Master Computer Science

Forecasting Stone Loss Percentage on Dutch Highways with Machine Learning and Deep Learning Models

Name:	Weiji Wang
Student ID:	s3644677
Date:	19/08/2024
Specialisation:	Computer Science: Data Science
1st supervisor:	Dr. L. Cao
2nd supervisor:	Dr. N. van Stein
3rd supervisor:	Greet Leegwater
4th supervisor:	Shang Jen Wang
Daily supervisor:	Andrius Bernatavicius

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Modern roads now offer unprecedented levels of mobility, and the number of vehicles on these roads has been steadily rising since the early 20th century [1]. In the Netherlands, the majority of paved roads in highway network have a top layer of PA (Porous Asphalt Concrete). Ravelling, which refers to the loss of stones from the pavement surface, is the most common form of damage in PA top layers [2]. This can result in windshield damage, hazardous traffic conditions, and loss of performance with respect to noise reducing properties. As a result, well-planned road maintenance is essential to ensure optimal performance for users and minimize disruptions, ensuring seamless and efficient traffic. The Dutch Highway Agency annually detects the stone loss on road surface to plan maintenance. This information has potential to be used as bases for service life prediction models, however due to the time horizon and extensive road network in the Netherlands, the data size is rich and it is of high need to analyze the data to make asphalt pavement performance predictions since maintenance decision should be made. Therefore, in this project, the purpose of this research is to utilize information on actual performance in machine learning and deep learning settings to predict future stone loss.

Keywords:Asphalt Pavement, Deep Learning, Machine Learning, Neural Networks, Degradation Model

Contents

1 Introduction

Road infrastructure plays a crucial and dynamic role in the progress of cities and communities. It is regarded as one of the key elements contributing to the well-being and comfort of road users. Additionally, it is one of the sectors that influence the socio-economic development of nations. [3]. In the Netherlands, which has one of the most densely branched road networks in the world [4], the majority of the paved roads in the highway network have a top layer of PA. And one of the advantages of using PA mixes is the reduction in noise levels, the improvement of driving comfort and skid resistance in rainy conditions. Ravelling, the loss of stones from the surface is by far the decisive damage mechanism for PA service life [5]. Severe ravelling problems reduce ride quality, increase noise, and the risk of windscreen damage [4]. Therefore, evaluating and forecasting the performance of porous asphalt pavement with respect to ravelling is of great importance, after which a maintenance decision will be made based on the results. The primary challenge lies in developing an effective prediction model that integrates all road pavement and environmental parameters and variables. [6].

In order to determine the stone loss, without having the need of inspectors on the pavement, a new automated road pavement inspection system was developed and deployed by the Dutch Highway Agency in 2009 [7]. They installed sensors that are integral components of pavemetrics' Laser Crack Measuring System (LCMS) on the vehicle to generate a detailed 3D road surface. Then based on the 2D 'Stone(a)way' algorithm described by Van Ooijen et al [8], the stone loss per area was determined. The ravelling per square meter is then used to compute the percentage of stone loss in each individual 1 meter section. Once a year the agency conducts this survey. Due to the high resolution, the time horizon of more than 10 years and the 6000 km of road sections, this results in a large amount of data to be mined for prediction.

TNO has released this project. TNO (Dutch organization for Applied Scientific Research) is an independent statutory research organization in the Netherlands dedicated to applied science. [9]. It has 9 units focusing on several domains, and this project is under the purview of the Road Structures Expertise team from the Building Materials & Structures (BMS) Unit.

The thesis is organized as follows. In Section 2 the problem description is presented with more detail and research questions are formulated. Section 3 reviews literature on ravelling and service life predicting methods. Section 4 introduces the dataset and preprocessing work.

Section 5 explains approaches and methodologies implemented. Experiments and results are included in Section 6 and 7 respectively. Finally conclusions and future works are included in Section 8 and 9 respectively.

2 Problem Statement

In the Netherlands, specifically, for PA, the main damage is ravelling, which is the release of stone/minerals from the surface of PA. Most importantly, maintenance of ravelling is the most expensive one [10]. In this section, the mechanisms and causes of ravelling will be introduced.

2.1 Ravelling

Ravelling is a common form of deterioration in different asphalt pavements. Possible factors contributing to the loss of aggregate particles include inadequate binder, incorrect aggregate grading, poor adhesion between the binder and aggregate, errors in compaction, excessive wear from traffic, and deterioration caused by climatic conditions. [11].

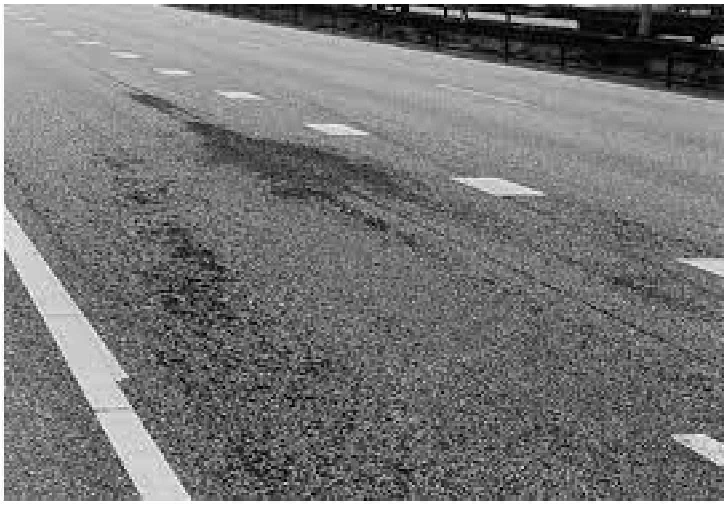


Figure 1: Ravelling of PA in the Netherlands [12]

There are two sides of failure, as we can see in the right part of Fig 2 , on the left top side is the load and on the right bottom side the resistance to this load. For a bitmounius material the material's ability to withstand the applied force changes over time, due to ageing of the material. As a result the material becomes more brittle over time and the risk of a load being more than the material can resist becomes larger.

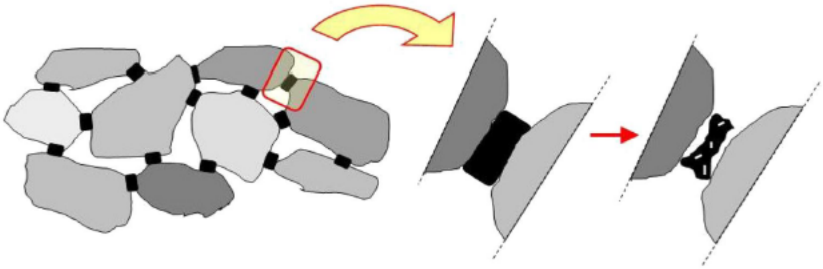


Figure 2: Mechanism of ravelling [13]

There is a wide spread in performance between different roads. Brown and Johnson indicate that roads experiencing heavier traffic or oversized vehicles are more prone to accelerated wear and ravelling, necessitating ongoing maintenance and repair efforts to sustain optimal performance [14]. Construction quality is another critical factor affecting road performance. During construction, the degree of compaction of the asphalt mixture is directly related to the road's resistance to ravelling. Kim et al. found that insufficient compaction can lead to ravelling on the road surface, while over-compaction can cause asphalt aging and cracking problems [15]. Materials also play a crucial role in influencing road ravelling. Research indicates that different types of asphalt mixtures exhibit significant differences in resistance to ravelling. For example, the use of high-quality polymer-modified asphalt (PMA) can significantly enhance the anti-ravelling performance of roads. Sullivan et al. noted that PMA materials, due to their superior adhesive properties and elastic recovery abilities, demonstrate better durability under prolonged load conditions [16].

