statistical techniques to improve accuracy and reliability [2].

Empirical methods rely on historical data to establish relationships between IRI and various influencing factors such as traffic load, pavement age, and environmental conditions. These are classified into three main categories [2]:

- **Surface Characteristics-Based Models**: These models focus on surface distress indicators like roughness, rut depth, and cracking. For example, the Indiana Department of Transportation (INDOT) developed prediction models using roughness data to plan rehabilitation needs.
- **Environmental Factor-Based Models**: These models assess the impact of environmental conditions on pavement performance. They incorporate variables such as temperature, moisture, and freeze-thaw cycles to predict pavement deterioration.
- **Pavement Performance Rating Models**: These models use composite indices like the Present Serviceability Index (PSI) and the Pavement Condition Index (PCI) to rate pavement performance based on various characteristics.

Recent advancements have seen the integration of machine learning (ML) techniques in pavement condition evaluation, significantly enhancing the predictive capabilities and efficiency of pavement management systems (PMS). ML models can analyze vast amounts of data from various sources, including digital photography, GPR, laser scanning, optic fiber sensors, vibration analysis, acoustic emission, and deflection testing to collect comprehensive data on pavement conditions [3].

Convolutional Neural Networks (CNN) are particularly effective for analyzing images of pavement surfaces, detecting and classifying surface distresses with high accuracy. Reinforcement Learning (RL) is used to develop optimal maintenance strategies by simulating various scenarios and learning the most effective actions to preserve pavement health. When combined, these models increase predicted accuracy, allowing for proactive maintenance and more efficient use of resources, which eventually reduces costs and improves road safety [3].

Other ML algorithms that are commonly used include Support Vector Machines (SVM) or Random Forests, which classify pavement conditions based on features like surface roughness and cracking, and which can handle the joining of this features with large datasets considering factors like traffic load and environmental conditions. Naïve Bayes classifiers quickly assess pavement health by categorizing conditions using historical data.

Neural Networks (NN) [4] model complex relationships between variables such as traffic and weather to predict future performance [3]. They are described as powerful function approximators. Its performance relies on the extreme flexibility associated with having many model parameters (weights and biases), whose values can be learned from data through gradient-based optimization [5]. They are good at approximating functions (input-output relationships) with massive amounts of data. Therefore, neural networks are well-suited to artificial intelligence tasks like speech recognition and image classification. For our use case, NN are also an important approach to consider; as they are widely used for classification task, they can be adapted to perform regressions. Nevertheless, the extreme flexibility of NNs has a downside: they are particularly vulnerable to overfitting. Overfitting [6] happens when the learning algorithm does such a good job of tuning the model parameters for performance on the training set—by optimizing its objective function—that the performance on new examples suffers.

Deep Neural Networks (DNNs) [7] suffer especially from overfitting and face a challenge known as the vanishing gradient problem [8] that is pronounced due to their architecture, which involves millions of parameters. In the training process, these parameters are adjusted iteratively using optimization algorithms such as gradient descent [5] and backpropagation [9]. The vanishing gradient problem refers to the decreasing impact of gradient updates on certain parameters during training. As the algorithm passes through the layers of the deep neural network, the gradient can become very small and approach zero. As a result, there are few updates of the corresponding parameters, which prevents effective learning of these specific features and patterns. A fundamental part of the backpropagation is the chain rule [10], which multiplies gradients as it moves backward through the network layers. If the gradients are small, the multiplication results in extremely small values. As a result, the model struggles to capture long-range dependencies and correlations between input features, leading to suboptimal performance.

Bayesian Neural Networks (BNNs) could potentially mitigate some of the issues associated with traditional neural networks, particularly the overfitting. The probabilistic framework, used by BNNs, treats the model parameters, such as weights and biases, as distributions as opposed to fixed values. By taking into account a range of reasonable values for the parameters rather than a single point estimate, this method enables the network to quantify uncertainty in the model parameters, which can help prevent overfitting [11]. It also allows the incorporation of prior knowledge about the weights and biases into the model, while updating the parameters as data is observed. The formula that expresses the training phase of a BNN would be:

$$\theta \sim N(\mu, \Sigma) \quad \theta_0 \sim N(0, I)$$

$$\mu^*, \Sigma^* = \arg arg \sum_{(x_i, y_i) \in D_{tr}} L\left(F_\theta\left(x_i\right), y_i\right) + KL\left[p(\theta), p\left(\theta_0\right)\right]$$

The expression indicates the objective is to find the values of *μ* and *Σ* that will minimize the total objective function, which is the sum of the *data likelihood* term $L\left(F_\theta\left(x_i\right), y_i\right)$ and the *KL divergence* $KL\left[p(\theta), p\left(\theta_0\right)\right]$, which acts as a regularization term, providing stochasticity or variation in a neural network. *KL* divergence finds the distribute distance between two distributions, and in this equation, it is preventing the distribution of θ becoming too different from the normal distribution. Moreover, knowing the posterior distribution of the parameters allows to do probabilistic predictions by expressing a distribution over predicted output $\hat{y}$ as a function of the new input *x.* A mathematical representation of the inference equation would be [12]:

$$p(x, D) = \int p(x, \theta) p(D) \, d\theta$$

The predictive distribution $p(x, D)$ returns a distribution that represents the prediction's uncertainty rather than a single expected number. It can be obtained calculating the integral of the test likelihood $p(x, \theta)$ (it represents which is the probability of the observed output y, given the input x and a set of parameters θ) under the posterior distribution $p(D)$ (it represents the updated distribution of the parameters θ after observing the data *D*). Essentially, the integral combines these two distributions averaging predictions over all possible values of θ weighted by their posterior probability. This, integral can be computationally intensive due to the high dimensionality the problems where BNNs are applied have. Therefore, various sampling methods such as Monte Carlo integration, variational inference, or Markov Chain Monte Carlo (MCMC) are used to estimate the integral. For example, the distribution could be approximated using a finite number of Monte Carlo samples from the posterior probability [12].

Despite these advantages, the complexity of Bayesian inference and the computational intensity required for training BNNs may have limited their application in pavement performance projects. Consequently, there is significant potential for future research to explore and integrate BNNs into this field, potentially leading to more reliable and insightful pavement condition assessments.

In the future, the incorporation of real-time data from Internet of Things devices is expected to transform pavement performance assessment by facilitating ongoing observation and more accurate forecasting. Furthermore, improvements in sensor technology, including 3D imaging and laser scanners, should yield even more precise and in-depth information about the state of the pavement, improving the predictive power of pavement management systems [13].

2.1. Proposal

In this project, probabilistic models will be covered, with a focus on the machine learning technique known as Bayesian Neural Networks (BNNs), using the Markov Chain Monte Carlo (MCMC) sampling method for the training phase. As mentioned earlier, some approaches are taken using Naïve Bayes, while others use NN, which are admittedly powerful, but in this occasion, a model that combined both approaches was selected.

This project will train the BNN utilizing empirical data generated from surface characteristics, environmental parameters, and traffic density (all of which have a substantial impact on road conditions) as opposed to some other efforts that use image-based training. This approach leverages the strengths of both empirical data and advanced machine learning techniques to provide more accurate and reliable predictions of pavement conditions.

3. Problem Analysis

Governments and civil engineers face a significant challenge in preventing road deterioration and maintaining the quality of road networks. This challenge is compounded by several factors, including the traffic loads on the roads, the environmental conditions, the aging infrastructures, the budget constraints and the difficult data gathering.

Given these factors, millions of dollars are being invested annually to mitigate road deterioration and maintain road quality. However, the current methods employed often fall short in addressing these challenges comprehensively and efficiently.

Pavement management systems (PMS) innovation and business prospects must be systematically analyzed in order to properly solve these difficulties. A SWOT analysis will be used in this situation [14]:

- **Strengths**: Advanced modeling techniques like Bayesian Neural Networks (BNNs) and machine learning can provide accurate predictions of pavement conditions, enabling a quicker maintenance.
- **Weaknesses**: High initial setup costs and the need for extensive data collection and processing can be barriers to adoption.
- **Opportunities**: Pavement management could promote a transformation with the integration of IoT devices and real-time data collection. Collaborations with IT companies and sensor producers can improve the predictive and accurate qualities of data.
- **Threats**: Rapid technological changes and budget constraints can impact long-term project sustainability and scalability.

To complete the analysis, a review of the legal and ethical framework is mandatory. Several aspects will be discussed below:

- **Data Protection Analysis:** All the data obtained and used is publicly available on the internet, as it is generated from official organizations. Therefore, no especial treatment must be committed to ensuring that the data is protected.
- **Intellectual Property**: The models created for these duties can be coded using open-source software like python, thus no specific licenses must be bought, nor sensitive software should be hidden.

3.1. Identification and analysis of possible solutions

There are many open strategies that try to solve this problem. The following are possible solutions that are related to the field of study lectured in the degree of Data Science:

- **Predictive Maintenance Systems**: Development of advanced predictive maintenance systems using machine learning techniques to forecast pavement deterioration and plan periodic interventions.
- **IoT Integration**: Incorporating IoT devices for real-time monitoring of road conditions. Sensors embedded in the pavement and vehicle-based data collection systems can provide continuous updates on pavement health.

- **Automated Data Collection and Analysis:** Utilizing UAVs, 3D imaging, and laser scanning technologies for detailed and automated data collection, reducing the need for manual inspections and improving data accuracy [3].

Although more approaches could be taken into consideration, these three key points summarize well where are the efforts being put forward.

3.2. Proposed solution

Regarding the previous solutions commented, the focus of this project will aim to develop a predictive maintenance system. As commented earlier, they rely on machine learning techniques, which are well suited to forecast the IRI, thus providing civil engineers with insights into how road deterioration may progress. The choice of a Bayesian Neural Network (BNN) is particularly appropriate because it accounts for intrinsic variability such as weather conditions, material properties, and traffic patterns. In essence, BNNs are well-equipped to handle the inherent uncertainties in pavement performance, offering a more robust and reliable approach to predicting road maintenance needs.

4. Data Preparation and Analysis

This project has relied mainly on three different data sources, each of which has been treated and prepared independently so that they can be joined together to work on a complete database. The datasets are:

1. The IRI dataset, that has been provided by my cotutor Joao. It contains information about the IRI over the road network of the Netherlands.
2. The Traffic Density per Road Segment (INWEVA in Dutch, I'll use it for abbreviation) dataset of the Netherlands, that I have obtained from the *Rijkswaterstaat Data Registry*.
3. The Environmental Conditions dataset, that I have obtained from the *Koninklijke Nederlands Meteorologisch Instituut (KNMI).* It contains information about several climatological conditions per region of the Netherlands.

To fully comprehend the datasets, a detailed explanation of each dataset and the transformations it has undergone is essential, being Python the programming code used to process all requirements. The same methodology will be applied to all three datasets: initially, a brief description of some key features within the dataset will be provided; secondly, the data wrangling performed will be outlined, converting it to a usable version; and finally, the features remaining after the previous process will be explained in detail.

4.1. IRI Dataset

As previously commented, this dataset contains the IRI per road segment over the Netherlands' Road System. Each row of the dataset represents a road segment which is identified by a key that is called *BPS3*.

Here is an example of the key: *WNN_10_1HRL_2R-L_21.0_21.1.* Its break down would be:

- **WNN**: the region where the road segment is located.
- **10**: the ID of the road set by *Rijkswaterstaat.*
- **1HRL**: the last letter is the representative; R means right side of the road, and L means left side.
- **2R-L**: the number indicates the lane, in case of several lanes per side.
- **21.0**: indicates the kilometer where the road begins.
- **21.1**: indicates the kilometer where the road finishes.

Having commented on the several values of the key, most of the relevant columns of the IRI dataset have been mentioned. Ultimately, these features are:

- **Region**: String, the region where the road segment is located. One-hot encoding is applied.
- **ID**: Integer, the ID of the road segment.
- **Side**: String, the side of the road (left or right). One-hot encoding is applied.
- **Lane**: String, the lane inside the side of the road. One-hot encoding is applied.

- **From**: Float, the starting km of the road segment.
- **To**: Float, the ending km of the road segment.

The columns which form the complete dataset are the previously remarked, that form the key, plus a few more:

- **Surface Layer Type**: String, the type of the road's surface. One-hot encoding is applied.
- **IRI**: Float, the IRI of the road segment.
- **Measurement Date**: Date, the date when the IRI was measured.
- **Construction Date**: Date, the date when the road segment was constructed or repaired.
- **Age**: Float, the years that passed since the construction date until the measurement date.

Now that the features have been explained, it would be helpful to recapitulate the modifications made to the dataset. Initially, the dataset was not suitable for integration with the other two datasets or for input into a machine learning model. First, minor modifications were performed: the removal of negative values from the Age column and the addition of the construction date to the key. Negative Age values were essentially noise due to data wrangling errors and should not be passed to the model. Additionally, the inclusion of the construction date was necessary as it indicated whether a road segment had been repaired. A repaired segment was considered a newly constructed segment, with its IRI progression effectively restarting. For modeling coherence, such segments should be treated as new. The best way to do this was to include the construction date in the key, thereby creating a new road segment each time it had been repaired. The new key would look like this: WNN_10_1HRL_2R-L_21.0_21.1_2015.

Following this, it was mandatory to remove the duplicated rows in the dataset. Although each row had a key to be uniquely identified, there was still one more column in the dataset that was causing problems: the *Original Version Number*. During previous modifications to this dataset, multiple versions of the same segment were being stored, each time with an increasing value of the *Original Version Number* column, thus accumulating duplicated rows. To solve this problem, it was necessary to create a loop that checked every row. The methodology was:

1. Iteration over different tuples of (*Age, BPS3*). The duplicates were on the age level, i.e. there were several identical road segments with the same age. The column *Age* was selected for commodity, but the *Construction Date* could also had been used (*Construction Date, BPS3*).
2. Selection of the row that had the highest *Original Version Number* value for each subset of rows with the previous tuple, that was the valid segment. If there were several rows with the same value, simply the last row was selected.
3. Storage of the selected row and removal of the rest.

Once the cleaning was done, it was time to prepare the dataset for the modeling. Initially, the dataset represents a static problem: the IRI is the dependent variable, and predictions are based on features fixed within a specific time range. This means that it is not being considered how the IRI changes over time, but rather trying to understand or predict the IRI based on current conditions. However, the aim of this project is to solve a dynamic problem: how the variables change over time. If the average IRI Increase (i.e., the average change in IRI over the years for a segment) is used as the dependent variable, the focus is on the temporal evolution of the road roughness. This approach acknowledges that road conditions are not static and can deteriorate or improve over time due to various factors.

This implementation significantly simplifies the process for users to predict the IRI for a specific road segment. To use the model, users first need to gather data from within a year of the measurement date, which will serve as input. The model will then provide the average IRI increase for that segment. To estimate the IRI over a desired number of years, the user simply multiplies the predicted IRI increase by the number of years and adds this to the initial measurement value.

To reconvert the dataset, another loop was created, but this time its purpose was to create the two new columns needed to satisfy the dynamic problem structure. The new columns are:

1. **Years Until Next Measurement**: The information about the segments in the dataset is not complete over all the years. Therefore, is not possible to simply subtract the IRI of a year from its previous year. Calculating the years that pass from a measurement until de next measurement, from a given segment, is a solution.
2. **IRI Increase:** After subtracting the IRI of a measurement from its previous one, its needed to divide it by the obtained number of years that passed, thus returning an average per year.

To obtain these new columns, an iteration over all the unique *BPS3* keys of the dataset is done while computing the following operations:

1. Ordering of the measurements by age (please note that a given key usually has several measurements).
2. The years that have passed from one measurement to another are calculated. For greater accuracy, the month is used, returning a decimal value.
3. The IRI value of the consecutive measurements is subtracted and divided by the elapsed years.
4. The last measurement of each segment is eliminated.

It is worth noting that some of the road segments end with a negative IRI Increase, what means that the latest IRI measurement is lower than its previous one. This situation is considered as noise, as the IRI of a road segment could be the same over the years but it could not get smaller. Therefore, this type of road segments with negative IRI Increase are removed.

After all the processing, the dataset is finally ready to work with. Next, there are shown a sample of the features for a random road segment:

BPS3	ID	Date measurement	Years Until Next Measurement	IRI Increase	IRI
MN_12_1HRL_2R-L_57.3_57.4_2018	12	2019-04-19 00:00:00	0.96	0.208333333	0.9

MN_12_1HRL_2R-L_57.3_57.4_2018	12	2020-04-02 00:00:00	1.06	0.188679245	1.1

4.2. INWEVA Dataset

This dataset contains the traffic density per road segment over the Netherlands network. When the data is obtained from the Rijkswaterstaat Data Registry web page, it comes segregated by year. Therefore, a union process of the several datasets was done to make it possible to join it with the IRI dataset. Finally, a few more adjustments were needed:

1. The name changing of some columns that came in Dutch.
2. The format changing of wrongly formatted data types.
3. The removal of columns not needed to the model.
4. The creation of helpful ones like the *Year*, obtained from the measurement date.

Overall, the data wrangling applied to this dataset has not been demanding. Once completed, the columns that remain are:

- **ID**: the ID of the road set by *Rijkswaterstaat.*
- **Year**: the year when the measurement was taken.
- **Side**: the side of the road.
- **From**: the starting km of the road segment.
- **To**: the ending km of the road segment.
- **etmaal_AL**: Daily INWEVA intensity of all vehicles on a working day (in motions).
- **etmaal_L1**: Daily INWEVA intensity of passenger cars on a working day (in motions).
- **etmaal_L2**: Daily INWEVA intensity of medium truck traffic on a working day (in movements).
- **etmaal_L3**: Daily INWEVA heavy truck traffic intensity on a working day (in motions).

These remaining columns provide a comprehensive view of traffic density, segmented by vehicle type and road attributes. By integrating this traffic data with the IRI dataset, we can gain valuable insights into the relationship between traffic patterns and pavement deterioration.

4.3. KNMI Dataset

This dataset includes various climatological factors that can impact road conditions over time. The KNMI collects the climatological data using weather stations that are distributed across the Netherlands. Therefore, the information comes segmented into different datasets, each one referred to a weather station. As it has been explained before, in the IRI dataset only the region is stored, which is an aggrupation of different provinces.

The differences in data segregation required extensive data processing to enable the merging of the datasets. To achieve this, folders were created corresponding to each region in the IRI dataset. Within each folder, the datasets from weather stations located in those regions were stored. After that, a Python script was used to combine all the datasets, creating a new column named *Region* based on the folder names. Moreover, a column named *Year* was subsequently created from the date.

Initially there were many features present in the dataset, so the most meaningful ones were selected:

- **FG**: Wind speed.
- **TG**: Temperature.
- **TN**: Min temperature.
- **TX**: Max temperature.
- **RH**: Precipitation Amount.
- **SQ**: Sunshine duration.
- **SP**: Sunshine percentage.
- **UG**: Humidity.
- **Year**: Year of the measurement.
- **Region**: Region where weather station is located.

Although the data comes from its source formatted in daily measurements, it was decided to use yearly averages. This approach aligns with the use case of the problem and ensures consistency with the other datasets. All the daily measurements were grouped by *Year* and *Region*, -the two newly created features- to ensure the data matched the required format.

It was considered that more meaningful features could be obtained from these ones, so an extra step was taken to create the next variables:

- **FRZ** (Freeze Index): Summation of difference between 0°C and daily average air temperature, when daily average air temperature is less than 0°C.
- **FRZT** (Freeze–thaw): Number of days in the year when the maximum air temperature is greater than 0°C (32°F) and minimum air temperature is less than 0°C (32°F) on the same day.
- **HOT** (Days above 32°C): Number of days in the year when the maximum air temperature is greater than 32.2°C.
- **COLD** (Day below 0ºC) = Number of days in the year when the minimum air temperature is smaller than 0°C.
- **RHA**: Annual accumulated precipitation.

These derived features, such as *FRZ* and *FRZT*, are crucial for understanding the impact of extreme weather conditions on pavement performance. By integrating these climatological variables with the IRI dataset, we can develop more robust predictive models that account for environmental factors.

4.4. Complete Dataset

The full dataset is formed by the combination of the three previous datasets. First, the IRI dataset was joined with the INWEVA dataset and then, the intermediate combination with the KNMI dataset.

However, the joining process could not be performed with simple operations, such as a join by the *Year* column. As explained, the purpose of each row of the dataset is to store the information that explains the increase in IRI which the road segment would experience within one year. Therefore, it was necessary to join each row of the dataset with the weather and traffic density information found in the one-year period from the date of the measurement day. To achieve this purpose, the joining was divided into several steps.

The first was to join the IRI dataset with the INWEVA dataset without any column that indicated the date, using just the road identifiers: the *ID*, the *Side* and the *From* and *To* columns. In this way, each road segment was matched with every INWEVA record of itself, without taking the measurement year into account. This merged dataset acted as an auxiliary for later steps and looks like this:

BPS3	ID	IRI Date measurement	INWEVA Year	Etmaal_AL
MN_12_1HRL_2R-L_57.3_57.4_2018	12	2019-04-19 00:00:00	2015	19456
MN_12_1HRL_2R-L_57.3_57.4_2018	12	2019-04-19 00:00:00	2019	21233
MN_12_1HRL_2R-L_57.3_57.4_2018	12	2019-04-19 00:00:00	2020	22456

The purpose of this intermediate dataset was to apply a formula that served to compute the weighted average of the INWEVA yearly averages to each of the IRI road segments, thus solving the problem that was being faced. The formula is:

$$(1 - a) * row(N) + a * row(N + 1), a = dayoftheyear/365, N = measurementyear, row = IN$$

.

To achieve this, a Python script traversed all the unique *BPS3* keys, selecting a subset of records for each key in each iteration. The correct records were identified using the date columns. For each subset, another loop applied the specified formula, and the calculated values were stored in a new dataset.

To exemplify the formula: imagine the aimed row to join was measured on July 1, 2020, which is approximately halfway through the year, $a \approx 0.5$, (for the real task, using a developed Python program the exact day is obtained and divided by 365). Then the interpolated value would be calculated: $(1 - 0.5) \cdot row(2020) + 0.5 \cdot row(2021)$. The result is the average of the traffic densities of 2020 and 2021, weighted equally.