Therefore, there is a need to predict the performance of pavements, as raveling problems reduce ride quality, increase noise, and pose a risk of windscreen damage. Additionally, it is believed that using actual field performance data could lead to more accurate predictions.

2.2 Research Question

In this master thesis project. The primary emphasis of the thesis is the prediction of ravelling, the relevant data is provided by Rijkswaterstaat. However, currently, this data is only used to describe the current state of the pavement. In order to predict its future state, it is a high priority to include the historical data and implement proper models to make asphalt lifespan predictions. Generally, the research question is "**How to investigate life span prediction through data-driven models using machine learning and deep learning techniques**".

2.2.1 Research Sub-Questions

The topic can be further broken into some sub-questions based on the methodology and models chosen. Here are some refined sub-questions:

- Will Deep Learning models surpass traditional Machine Learning models in performance?
- Does stacking segments improve prediction accuracy, given that nearby hectometers are assumed to exhibit similar stone loss trends due to their geographical proximity? By stacking them together to create a new dataset, could the prediction accuracy be improved?"
- Does the length of historical data influence the prediction accuracy?
- Will averaging over multiple meters in particular segments of the road improve the results?

3 Related Literature

3.1 The cause of ravelling

There are several factors that contribute to the development of ravelling. Wang et al. introduced spatial-temporal maps as a means to qualitatively examine and compare data, aiming to comprehend the correlation between pavement ravelling and traffic characteristics. This methodology was implemented in five different study areas within the Dutch highway network. The findings demonstrate that the movement of vehicles significantly affects the smoothness of travel. The correlation can be classified by lanes, revealing that the ravelling is uniform in both the through and auxiliary lane. [17]. In addition, ravelling will occur more frequently after a period with more freeze-thaw events. Furthermore, ravelling is also affected by the mixture composition since construction can cause a large variation in the mixture composition within a section as well as between different road sections [2]. Results from laboratory tests. Cheng et al found that increased number of UV aging cycles showed an increased trend of mass ravelling loss rate in asphalt specimens. Additionally, an increase in freeze-thaw cycles also lead to greater mass ravelling loss rate [18]. Besides, binder aging is a significant factor that has a substantial impact on the overall characteristics of asphalt mixtures and can lead to an increase in ravelling. [19]. Furthermore, the air void level in the mixture gradation also has an impact on ravelling [20]. Temperature also affects the process of ravelling. Visscher et al. tested the impact of temperature by altering the temperature during the ravelling tests. It was discovered that the occurrence of ravelling became more frequent as the test temperature increased [21].

To find the mechanism of ravelling, Ahmed et al concluded that concerning the process of ravelling in asphalt surfaces, there are two basic theories that can be applied [20]. One theory is connected to the load and is based on the stress imparted to a specific aggregate particle that exceeds the bond strength between the particle and the surrounding mastic. The other theory relies on the deterioration of the cohesive and adhesive properties of the bituminous binder as it ages, mostly owing to oxidative and UV aging. The two approaches are mutually beneficial and it is anticipated that a combination of both impacts will occur in most real-world situations.

3.2 Approaches for predicting pavement lifespan

It is of great importance to predict road lifespan to make maintenance and prevent potential economic losses. Since ravelling can cause significant hazards, such as reduced skid resistance, increased risk of accidents, and higher maintenance costs, it is crucial to understand and mitigate this issue. In order to predict pavement lifespan, different approaches are implemented to make predictions. Artificial Neural Network (ANN) has been widely discussed since the 1980s [22], and it has been widely used in other areas of pavement analysis [23]. Mostafa et al implemented ANN to forecast the Pavement Condition Index (PCI) by considering qualitative variables such as the year of inspection and the type of pavement. The study revealed that the recommended multi-step approach resulted in an optimal ANN that greatly improves the performance and accuracy of the model. Furthermore, the error of this optimized ANN is less than half of that observed in a standard ANN model. [24]. Similarly, Amjad predicted the PCI based on pavement features and the result shows that using ANN to predict PCI is a feasible and effective method [25].

Karballaeezadeh et al. used multi-layer perceptron (MLP) and radial basis function (RBF) neural networks to predict PCI by analyzing pavement surface deflections [26] in Iran. However, the Ministry of Roads and Urban Development did not utilize advanced pavement inspection devices, such as 2D/3D laser scanning and image-based approaches, due to financial constraints. Instead, a convolutional neural network approach is implemented by Spek et al to improve the classification of pavement cracks in the Netherlands [1] based on 3D pavement surface measurements, and the classification accuracy could reach 99%. Furthermore, Lin et al. employed artificial neural networks with a back-propagation approach to predict IRI (International Roughness Index) for pavement distress in Taiwan [27], the results indicate a coefficient of determination (R^2) of 0.944 between the 25 testing records of the input variable and their corresponding prediction values.

Machine Learning (ML) models are also widely used. Karballaeezadehet et al. proposed a Random Forest model to predict tensile strength (TS), compressive strength (CS), and flexural strength (FS) of roller-compacted concrete pavement (RCCP) in Iran [28], and the results outperformed other regression models. Inkoom et al. implemented Regression Tree (RT) and ANN to predict cracking condition on pavement surfaces in Florida based on the road features. The RT outperformed ANN with R^2 0.89 compared with 0.41 for ANN [29,30].

4 Dataset and Preprocessing

4.1 Dataset Description

The Dutch Highway Agency (Rijkswaterstaat) implemented pavement monitoring system based on Laser Crack Measurement System-2 (LCMS) [31]; it is also called DOS(Detectie Oppervlakte Schade / Detection Surface Damage)-LCMS. To perform the measurements, a vehicle is equipped with sensors, as we can see in Fig 3. The vehicle drives along the roads that need to be measured and the sensors are able to generate a detailed 3D road surface. Based on the 2D 'Stone(a)way' algorithm, stone loss was determined [32]. Ravelling per meter square is then used to compute the percentage of stone loss in each individual meter. Fig 4 illustrates the working mechanism of LCMS. As depicted in the right picture of Fig 4, the white dots indicate a loss in the number of stones.

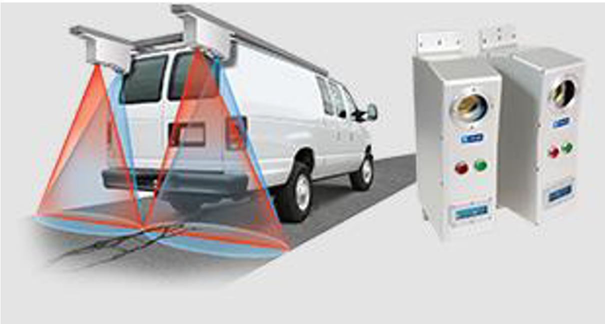
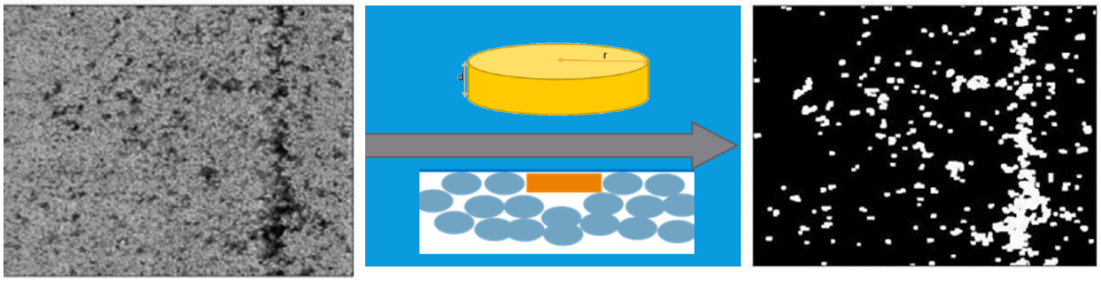