Once the IRI and INWEVA datasets were prepared, the same process was applied to integrate the KNMI dataset, but without any initial joins. Given the simplicity of the KNMI dataset, the program could accurately match the road segments using the *Year* and *Region* columns. With this final addition, the dataset was complete, allowing the model training and evaluation to start. The appearance of the complete dataset would be something similar to this table:

BPS3	ID	Date measurement	IRI Increase	INWEVA Features	KNMI Features
MN_12_1HRL_2R-L_57.3_57.4_2018	12	2019-04-19 00:00:00	0.208333333	Weighted average values	Weighted average values
MN_12_1HRL_2R-L_57.3_57.4_2018	12	2020-04-02 00:00:00	0.188679245	Weighted average values	Weighted average values

For simplicity and space reason several columns of the initial IRI dataset have been omitted. Also, the columns of INWEVA and KNMI datasets have been grouped in representative columns.

5. Gained Knowledge and Model Evaluation

This section presents the outcomes of the Bayesian Neural Network (BNN) applied to the problem at hand, including a detailed analysis of its performance. Through the application of the BNN, the aim is not only to make accurate predictions but also to understand the underlying uncertainties and patterns within the data. The evaluation process includes visual representations of the results, such as graphs and charts, which help to illustrate the model's predictions, uncertainty quantifications, and overall effectiveness.

Before presenting the results, the training process will be explained. An intensive grid search was performed during the training phase. It was mandatory due to the complexity of the data preprocessing choices and the broad range of hyperparameters. To put in context, these variable components were:

- The IRI Increase threshold, ranging from 0 to 0.8 in increments of 0.2. The data above upper thresholds are scarce.
- The number of PCA components, ranging from 3-13. The numbers are based on the scree plot of the PCA.
- The number of layers in the network, ranging from 2 to 5.
- The number of neurons per layer, ranging from 5 to 25 in increments of 5.

One of the parameters is noticeable, the IRI threshold. It was introduced due to the distribution of the *IRI Increase* variable, which contains mainly values closer to the increase of 0. Its purpose is to complete the study and determine whether the model would work better for higher values of *IRI Increase*. This plot shows the distribution of it:

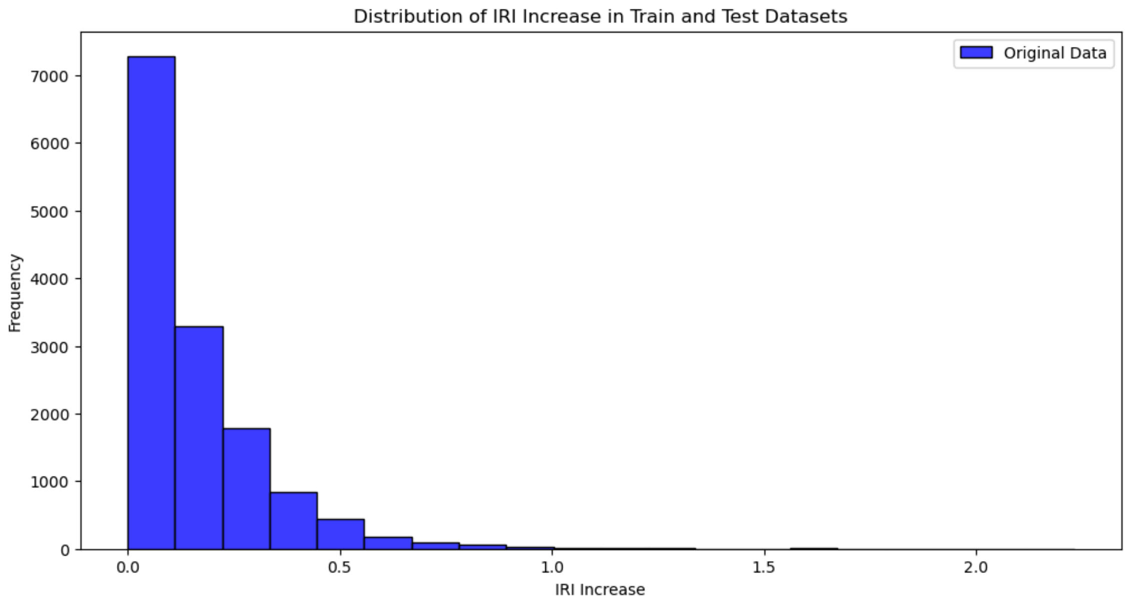

Figure 2: IRI Increase distribution

The objective of the training was to find a combination of parameters that retrieved a model which did not overfit the data. Therefore, the selected combination consisted of the parameters that originated a similar Pearson Correlation between the mean of the posterior sample $\hat{y}$ and y. Furthermore, it was crucial to ensure that the model's complexity—defined by the total number of parameters—was appropriately scaled to the size of the dataset. To this end, the selected combination of parameters adhered to the widely accepted guideline of maintaining a ratio of at least 10 samples per model parameter. This ratio serves as a safeguard, reducing the risk of overfitting by ensuring that the model has sufficient data to learn meaningful patterns, rather than simply memorizing the training data.

5.1. Results

The following graphs provide a detailed overview of the model's performance, as they show the scatter plots of the best configurations obtained by the grid search. The posterior distributions of the BNN weights and biases will not be included so as to not overload the section. Predictions for the train and test set are displayed:

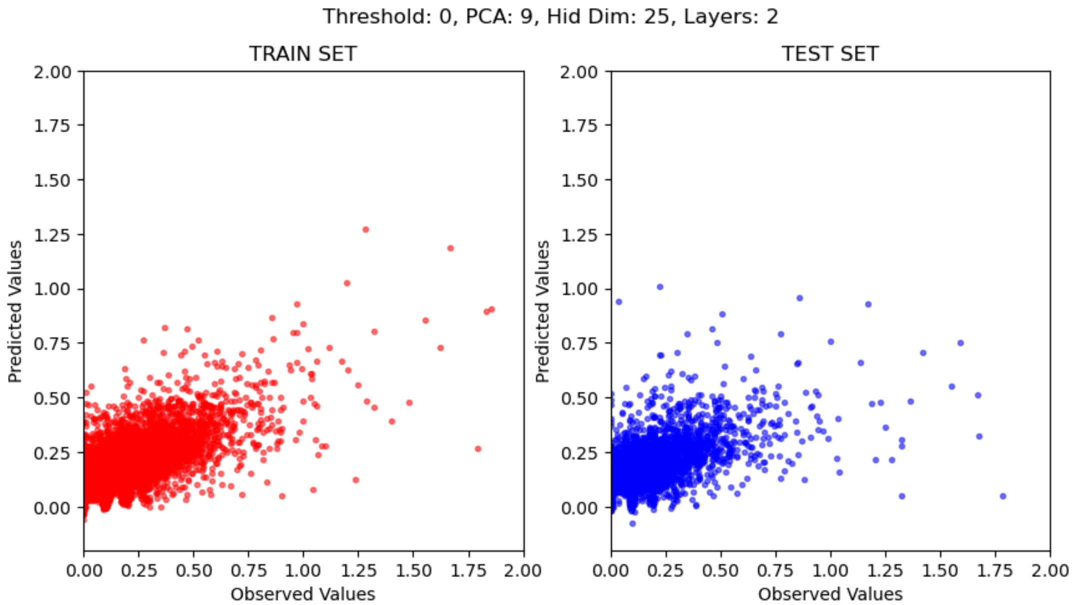

Figure 3: BNN threshold 0

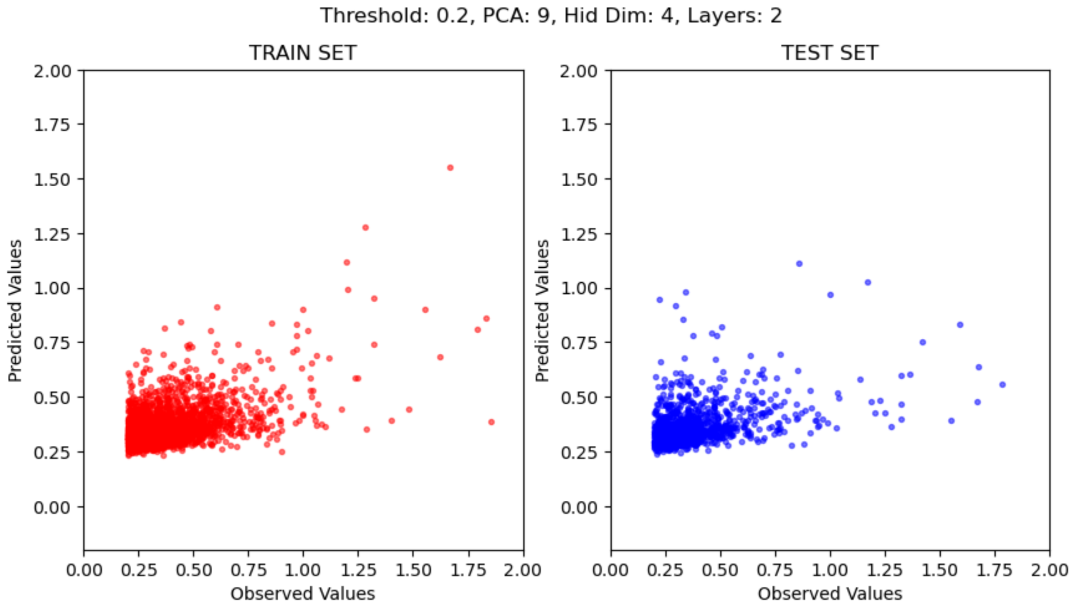

Figure 4: BNN threshold 0.2

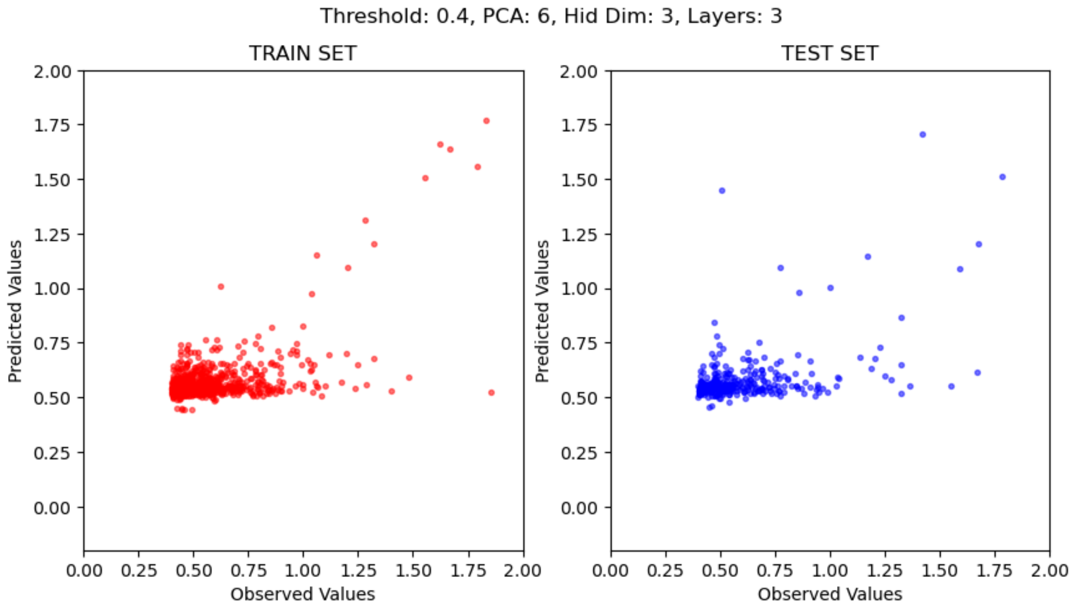

Figure 5: BBN threshold 0.4

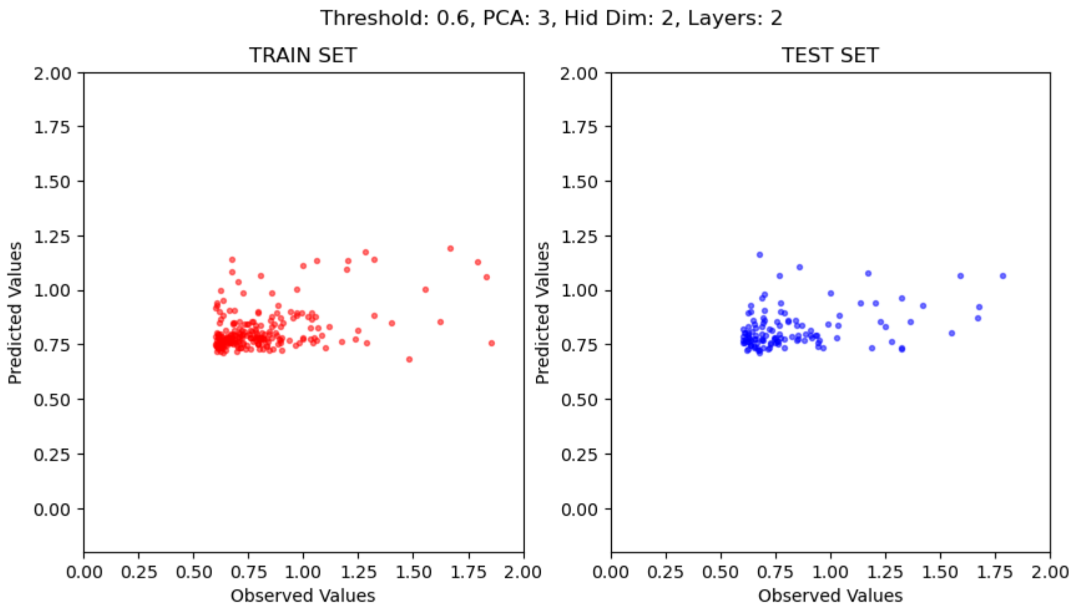

Figure 6: BNN threshold 0.6

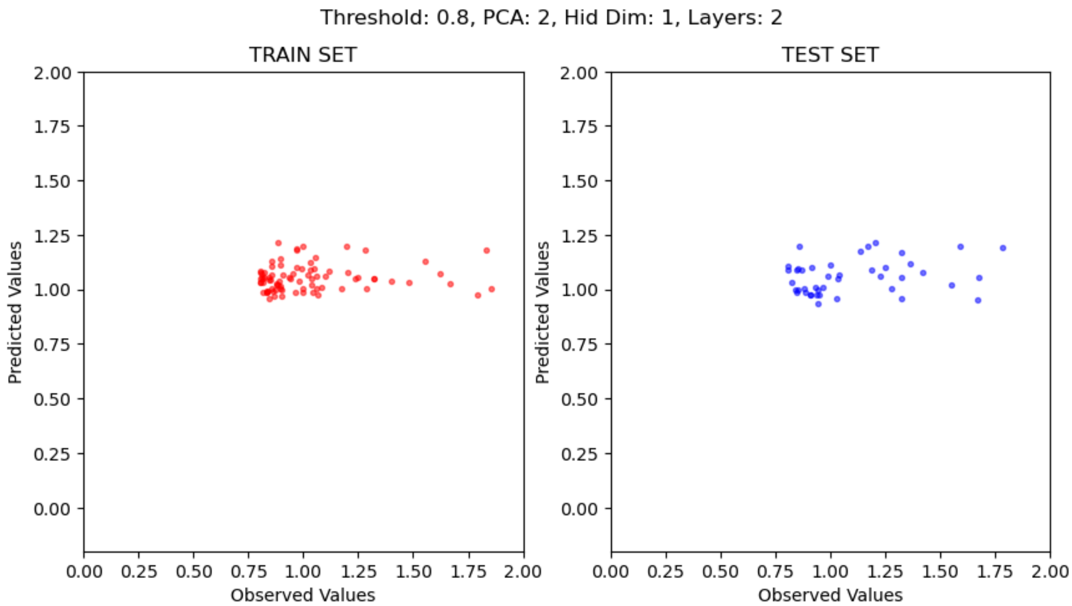

Figure 7: BNN threshold 0.8

Summary of metrics:

	Train Correlation	Test Correlation
Threshold 0	0.737155542763503	0.6489600820399735
Threshold 0.2	0.5236611622736915	0.41833971246750734
Threshold 0.4	0.5905217477712591	0.45671483291587894
Threshold 0.6	0.4394893281911075	0.3310962074469568
Threshold 0.8	0.22606797130957104	0.421327996595717

It is noticeable how the performance of the BNN decreases as the threshold increases, effect which could be caused by the inferior number of samples present in datasets with greater thresholds (as shown in the distribution). Nevertheless, thanks to the constraint of parameters commented earlier, it is ensured that the ratio of samples per parameter is respected. Therefore, the higher results with lower thresholds could be occasioned by the more complex BNNs that can be trained thanks to the wider range of data available. Ultimately, the best model is obtained without threshold restrictions in the data, with a correlation of 0.74 in the train dataset and a correlation of 0.65 for the test dataset. This metric represents just represents how accurate the model has been in predicting each point, however, the usage of the model would be enhanced by its advantages, like the uncertainty distribution that is being generated for each prediction.

This best model arises from a network created with 2 layers of 25 neurons each. With two hidden layers, the network can model non-linear relationships between input features and the target variable, capturing more complex patterns than a simple linear model. What is more, the number of parameters (weights and biases) in this network is substantial but manageable. This network has enough parameters to capture complex relationships but is not overly large, which helps maintain generalization and prevent overfitting.

On the other hand, there is the number of PCA components involved. To understand why nine, the following graph will be used:

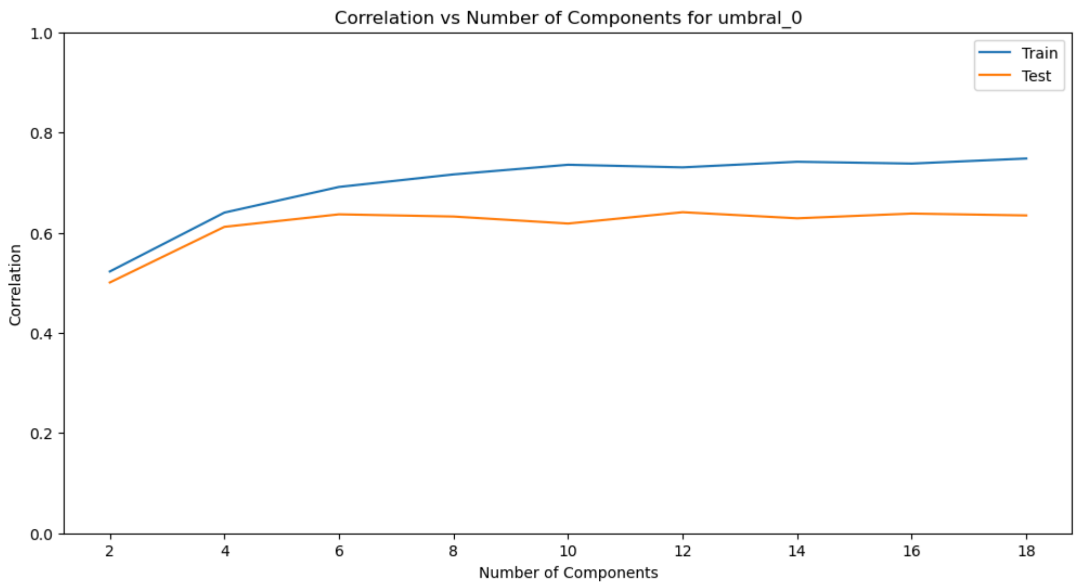

Figure 8: Correlation per number of principal components

To generate this graph, the number of layers and neurons per layer were set according to the optimal values determined earlier. As shown in the graph, there is a point around ten principal components, beyond which the correlation in the training dataset no longer improves, meaning this could be considered the point with the best balance between complexity and performance. This indicates that, increasing the number of principal components beyond this point does not yield additional predictive power and might lead to overfitting (especially evident from the gap between train and test correlations).

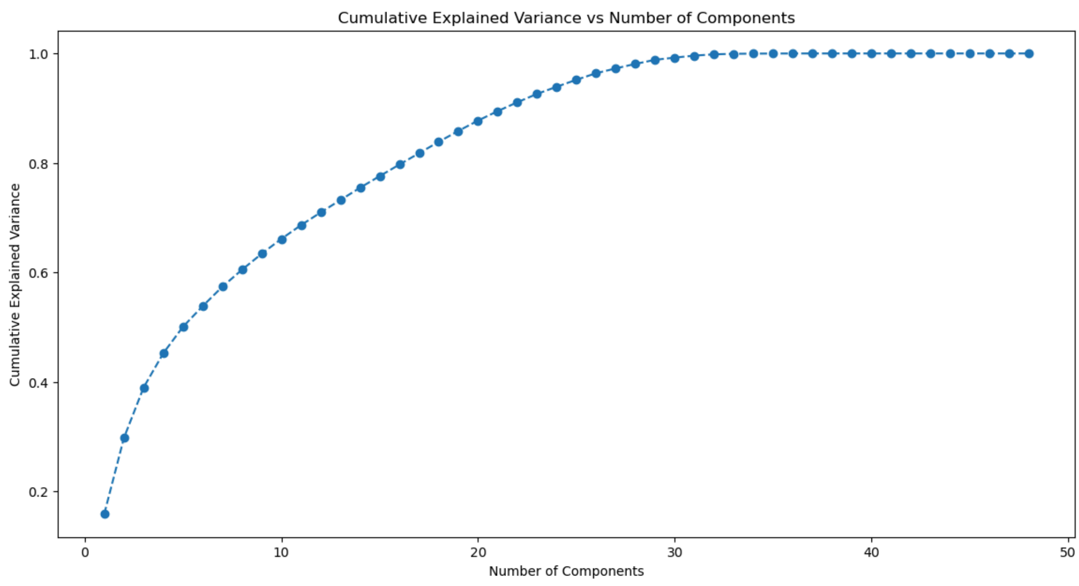

Figure 9: Scree plot

As it is seen in the scree plot, ten principal components only explain about sixty percent of the explained variance of the data; around this point the curve starts to flatten a little bit, which aligns with the point where the BNN's correlation stops improving significantly. This suggests that the first ten PCs capture the most critical information in the dataset, which is sufficient for the BNN's performance.

To continue, the graphs obtained with the LR models will be displayed. As the LR is a basic model, these results will serve as a reference for the BNN performance.

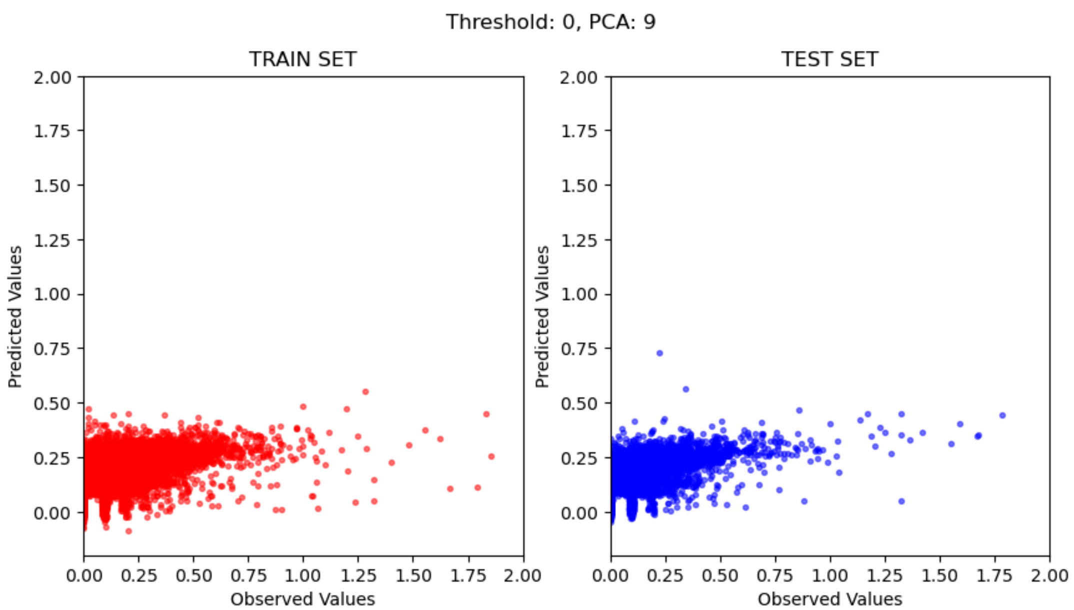

Figure 10: LR threshold 0

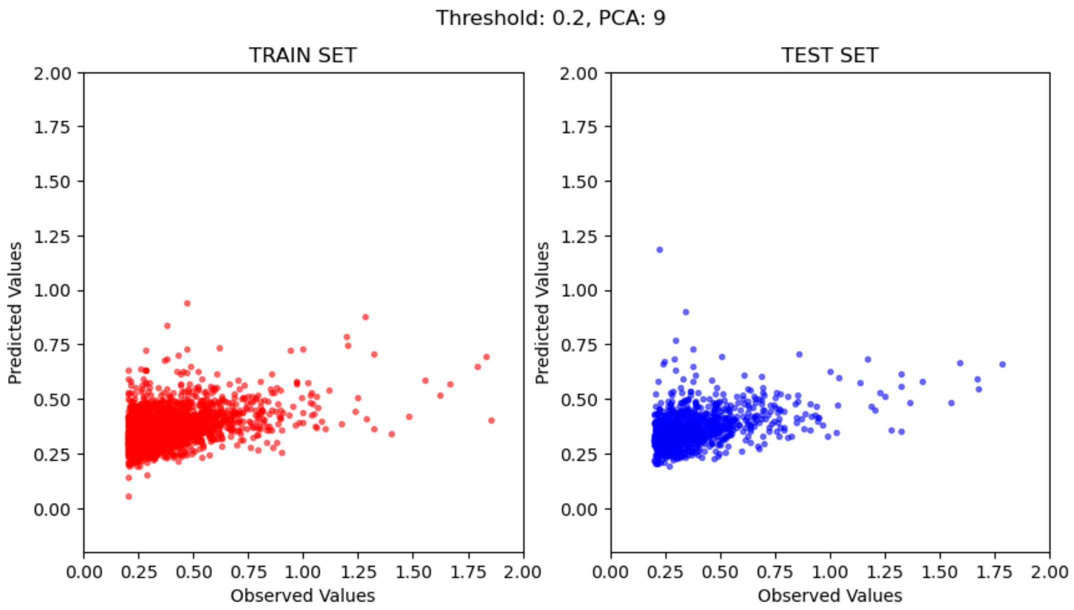

Figure 11: LR threshold 0.2

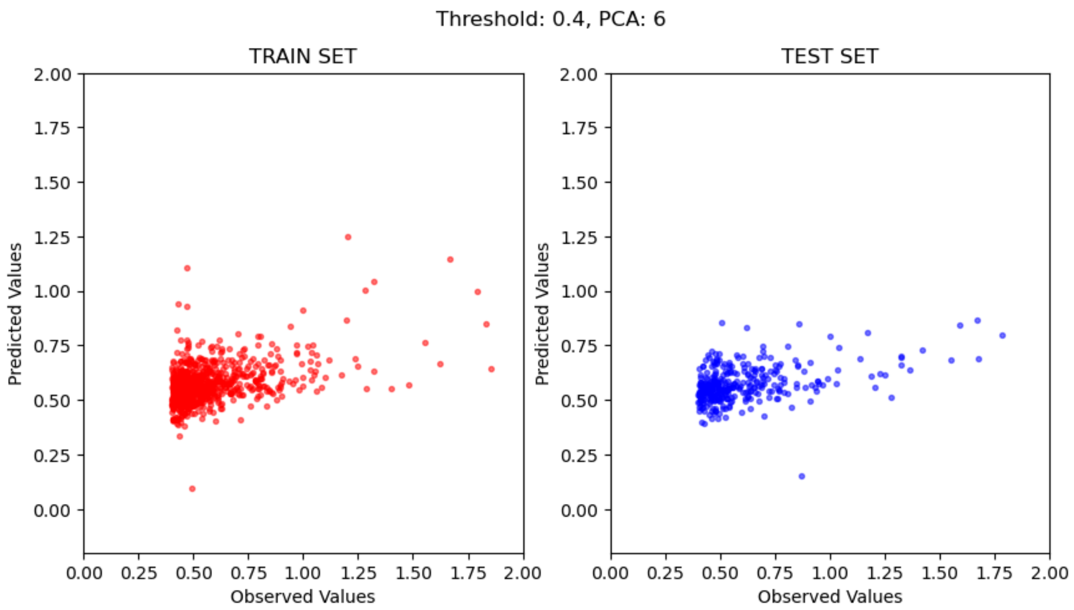

Figure 12: LR threshold 0.4

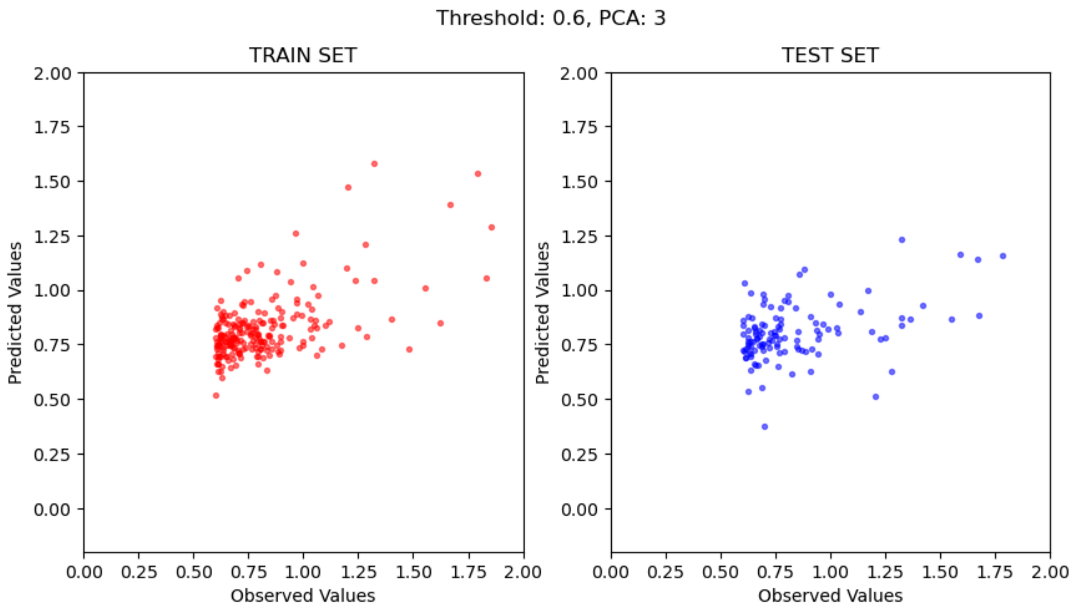

Figure 13: LR threshold 0.6

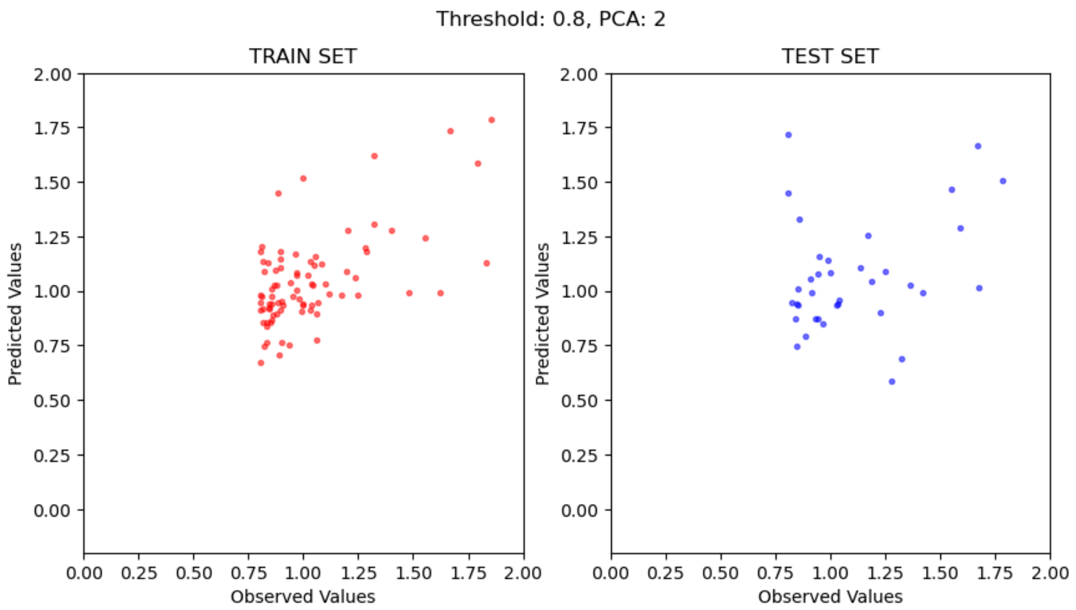

Figure 14: LR threshold 0.8

Summary of metrics:

	Train Correlation	Test Correlation
Threshold 0	0.5660836847952768	0.5898636292204765
Threshold 0.2	0.42011607041774374	0.4240273168589845
Threshold 0.4	0.44141540367673765	0.4215875901476355
Threshold 0.6	0.5568015523383234	-0.023788971762854726
Threshold 0.8	0.6316588830854764	-0.03212347942443506

Although the overall correlation scores are lower than before, the difference between the test correlation scores of the Linear Regression (LR) and the Bayesian Neural Network (BNN) is not significant. The first three models, particularly those with 0.2 and 0.4 thresholds, achieve test correlations comparable to the BNN. However, the last two models perform poorly, resulting in a test correlation of zero, which indicates severe overfitting during training. The best performance is again observed in the model without threshold restrictions, achieving a test correlation of 0.59—remarkably close to the 0.65 achieved by the BNN.