Figure 3: LCMS-2 Vehicle



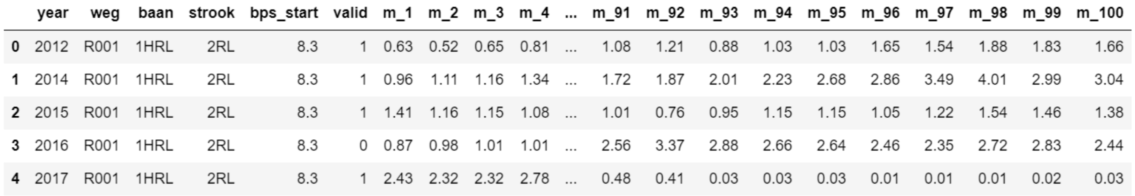
3D Data

2D 'Stone(a)way' algorithm

Ravelling surface

Figure 4: The working mechanism of LCMS [33]

Subsequently, the data is transferred to an Excel spreadsheet, comprising 140 columns and encompassing historical data spanning from 2011 to 2022. I streamlined the dataset to retain only the pertinent columns and the dataset could be seen in Fig 5. Our focus is limited to specific columns that include the features related to road imports. These are:



	year	weg	baan	strook	bps_start	valid	m_1	m_2	m_3	m_4	...	m_91	m_92	m_93	m_94	m_95	m_96	m_97	m_98	m_99	m_100
0	2012	R001	1HRL	2RL	8.3	1	0.63	0.52	0.65	0.81	...	1.08	1.21	0.88	1.03	1.03	1.65	1.54	1.88	1.83	1.66
1	2014	R001	1HRL	2RL	8.3	1	0.96	1.11	1.16	1.34	...	1.72	1.87	2.01	2.23	2.68	2.86	3.49	4.01	2.99	3.04
2	2015	R001	1HRL	2RL	8.3	1	1.41	1.16	1.15	1.08	...	1.01	0.76	0.95	1.15	1.15	1.05	1.22	1.54	1.46	1.38
3	2016	R001	1HRL	2RL	8.3	0	0.87	0.98	1.01	1.01	...	2.56	3.37	2.88	2.66	2.64	2.46	2.35	2.72	2.83	2.44
4	2017	R001	1HRL	2RL	8.3	1	2.43	2.32	2.32	2.78	...	0.48	0.41	0.03	0.03	0.03	0.01	0.01	0.01	0.02	0.03

Figure 5: Example of HiRav data [34]

- Weg: The name of the road, there are 48 different roads with 1748.5km in total.

- Bps_start , Bps_stop: The measuring begins and ends at a spot, with a length of around 100 metres.

- Baan: If the string finishes with 'L' (for example, '1HRL'), it indicates that the road is in a left direction. Additionally, the values of bps start from a larger number and decrease, and vice versa.

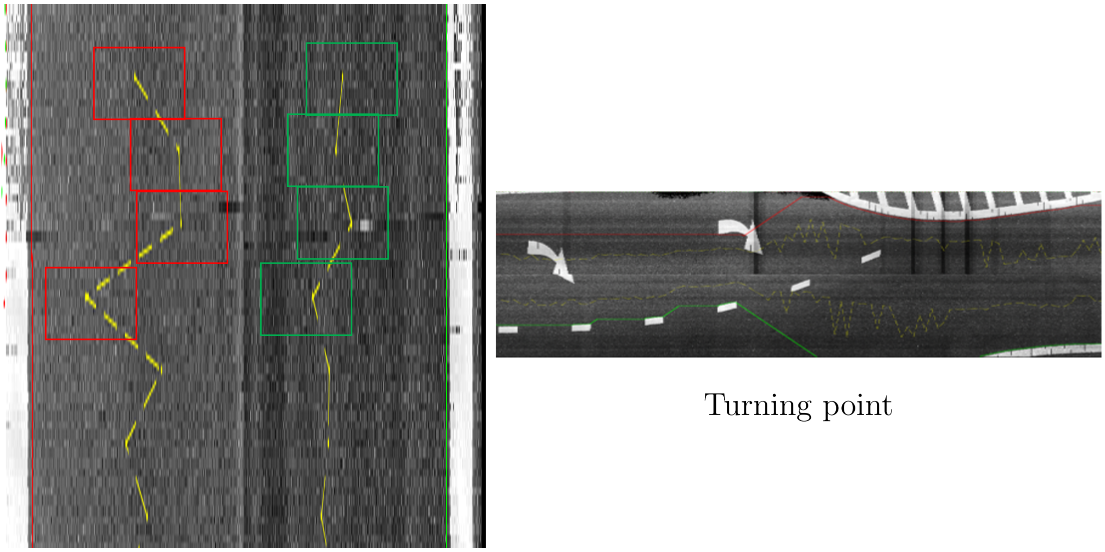
- Strook: Different lanes within a road. Typically, the lane that experiences a high volume of traffic tends to have a shorter lifespan.

- m_i: The stone loss percentage per square meter section is represented as follows: for example, $m\_91 = 1.05$ indicates that the stone loss percentage at the 91st meter square is 1.05%.

4.2 Dataset Preprocessing

Since the initial data includes 140 columns and it cannot be used directly due to unqualified entries (string values, missing values), it needs to be cleaned and formatted. Therefore, various data preprocessing techniques are employed to implement models. These techniques include:

- Valid Data: Only use valid data will be used, whereas invalid data is not considered. Fig 6 provides instances of measurements that are considered invalid, with the colour green indicating validity and red indicating invalidity due to the excessive degree of bending in the tested section. In the CSV file, in the 'valid' column, 1 represents valid and 0 represents invalid.



Turning point

Discontinuous measurement

Figure 6: Examples of invalid measurements [35]

- Average sv_l and sv_r: we average the performance of sv_l and sv_r instead of sv given that sv contains outliers, it is crucial to acknowledge that sv_l and sv_r represent the raveling in the wheel paths, while sv encompasses the entire width of the lane.

- Numeric string data: As models do not accept strings as valid input, we convert columns containing string data into numeric format for training purposes.

- Average multiple measurements: The road sections with more measurements in a single year are the measurement calibration sections, where the measuring vehicles demonstrate their performance. The observed variation in this context provides insights on the extent of measurement variability. We average the corresponding data.

Finally, unrelated columns have been deleted and all the columns used for models have been formatted properly.

5 Methodology

This project aims to forecast the future percentage of stone loss using a regression program. In order to obtain precise forecasts, both Machine Learning (ML) and Deep Learning (DL) methodologies are utilized. Machine Learning (ML) is a specific field within the broader domain of artificial intelligence (AI) that is dedicated to constructing systems capable of acquiring knowledge and making informed choices by analyzing and interpreting data. ML systems are not explicitly programmed for tasks; instead, they are trained using extensive data to recognize patterns and make predictions or decisions. It is also adept at handling non-linear behavior, which we expect to see in ravelling. In section 5.1, traditional machine learning models implemented for this project from Scikit-learn will be introduced. Since it is a multi-output regression problem, some ML models could not be implemented.

Deep Learning (DL) is a subfield of machine learning, which is itself a subfield of artificial intelligence. The primary focus is on utilizing deep neural networks, which consist of multiple layers, to effectively represent intricate patterns found within extensive datasets. The revolutionary nature of this field has driven advancements in diverse areas, including computer vision, natural language processing, healthcare diagnostics, and autonomous driving. In Section 5.2, two Deep Learning models and their key concepts are introduced.

Then several experiments are conducted based on these two approaches to improve prediction accuracy. These include averaging scores over 10 meters, using different ML and DL methods, and incorporating multi-year scores. Section 6.2 will explain the different experiments in detail.