5.2. Supporting Explanation

The understanding of the decision process in a complex model such as Bayesian Neural Networks can be challenging as it is essentially a NN, which also is a black box model. However, the combination with complementary analytical tools, such as PCA loadings and LR coefficients will bring important insights. A PCA loading plot shows the amount to which each original variable contributes to each principal component., which could indicate the most influential characteristics in the data. This could help to understand which are the key drivers of variance in the data that might be leveraged by the BNN. On the other hand, the coefficients plot from the LR model that shows us a direct view of the linear relationship between features and the target. The coefficients that are shown are the ones which remained after applying ANOVA. By comparing these two plots, we can gain a better understanding of how the BNN might be making decisions, identifying which features it prioritizes and how they influence its predictions.

First, the loading plots per threshold of the two main principal components will be displayed:

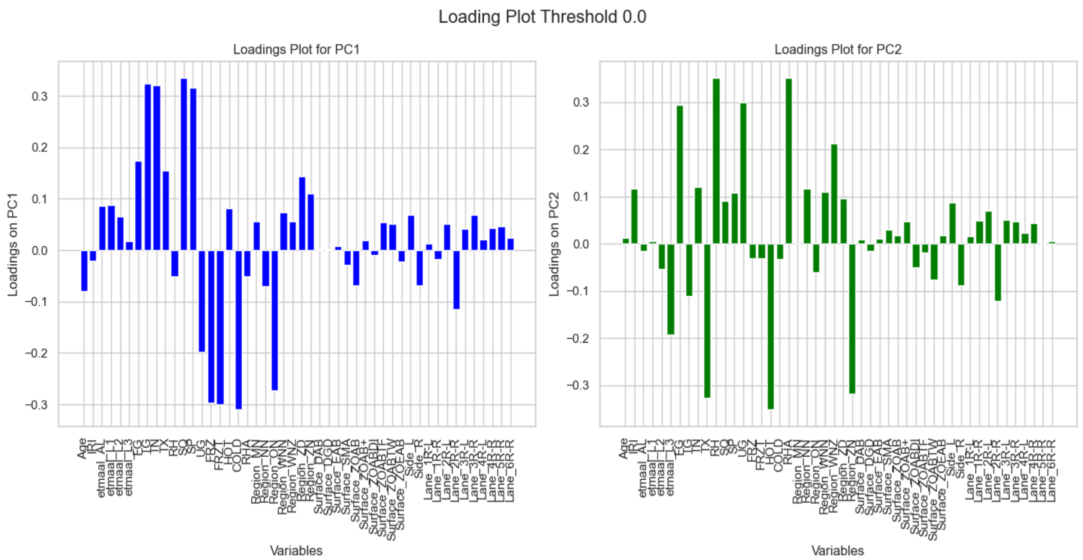

Figure 15: Loading Plot threshold 0

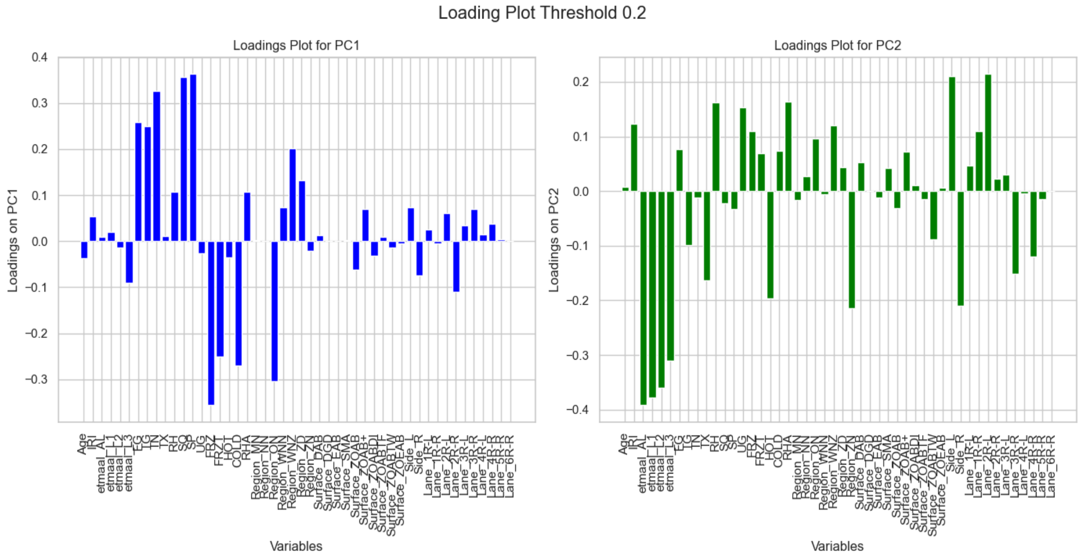

Figure 16: Loading Plot threshold 0.2

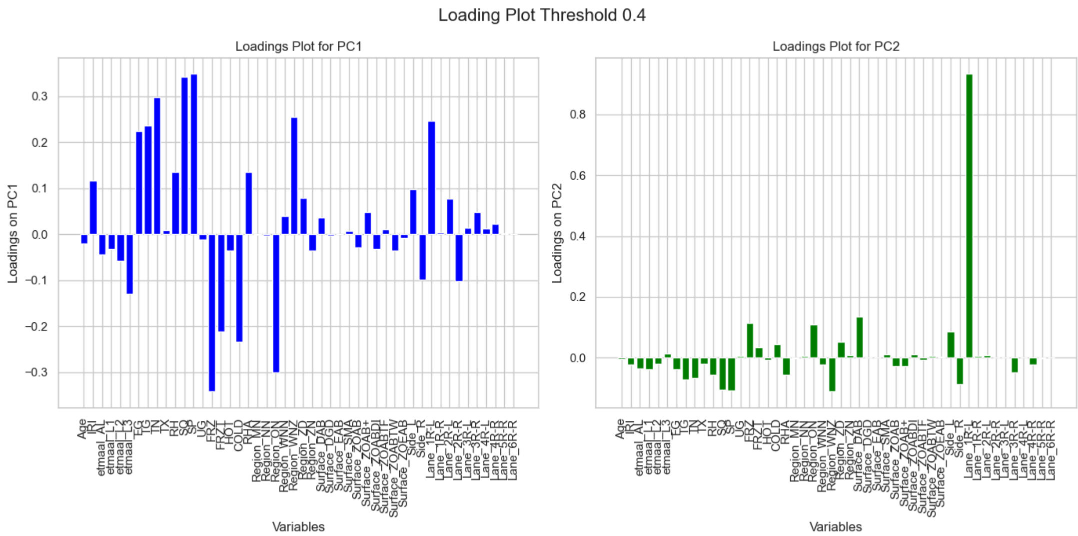

Figure 17: Loading Plot threshold 0.4

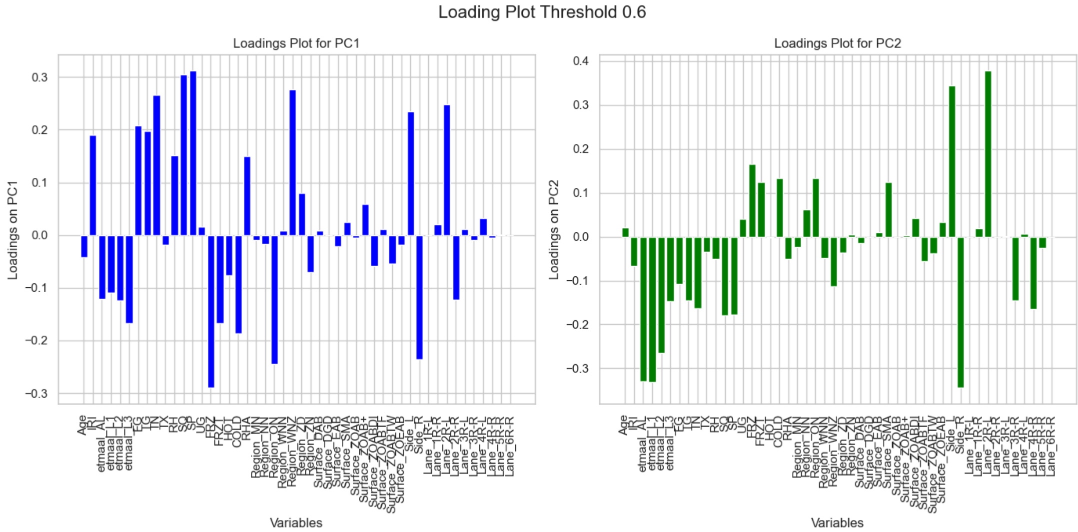

Figure 18: Loading Plot threshold 0.6

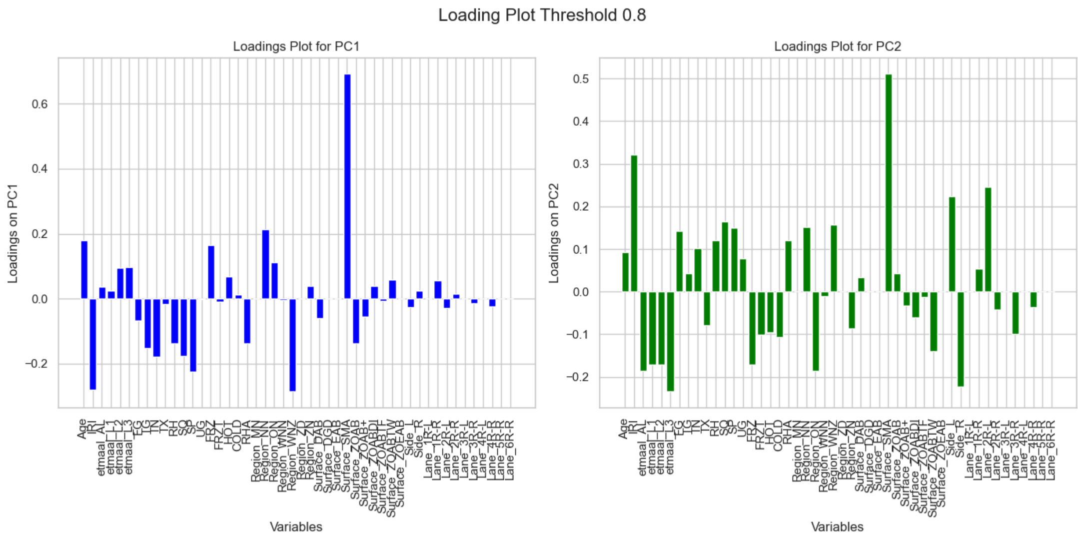

Figure 19: Loading Plot threshold 0.8

Despite the reduction in data per threshold increasing, the loading plots reveal a consistent pattern for the PC1 (unless for the 0.8 threshold): mainly the KNMI features maintain high loadings (in absolute terms), while other features become relevant depending on the threshold. This behavior suggests that the KNMI features are the most influential in explaining the variance within the dataset, even as data becomes scarcer. These features likely represent key factors that drive the underlying patterns in the data, regardless of the threshold applied. Conversely, the remaining features do not show significant contributions nor the PC2 does show a significant pattern.

What the graph suggests that the features *FG, TG, SQ and SP* with the highest loading values, have a meaningful positive relationship with the *IRI Increase*, i.e. as these increase, the component's score augments. Alternatively, the features *UG, FRZ, FRZT and COLD* with low negative loadings can be interpreted as having an inverse relationship with the IRI Increase. This means that as these features increase, they tend to contribute to a decrease in the component's score. The variations occasioned in the component's score by these features would influence indirectly the *IRI Increase* prediction of the BNN, as these changes are reflected in the input to the BNN, thereby affecting the model's interpretation of the underlying patterns and, consequently, its prediction of *IRI Increase*.

The other features with minimal loadings likely have little to no impact on the principal component's ability to capture the variance in the data, which means they are less relevant for understanding the primary trends in the dataset as the threshold increases.

The next analysis is based on the coefficients obtained from the LR model after applying ANOVA:

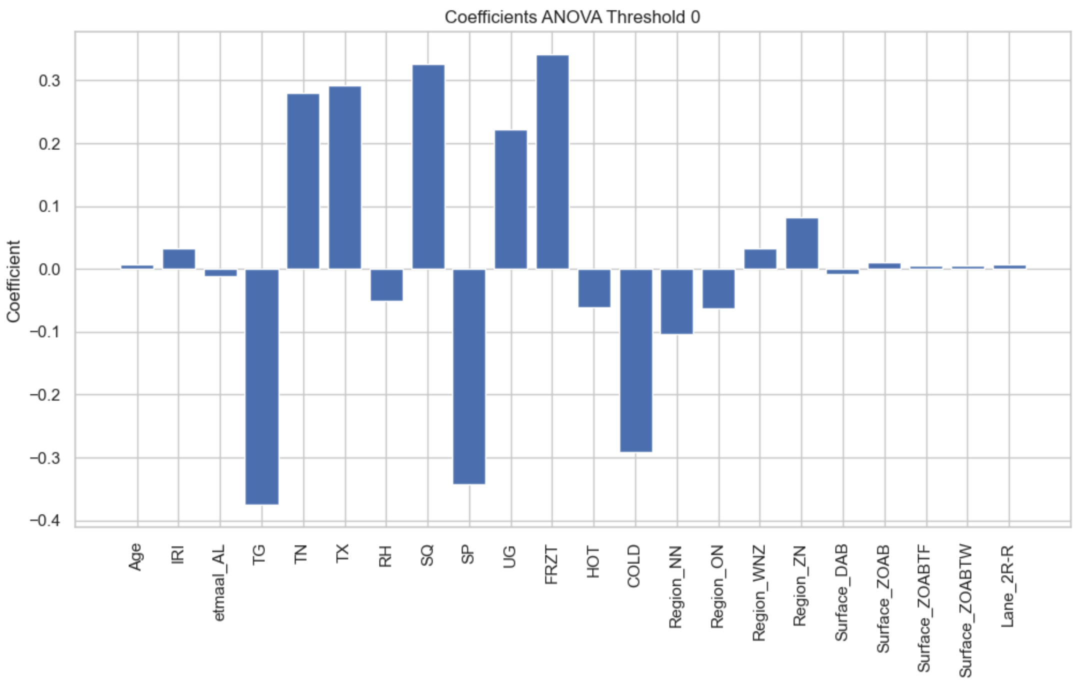

Figure 20: ANOVA threshold 0

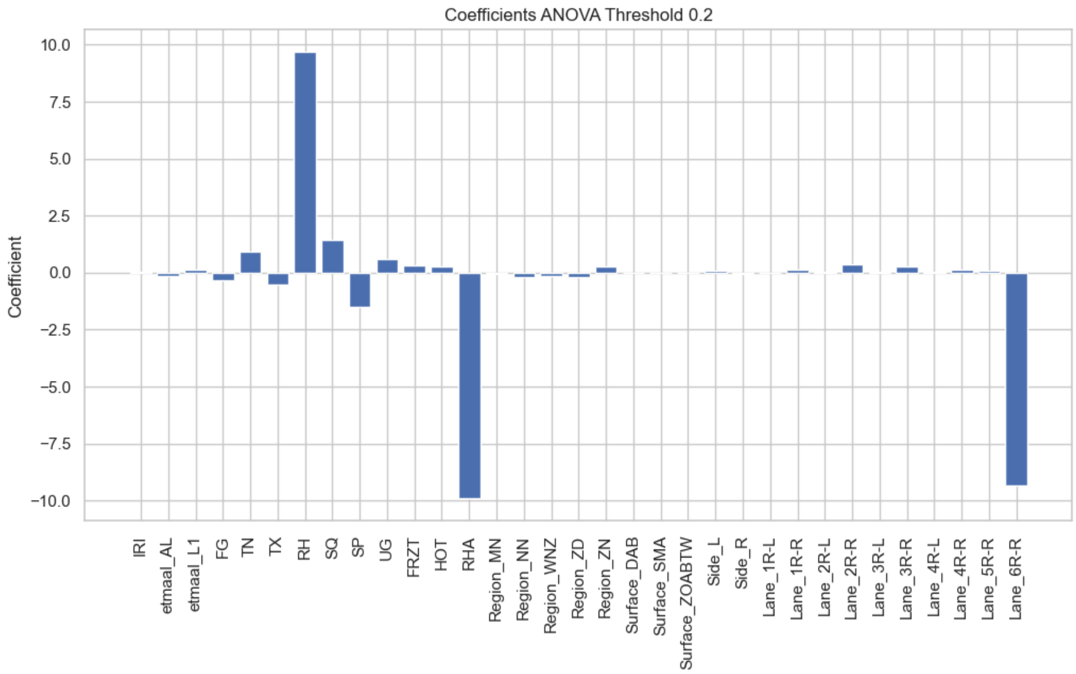

Figure 21: ANOVA threshold 0.2

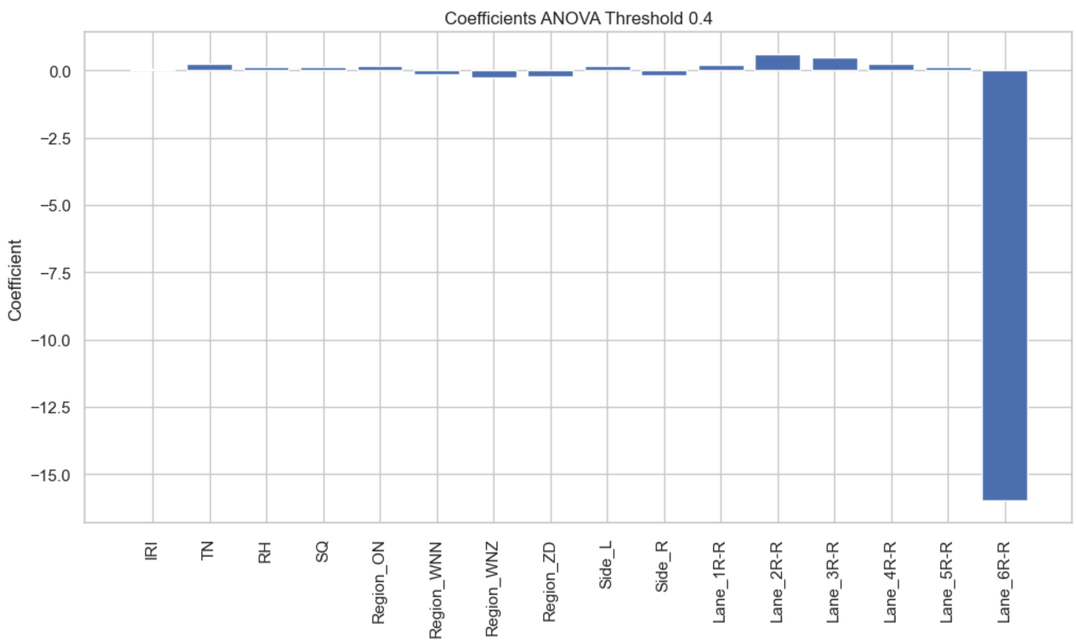

Figure 22: ANOVA threshold 0.4

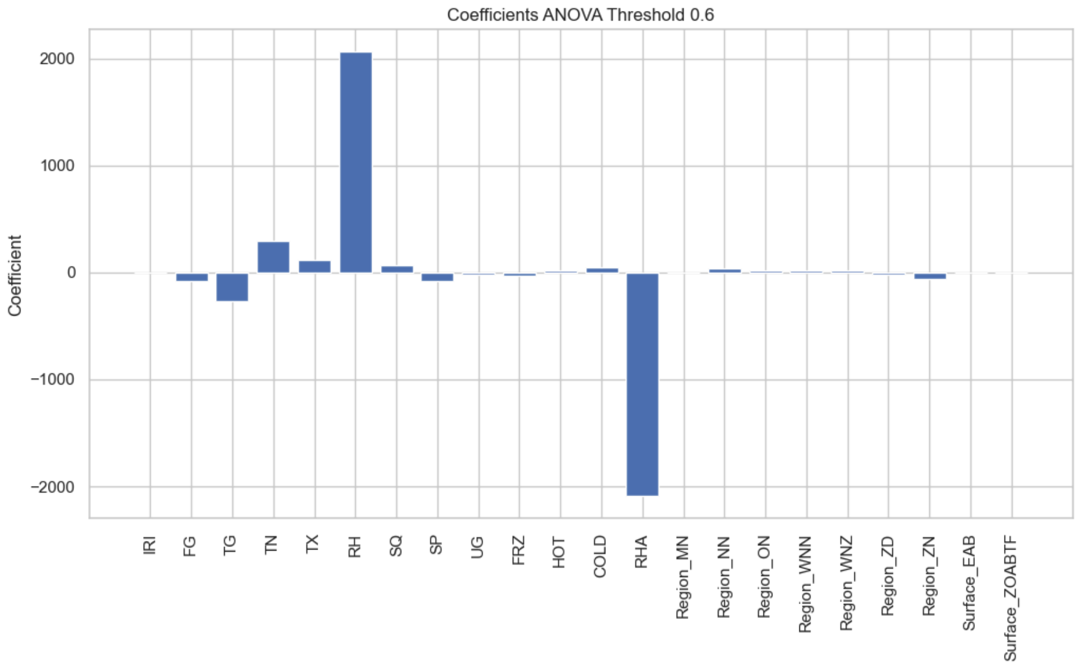

Figure 23: ANOVA threshold 0.6

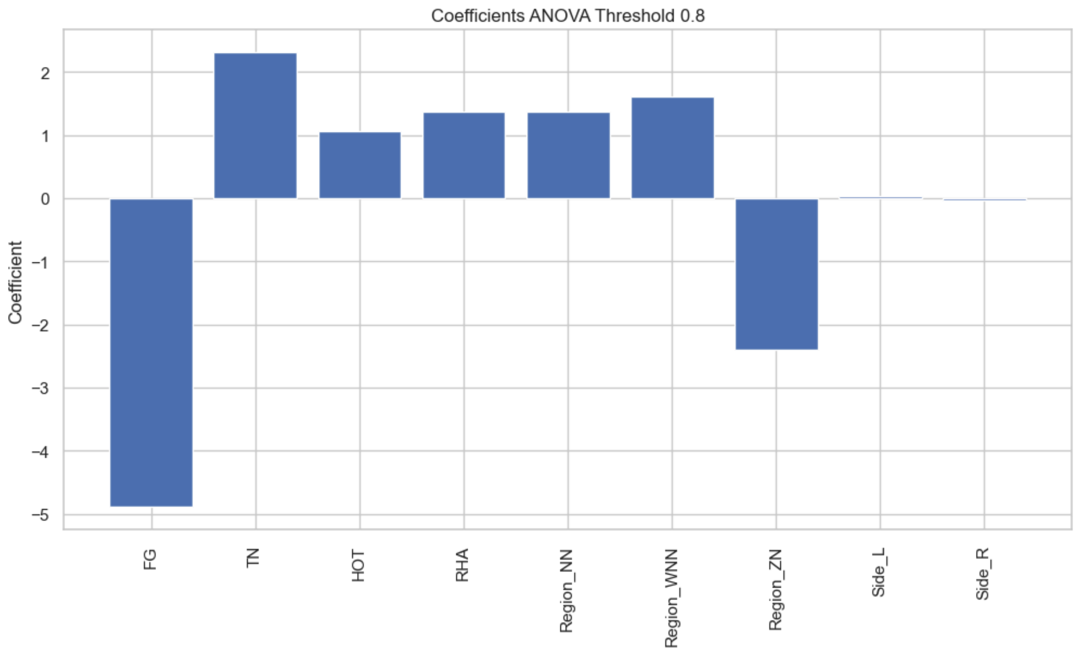

Figure 24: ANOVA threshold 0.8

The LR model indicate the direct relationship between each feature and the *IRI Increase*. The lack of a consistent pattern in the LR coefficients suggests that the influence of features on the target variable may not be stable or consistent across different thresholds, probably due to complex interactions or non-linear relationships that the LR model, which is linear, cannot capture. Moreover, as the thresholds increase and the data becomes sparser, the LR model may struggle to identify stable relationships, leading to fluctuating coefficients.