5.1 Machine Learning Models

5.1.1 Random Forest

Random forest is a very classic ensemble learning method. It belongs to the class of bagging algorithms. Random forests are well-suited for problems and datasets with a large number of features as each decision tree in a random forest is constructed using a randomly selected subset of the features [36]. This means that each tree is trained on different combinations of features, which helps in dealing with datasets with many features. It reduces the risk of overfitting and ensures that even less important features get considered. It is used for regression (denoted as RandomForestRegressor) to make stone loss predictions.

5.1.2 Multilayer Perceptron

The Multilayer Perceptron network models are widely utilized network architectures in various research applications [37]. The architecture comprises multiple input layers, one or more hidden layers, and output layers. (Fig 7). Every layer contains multiple processing units, and each unit is completely interconnected with weighted connections to units in the following layer. The MLP maps m inputs to n outputs using nonlinear activation functions. [38]. The input layer consisting of a collection of neurons that encode the input characteristics. Every individual neuron within the hidden layer undergoes a transformation on the summation of weight factors and input values which are from previous layers $w_1x_1 + w_1x_1 + \cdots + w_mx_m$, by applying an activation function, the weights are initialized randomly and will be updated by using gradient descent back propagation (BP) algorithm. The output layer obtains inputs from the final hidden layer and converts them into output values.

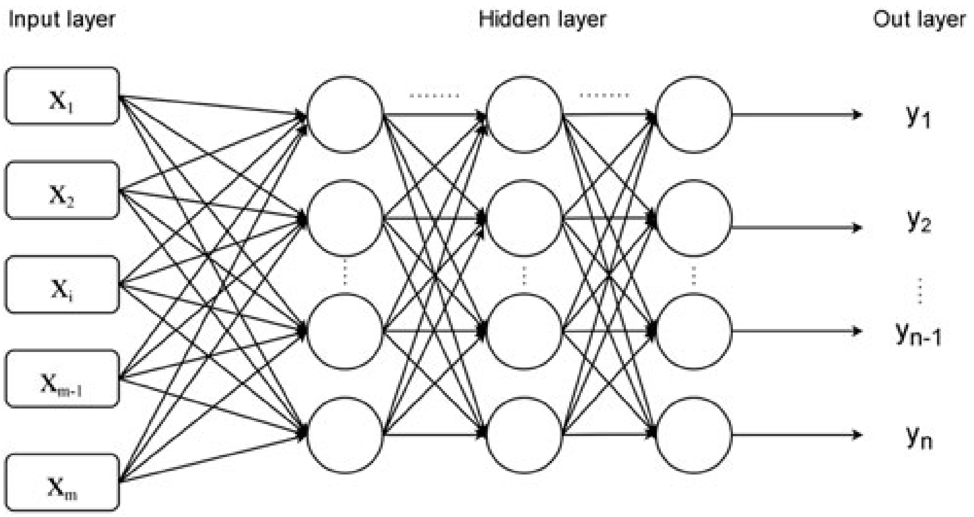


Figure 7: Architecture of MLP

5.1.3 XGBoost

Gradient boosting decision trees can take advantage of multiple weak decision trees to improve the overall performance of the model [39]. However, the traditional implementation is very time-consuming. We use LightGBM (i.e., Light Gradient Boosting Machine) for regression and prediction (denoted as LightGBMRegressor). LightGBM is fast and efficient, has low memory usage and can be parallelized [40].

5.1.4 Linear SVM

Support Vector Machines (SVMs) are supervised machine learning models that use max-margin algorithms to analyze data for classification and regression analysis. Since it is a multi-output regression problem, Linear SVM rather than SVM could be implemented, Linear SVM is a highly efficient method used for applications involving high-dimensional data, such as document classification and time series analysis [41].

5.1.5 CatBoost

CatBoost is a novel gradient boosting technology proposed by Yandex Comp [42]. This is a novel gradient boosting decision tree (GBDT) algorithm that effectively handles categorical features, distinguishing it from conventional GBDT algorithms. [43]. And it can effectively be utilized with various types and formats of data [44] and successfully be applied in various fields such as time series data [45, 46].

5.1.6 Hyperparaeter Optimisation

Hyperparameter tuning is an essential part of the data science process. Properly implemented hyperparameter tuning can indeed enhance the performance of the model [47]. This can be categorised into into two categories: model parameters for machine learning, such as pooling size and number of hidden layers, and training parameters for deep learning, such as regularization, learning rate, and batch size. [48]. This approach will be implemented after running the fundamental models.

5.2 Deep Learning

5.2.1 Multilayer Perceptron

Multilayer Perceptron is a well-established form of Artificial Neural Network that comprises several layers, such as an input layer, hidden layers, and an output layer. Every layer is composed of neurons that are fully connected. [49]. MLP is a feedforward neural network, meaning it does not have feedback connections where outputs of the network are fed back into itself. This method updates the weights through a process called backpropagation. During training, the network makes predictions, and the error between the predicted and actual outputs is calculated. The backpropagation algorithm then adjusts the weights by propagating this error backward through the network, using gradient descent to minimize the error over time.

Assume an L-layer neural network, where the l-th layer ($l = 1, 2, \ldots, L$) consists of $n^{(l)}$ neurons. Let $\mathbf{a}^{(l)}$ denote the activation vector of the l-th layer, $\mathbf{W}^{(l)}$ the weight matrix, and $\mathbf{b}^{(l)}$ the bias vector. The computation for the l-th layer is represented as:

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)} \tag{1}$$

$$\mathbf{a}^{(l)} = f^{(l)}(\mathbf{z}^{(l)}) \tag{2}$$

Here, $\mathbf{a}^{(0)}$ represents the input vector, and $f^{(l)}$ is the activation function for the l-th layer (typically a nonlinear function such as ReLU, sigmoid, or tanh).

The description for different layers could be seen as follows:

- Input Layer: It receives input data from external sources. Each neuron in the input layer corresponds to one feature of the input data.

- Hidden Layers: It can comprise one or more intermediate layers positioned between the input and output layers. The neurons in each hidden layer establish complete connections with the neurons of the preceding layer, which can be either the input layer or another hidden layer. Every individual neuron performs a calculation where it multiplies each input by a specific weight, adds them together, and then applies an activation function (such as sigmoid or ReLU) to generate an output. This output is then used as input for the next layer.

- Output Layer:It receives input data from external sources. Each neuron in the input layer represents a distinct characteristic of the input data.

5.2.2 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a widely used subset of neural networks that fall under the broader category of deep learning techniques. The key to their success lies in their meticulously crafted architecture, which is adept at taking into account both the local and global characteristics of the input data. [50]. A key benefit of CNNs is their capacity to autonomously acquire valuable features from complex data sets without the need for manual feature engineering [51]. Several recent studies have utilized Convolutional Neural Networks (CNNs) for time series forecasting tasks [52].

In Convolutional Neural Networks (CNNs), there are 1D, 2D, and 3D CNNs. 2D CNNs are primarily used for processing two-dimensional spatial data, such as in image processing and computer vision tasks. They extract features by applying convolution operations across the height and width of the input data. On the other hand, 3D CNNs are used for three-dimensional spatial data, such as video data or medical imaging, allowing for convolution operations across the height, width, and depth of the input data.

In this project, 1D CNNs approach is implemented, which is suitable for processing one-dimensional spatial data, such as time series data and signal processing. By sliding convolutional kernels along one dimension of the input data, 1D CNNs can effectively capture and learn patterns and features within sequential data.

1D CNNs is a convolution operation performed in a single direction (usually the time axis). The principle of 1D convolution is to perform convolution on the input by sliding a fixed-size convolution kernel (i.e., filter). In one-dimensional convolution, the convolution kernel is a one-dimensional tensor of length kernel_size, which is used to filter each time step of the input. The size of the convolution kernel affects the output shape after convolution, which can be calculated using the following formula:

$$outputlength = \frac{inputlength - kernelsize + 2 * padding}{stride} + 1 \tag{3}$$

- Input length: The length of each input vector it should be based on our data structure.

- Output length : The length of each output vector, it should be either 10 (average every 10 meters for each hectometer), 100 (individually) or 150 (stacking hectometer).

- Kernelsize: The number of elements the kernel covers along the input dimension.

- Padding: Padding is used to adjust the input sequence so that the kernel can be applied in a way that achieves the desired output dimensions.

- Stride: Stride is the number of positions the filter moves along the input sequence at each step..

Conv1D Layers can set several parameters, such as the size of the convolution kernel, the stride length, the filling method, the activation function, and so on. By adjusting these parameters, the time series features in the input data can be effectively extracted for subsequent tasks such as classification and regression.

Suppose the input data is x, the convolution kernel is w, the offset is b, the step size is s, and the size of the padding is p. For one-dimensional convolution, we can represent both the dimensions of x and w as length, for example:

$$x = x_1, x_2, x_3, \ldots, x_n \quad w = w_1, w_2, w_3, \ldots, w_m \tag{4}$$

And the output for y_i should be calculated as follows:

$$y_i = \sum_{j=1}^{m} w_j \cdot x_{(i-1)\times x+j} + b \tag{5}$$

The output of the first layer would be the input of the second layer, after stacking 1D CNN neural network layers, the output dimensions are 10, 100, or 150, depending on the chosen data structure. An simple explanation of convolution process could be seen in Fig 8.

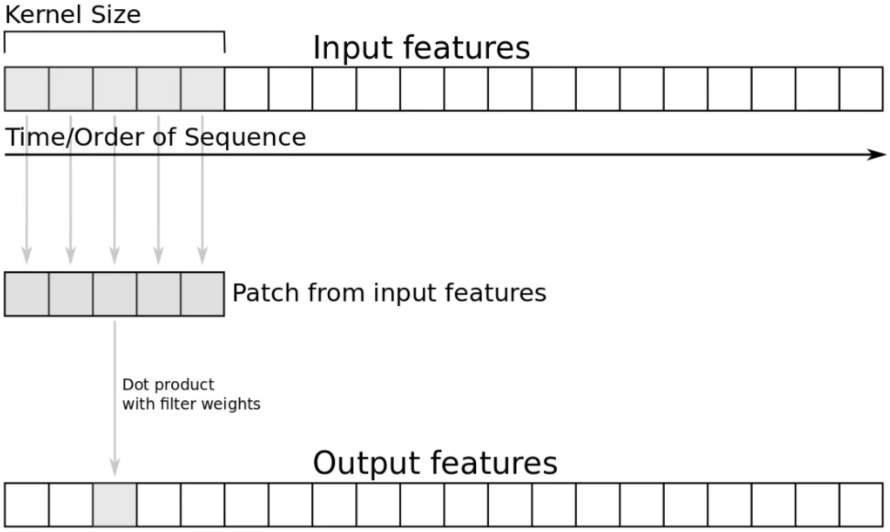


Figure 8: Explanation of Convolution process

5.3 Evaluation Metrics

As this is a regression problem, we will utilize Mean Squared Error (MSE) and R-square (R^2) as our evaluation metrics. These metrics are chosen to effectively gauge the predictive performance of the model.

MSE equals the the mean value of all the error square, and error is the difference between true and predicted values, the equation could be seen as follows :

$$MSE = \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N} \tag{6}$$

R-square (R^2) is the square of the correlation coefficient between the predicted values and the actual values of the dependent variable in a regression model. In simpler terms, it's a measure of how well the regression line (or curve) fits the observed data points. The value ranges from 0 to 1, with 1 indicating a good prediction result. The equation of R-square could be seen as follows:

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \tag{7}$$

- y_i: represents each true value of the dependent variable y, which is the percentage of stone loss.

- $\hat{y}_i$:represents each predicted value of the dependent variable y from the model.

- $\bar{y}$: represents the mean value of the dependent variable, in our case is the mean individual stone loss percentage.

- N: The number of observations, in our case is individual hectometer.

6 Experiments

6.1 Experimental setup

6.1.1 Software

The implementation is created in Python 3.9. Pytorch 2.0.1 is used as a deep-learning framework with Pytorch as programming software . CUDA 11.8 with cuDNN 7 is employed to provide GPU accelerated functionality.

6.1.2 Hardware

Training and testing of our models is done using a GeForce GTX 3090 Ti 24GB graphics card. It can significantly accelerate the training speed of machine learning models, especially deep learning models, which are computationally expensive.

6.2 Experiments

6.2.1 Run traditional ML Models

The application of traditional machine learning models are explored. These models, such as Random Forest, XGBoost, and Support Vector Machines, provide foundational techniques for analyzing and predicting data patterns without the complexity of deep learning architectures. Their performance in various tasks will be evaluated.

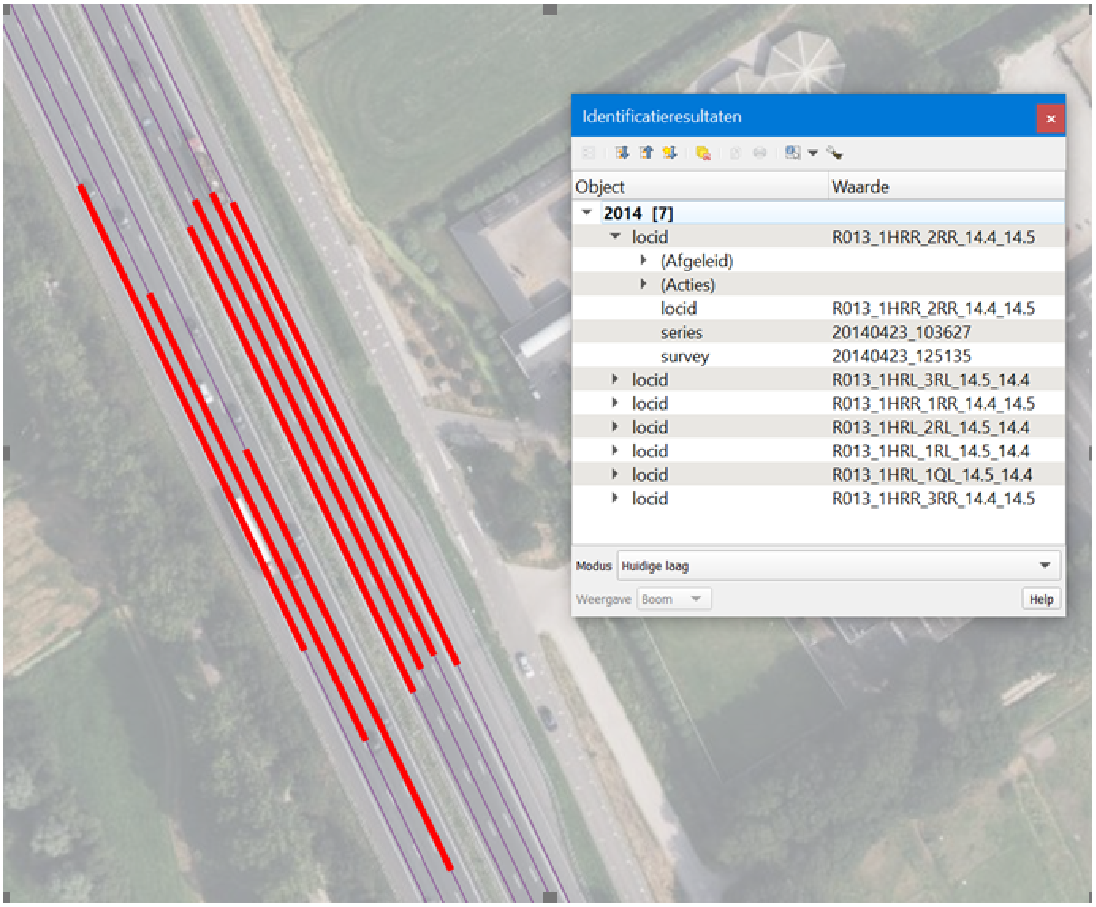
6.2.2 Average LCMS Data

It should be noted that location data can vary over the years due to inconsistencies in GPS data collection methods and accuracy. Therefore, measurements for specific 1-meter sections are not guaranteed to be accurate to within 1 meter. In practice, data collected in subsequent years is often not precisely aligned with previous measurements. While the general location matches, it is common for measurements to vary by up to 5 meters in either direction along the route. Additionally, maintenance could be focused at the individual meter level, and nearby sessions also need to be considered.