This contrasts with PCA, where the method focuses on capturing the maximum variance in the data, which might remain relatively stable despite the reduced data; even the loading plots change its pattern when this reduction in data exacerbates, like it has been shown in the 0.8 threshold loading plot.

When looking at the coefficients from the ANOVA obtained from the LR with threshold 0 (corresponding to the best BNN model and the dataset with the most available data), the KNMI are again the most meaningful features. Based on the graph, higher values of *TG, SP* and *COLD* will decrease the *IRI Increase* prediction, while higher values of *TN, TX, SQ, UG* and *FRZT* will increase its value. In the rest of the graphs, some of the KNMI features also gain importance.

6. Conclusion

This project set out to achieve three main objectives: the creation of a comprehensive dataset by integrating our data with public datasets, the development of a probabilistic machine learning model, specifically a Bayesian Neural Network (BNN), to generate and evaluate predictions from this dataset, and the implementation of a linear regression model to complement the BNN and provide a deeper understanding of the data.

The objectives were successfully achieved, though the process highlighted several key challenges. By integrating the IRI, KNMI, and INWEVA datasets using Python, a comprehensive dataset was created, serving as the foundation for training the Bayesian Neural Network. The probabilistic model developed—the BNN—was able to generate predictions, confirming that the IRI Increase is influenced by the dataset features. However, it became evident that accurately predicting the IRI increment with the available dataset is challenging, whether using linear models or Bayesian networks. This suggests that the current dataset may not sufficiently capture the complexities required for precise predictions. The linear regression model, although simpler, returned similar outcomes in some thresholds and effectively complemented the BNN by elucidating the linear relationships within the data, providing further insights into the model's behavior. This suggests that the linear relationships present in the data are strong enough to be effectively captured by a basic linear model, indicating that the complexity of the BNN may not be fully utilized given the current dataset.

Additionally, the project revealed significant issues with data overfitting, particularly in cases where the models performed well on the training data but failed to generalize to unseen data (as seen with the LR). The most important takeaway from this work is the recognition of the dataset's limitations and the need for improvement. To improve IRI increment predictions, the dataset could be enhanced by incorporating more diverse features, increasing sample size, and reducing data noise. These improvements could help create a more robust and predictive model.

Throughout the project, several challenges were encountered, particularly in managing and integrating large datasets and in fine-tuning the BNN to balance accuracy with interpretability. These challenges were addressed through a combination of data preprocessing techniques, careful model selection, and iterative tuning of hyperparameters. The use of state-of-the-art technologies, such as advanced machine learning frameworks and data analysis tools, was crucial in overcoming these obstacles.

This project has also served as an opportunity to deepen my understanding of Bayesian methods and their application in neural networks, as well as to learn about the integration of linear regression techniques with machine learning models; all while using Python. The experience of tackling these challenges and learning new concepts has been invaluable, particularly in understanding the importance of combining different technologies and methods to solve complex problems.

6.1. Legacy

To follow the work done in Python enter the following link: https://github.com/Sviallo/TFG_GCD_UPV

7. Future Works

Future efforts could focus on refining the BNN model to improve its performance on unseen data, exploring alternative methods of model interpretation, or expanding the dataset to include additional features that could enhance prediction accuracy. Additionally, other models could be investigated, like Recurrent Neural Networks, which are specifically designed to handle temporal dependencies. They can capture the evolution of features over time, which could improve the predictive accuracy for tasks like forecasting *IRI Increase* based on past conditions.

8. References

[1] A. Hu, Q. Bai, L. Chen, S. Meng, Q. Li, y Z. Xu, «A review on empirical methods of pavement performance modeling», *Constr. Build. Mater.*, vol. 342, p. 127968, ago. 2022, doi: 10.1016/j.conbuildmat.2022.127968.

[2] A. Gupta, P. Kumar, y R. Rastogi, «Critical review of flexible pavement performance models», *KSCE J. Civ. Eng.*, vol. 18, n.º 1, pp. 142-148, ene. 2014, doi: 10.1007/s12205-014-0255-2.

[3] S. Cano-Ortiz, P. Pascual-Muñoz, y D. Castro-Fresno, «Machine learning algorithms for monitoring pavement performance», *Autom. Constr.*, vol. 139, p. 104309, jul. 2022, doi: 10.1016/j.autcon.2022.104309.

[4] «Explained: Neural networks», MIT News | Massachusetts Institute of Technology. Accedido: 12 de enero de 2024. [En línea]. Disponible en: https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

[5] C. GOYAL, «Complete Guide to Gradient-Based Optimizers in Deep Learning», Analytics Vidhya. Accedido: 12 de enero de 2024. [En línea]. Disponible en: https://www.analyticsvidhya.com/blog/2021/06/complete-guide-to-gradient-based-optimizers/

[6] G. I. Webb, «Overfitting», en *Encyclopedia of Machine Learning*, C. Sammut y G. I. Webb, Eds., Boston, MA: Springer US, 2010, pp. 744-744. doi: 10.1007/978-0-387-30164-8_623.

[7] J. Johnson, «What's a Deep Neural Network? Deep Nets Explained», BMC Blogs. Accedido: 14 de enero de 2024. [En línea]. Disponible en: https://www.bmc.com/blogs/deep-neural-network/

[8] S. Hochreiter, «The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions», *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 6, pp. 107-116, abr. 1998, doi: 10.1142/S0218488598000094.

[9] T. School, «¿Qué es y cómo funciona la backpropagation en redes neuronales? | Tokio», Tokio School. Accedido: 14 de enero de 2024. [En línea]. Disponible en: https://www.tokioschool.com/noticias/backpropagation-redes-neuronales/

[10] S. Cristina, «The Chain Rule of Calculus for Univariate and Multivariate Functions», MachineLearningMastery.com. Accedido: 14 de enero de 2024. [En línea]. Disponible en: https://machinelearningmastery.com/the-chain-rule-of-calculus-for-univariate-and-multivariate-functions/

[11] A. G. Wilson y P. Izmailov, «Bayesian Deep Learning and a Probabilistic Perspective of Generalization», en *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2020, pp. 4697-4708. Accedido: 6 de agosto de 2024. [En línea]. Disponible en: https://proceedings.neurips.cc/paper/2020/hash/322f62469c5e3c7dc3e58f5a4d1ea399-Abstract.html

[12] J. Q. Yu, E. Creager, D. Duvenaud, y J. Bettencourt, «Bayesian Neural Networks», Distill. Accedido: 6 de agosto de 2024. [En línea]. Disponible en: http://distill.pub/

[13] N. Sholevar, A. Golroo, y S. R. Esfahani, «Machine learning techniques for pavement condition evaluation», *Autom. Constr.*, vol. 136, p. 104190, abr. 2022, doi: 10.1016/j.autcon.2022.104190.

[14] «How to Perform a SWOT Analysis», Investopedia. Accedido: 25 de julio de 2024. [En línea]. Disponible en: https://www.investopedia.com/terms/s/swot.asp

ANEXO

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. **Fin de la pobreza.**				X
ODS 2. **Hambre cero.**				X
ODS 3. **Salud y bienestar.**				X
ODS 4. **Educación de calidad.**				X
ODS 5. **Igualdad de género.**				X
ODS 6. **Agua limpia y saneamiento.**				X
ODS 7. **Energía asequible y no contaminante.**				X
ODS 8. **Trabajo decente y crecimiento económico.**		X		
ODS 9. **Industria, innovación e infraestructuras.**	X			
ODS 10. **Reducción de las desigualdades.**				X
ODS 11. **Ciudades y comunidades sostenibles.**		X		
ODS 12. **Producción y consumo responsables.**				X
ODS 13. **Acción por el clima.**				X
ODS 14. **Vida submarina.**				X
ODS 15. **Vida de ecosistemas terrestres.**				X
ODS 16. **Paz, justicia e instituciones sólidas.**				X
ODS 17. **Alianzas para lograr objetivos.**				X

- **Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.**

The present thesis, "Application of probabilistic modeling to predict road pavement deterioration", has a close relationship with three key Sustainable Development Goals: SDG 8 (Decent Work and Economic Growth), SDG 9 (Industry, Innovation and Infrastructure), and SDG 11 (Sustainable Cities and Communities).

Firstly, this project contributes to SDG 8 by promoting sustained, inclusive, and sustainable economic growth. Well-maintained road infrastructures are fundamental for economic growth, as they enable efficient transportation of goods and facilitate workforce mobility. By using machine learning techniques to optimize road maintenance strategies, this thesis helps ensure a reliable and resilient transportation network, which is a pillar of economic growth. Moreover, efficient and data-driven road maintenance can generate significant savings for public administrations, freeing up resources that can be allocated to other development priorities.

Furthermore, this work strongly aligns with SDG 9, which seeks to build resilient infrastructure and promote innovation. Applying Bayesian Neural Network models to predict pavement deterioration over time is a highly innovative solution. It leverages the latest advancements in artificial intelligence and machine learning to address the challenge of road maintenance. This application of cutting-edge technologies exemplifies the kind of innovation advocated by SDG 9. Additionally, having accurate predictive models enables proactive maintenance planning, resulting in more resilient and durable road infrastructures.

Lastly, this research makes a valuable contribution to SDG 11, which focuses on making cities and human settlements inclusive, safe, resilient, and sustainable. Optimal pavement maintenance is crucial for urban sustainability. Well-maintained roads improve road safety, reduce accidents, and make cities more livable. Furthermore, well-maintained pavements reduce vehicle fuel consumption and associated emissions, contributing to improved urban air quality. By facilitating efficient road maintenance planning, this project promotes more sustainable and resilient urban infrastructures.

In conclusion, this thesis, which applies machine learning techniques to predict pavement deterioration, clearly aligns with SDGs 8, 9, and 11. It contributes to sustainable economic growth by ensuring reliable transportation infrastructure, promotes innovation by applying cutting-edge technologies to road maintenance, and fosters more sustainable cities by optimizing pavement management. This project illustrates how technological innovation can be a powerful driver for advancing the Sustainable Development Goals.