Fig 9 illustrates the instance that the start of a hectometer is not exactly the same for each lane measured at a specific section in 2014, demonstrating one aspect of location inaccuracy.

Fig 10 shows the stone loss percentage over the years. Upon evaluating the peaks in the figure, it appears that there is a shift of 2 to 4 meters in measurements from 2021 to 2022.

Therefore, this ‘location inaccuracy’ explicitly is taken into account, for instance by averaging the stone loss data over 10 meters.



Identificatieresultaten

Object	Waarde
2014 [7]	
locid	R013_1HRR_2RR_14.4_14.5
(Afgeleid)	
(Acties)	
locid	R013_1HRR_2RR_14.4_14.5
series	20140423_103627
survey	20140423_125135
locid	R013_1HRL_3RL_14.5_14.4
locid	R013_1HRR_1RR_14.4_14.5
locid	R013_1HRL_2RL_14.5_14.4
locid	R013_1HRL_1RL_14.5_14.4
locid	R013_1HRL_1QL_14.5_14.4
locid	R013_1HRR_3RR_14.4_14.5

Modus: Huidige laag

Figure 9: Example of different start location for each line [53]

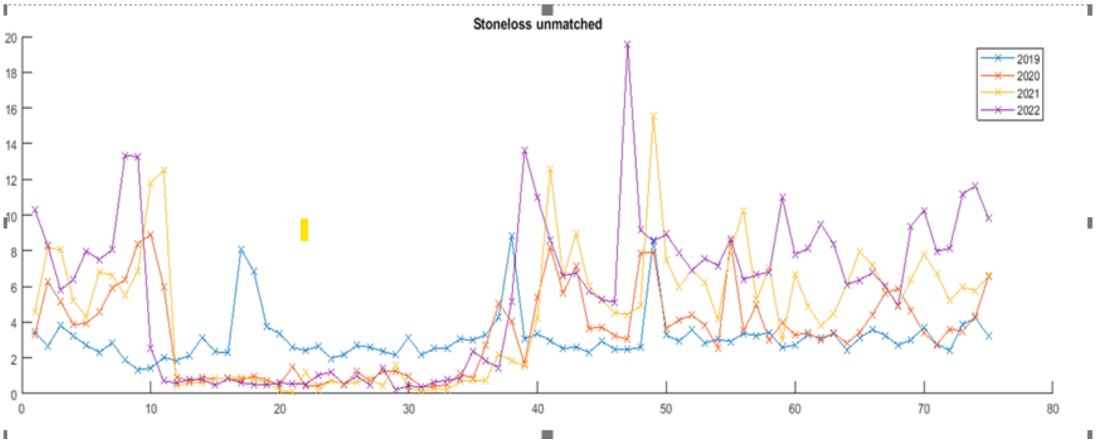


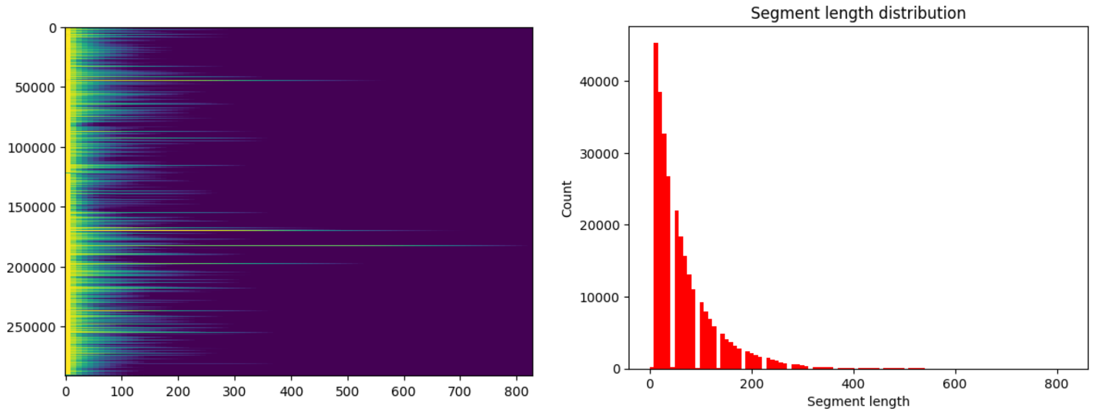
Figure 10: Measurement bias by years [53]

6.2.3 One-Hot Encoder

Since there are lots of unique values in road features, even if we convert them into numeric, the models might not be able to work directly with categorical data. One-hot encoding converts these categories into a numerical format that the algorithms can handle. One-hot encoding converts categorical data into a binary vector representation. A binary vector represents each category, where only one element represents that category will be 1 and all other categories for this vector will be presented as 0.

6.2.4 Stack nearby hectometer

Initially, there is an assumption that nearby hectometers would exhibit similar stone loss percentages trend due to their geographical proximity. Consequently, the nearby hectometers with consecutive BPS_Start values are stacked. During this process, directional discrepancies, as the data structure differs for hectometers in the left and right directions are addressed. The distribution of stacked segments could be seen in Fig 11.



Stacked Lines' distribution

Count of line length

Figure 11: Data for stacked lines

As shown in the figure, the data distribution of the segments is very sparse (the dark blue color indicates parts with no corresponding data), and there are significant differences among the data points, the longest segment spans 82 km, with most segments being shorter than 20 km. To expedite the training process, the focus is on segments shorter than 15 km, which comprise 90% of data. Without this filtering, the training time would be significantly longer due to the larger data size, with each vector containing 820 data points instead of 150. The results could be seen in Fig 12. It is obvious that after filtering the lengths of the segments are quite similar.

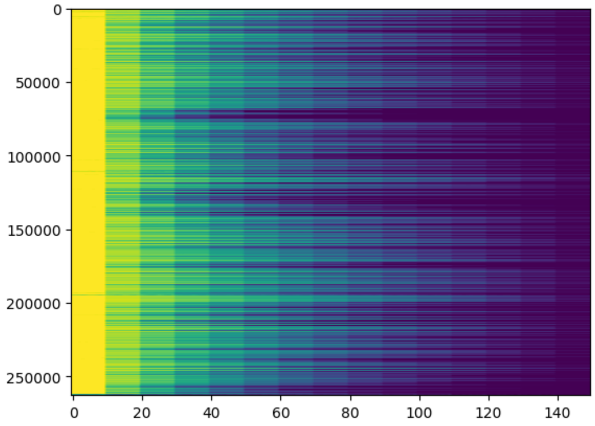


Figure 12: Stacked Lines' distribution after filtering

6.2.5 Stack multiple years

Building on the hectometer data stacking, data across multiple years are stacked, by which it allows us to capture long-term trends and seasonal variations, significantly boosting the model's predictive capability. Three, four, five, six years temporal data are stacked. Table 1 illustrates the number of datasets available after stacking by years. It is evident that as the years progress, the number of datasets decreases. This trend is attributed to the presence of missing values across different years, which hinders the stacking process.

	One Year	Three Years	Four Years	Five Years	Six Years
Dataset Length	292075	181420	151547	123840	97840

Table 1: Number of datasets for after stacking by years

7 Results

7.1 Results for Machine Learning Models

Historical data from 2011 to 2020 is used for training, data from 2021 for validation, and data from 2022 for testing. The results could be seen in Table 2. It's clear that predicting each individual meter for every hectometer yields relatively low accuracy. However, when we average every 10 meters, the results marginally improve. Additionally, incorporating extra road features using a one-hot encoder further enhances the accuracy, though there is still room for improvement.

Model	Normal	Average every 10 meters	Average + one hot encoder
Random Forest	0.6031	0.665	0.647
Multilayer Perceptron	0.5842	0.5999	0.6481
XGBoost	0.6082	0.6437	0.6879
Linear SVM	0.4049	0.6189	0.6211
CatBoost	0.6087	0.6584	0.6751

Table 2: Best results for each methodology based on hyperparameter fine tuning

Moving onto visualizing the results based on the best model identified in Table 2, deeper insights into the model's performance will be shown. Fig 13 illustrates the relationship between the true values and the predicted values for several roads. In this scatter plot, the x-axis represents the true values, while the y-axis represents the predictions made by the model.

The blue line in the figure corresponds to the equation y=x, which serves as an ideal reference line. Points that lie exactly on this line indicate perfect predictions, where the predicted values match the true values exactly. Ideally, all points should be as close to the blue line as possible, signifying that the model is making accurate predictions.

However, upon examining the scatter plot, it is evident that many points deviate significantly from the blue line. This indicates that the model has made numerous over-predictions and under-predictions. The spread of these points reflects the variability and inconsistencies in the model's performance across different roads. Despite this, it is worth noting that for certain roads, the predictions appear to be relatively accurate, as evidenced by points that are closely aligned with the blue line.

The scatter plot thus highlights both the strengths and weaknesses of the model. It shows that while the model can achieve good predictions in some cases, there is a considerable number of instances where the predictions are less reliable. This visualization is crucial for understanding where the model performs well and where there is room for improvement.

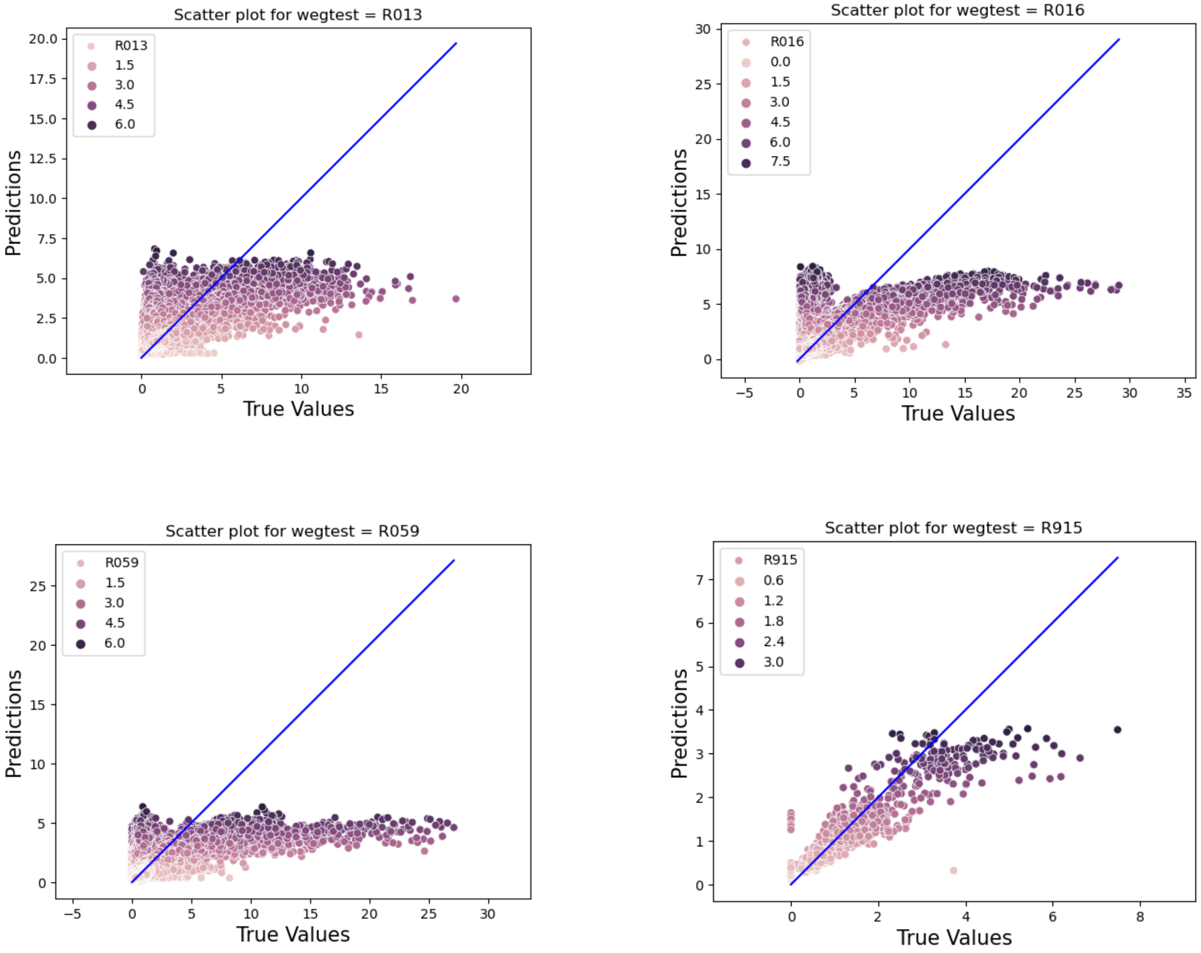


Figure 13: Comparison of model prediction versus true values for roads

7.2 Results for Deep Learning Models

7.2.1 MLP and CNN

Initially, the data was not stacked but instead averaged over every 10 meters, with different years used for the predictions. A random split approach is utilized to partition the dataset, allocating 80%, 10%, 10% for training, validation, and testing respectively. The outcomes of this approach are presented in Table 3.

From the table, it is evident that both the Multilayer Perceptron and Convolutional Neural Network models outperform traditional machine learning models in terms of predictive accuracy. Moreover, as the number of years included in the training data increases, we observe a corresponding rise in the R-square values. This indicates that incorporating more historical data enhances the model's ability to explain the variance in the target variable, leading to more accurate predictions. However, these improvements are still relatively modest, leaving ample room for further enhancement.

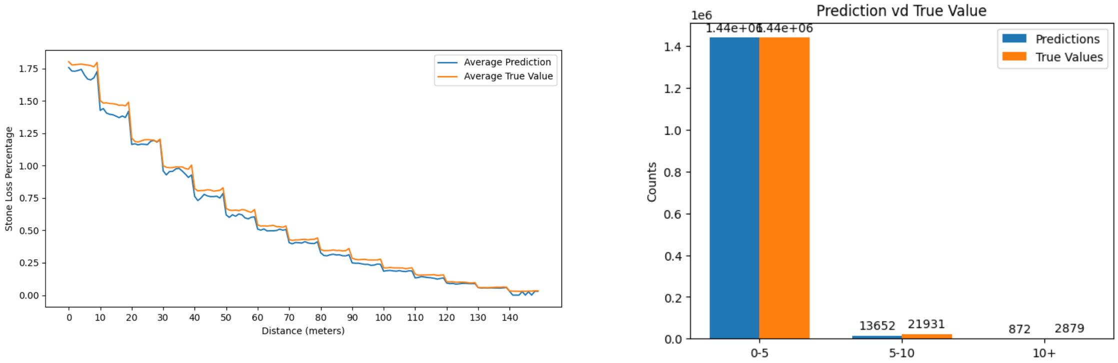
	MLP			1DCNN		
	Training	Validation	Test	Training	Validation	Test
Five Years	0.7432	0.7332	0.7143	0.7045	0.6945	0.7181
Four Years	0.7355	0.7242	0.7123	0.6935	0.7032	0.7015
Three Years	0.7229	0.7154	0.7008	0.6886	0.7155	0.6963
Two Years	0.6499	0.6178	0.6645	0.6857	0.5334	0.664
One Year	0.6558	0.6061	0.6286			

Table 3: Best results for MLP and CNN by years

7.2.2 Stack segments

After stacking the segments for CNN, the best performance achieved with an R Square value at 0.7234. The corresponding results are visualized in Fig 14.

As depicted in the left figure of Fig 14, the model generally performs adequately on average, yet there are slight discrepancies evident between the blue (predictions) and orange (true values) lines. As shown in the right figure of Fig 14, the x-axis represents the stone loss percentage intervals, while the y-axis depicts the number of counts (in thousands) within each



Overall prediction in ten meters level

Prediction Results vs True Value in intervals

Figure 14: Results for stacking the segments

interval. Upon examining the distribution of predictions, it becomes apparent that the model struggles notably with forecasting larger values. Notably, there are only 872 instances predicted in the 10+ interval, where ideally there should be 2879. Similarly, in the 5-10 interval, there is a noticeable gap of 8369 between the predicted and true values.

In detail, as depicted in Fig 15, some results are picked and visualized, the model exhibits a general deficiency in predicting locations with high stone loss percentages, often favoring those with lower percentages instead. This is because the training set contains many lines with a lower stone loss percentage and only 0.5% of the lines have a high stone loss percentage, which prevents the model from learning the trend for high stone loss effectively. However, the model effectively tracks maintenance trends, as evidenced in the top-left figure of Fig 15, a significant drop in values corresponds to maintenance activities in 2020, which the predictions also reflect.

The remaining three panels illustrate a notable disparity between predicted and actual values for locations experiencing high stone loss percentages. Overall, while the model demonstrates competence in following maintenance patterns, it struggles with accurately predicting locations where stone loss is significant due to a lack of sufficient corresponding data in the training set.

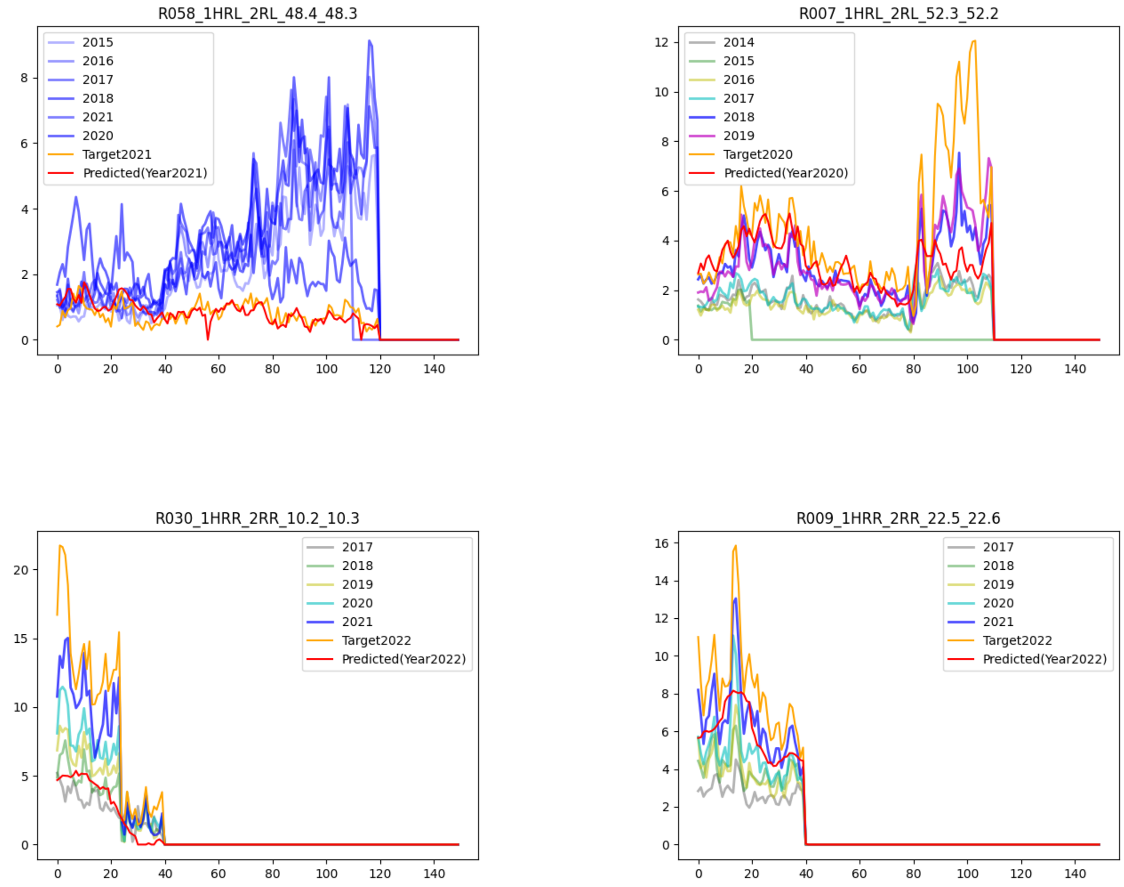


Figure 15: Prediction for places that exhibit high stone loss percentage

7.2.3 Over sample the data

Initially, challenges were encountered in predicting areas with severe damages characterized by high stone loss percentages. The model might not effectively learn the trends in these areas due to their low representation in the training set, which accounts for only about 1% of the data. To address this, oversampling strategy is employed.

I increased the representation of critical data points by oversampling instances with stone loss percentages between 5-10% by five times. Additionally, for areas where the stone loss percentage exceeds 10%, which are typically the focus of maintenance efforts, these instances are over sampled by ten times. The number of training data increases dramatically as we can see in Table 4. This approach ensures that the model better learns the patterns associated with significant stone loss, thereby improving its predictive accuracy in areas most affected by severe damage. Results could be seen in Fig 16 and Table 5.

In Table 5, in the CNN Architecture column, the number of square brackets in each row indicates the number of 1D CNN layers. Each set of four digits within the square brackets represents the channel in, channel out, filter size, and stride for the corresponding 1D CNN layer respectively. In the MLP Architecture column, the number of values indicates the number of layers, and each value represents the number of neurons in the corresponding layer.

	Dataset length originally	Training data originally	Training data after oversampling
One Year	292075	217660	
Three Years	181420	145136	537865
Four Years	151547	121237	508060
Five Years	123840	98784	457500
Six Years	97840	78272	394575

Table 4: Comparison of the number of training data

	Train	Validation	Test	Test	Batchsize	CNN	Dropout	MLP
	R Square	R Square	R Square	Loss		Architecture	Rate	Architecture
Three Years	0.8493	0.8212	0.8213	0.301	2048	[3,32,7,2], [32,32,7,2], [32,16,5,2]	0.4	[240,256,150]
Four Years	0.8385	0.8183	0.8067	0.3264	4096	[4,32,7,2], [32,32,7,2], [32,32,5,2], [32,16,3,2]	0.2	[112,256,150]
Five Years	0.863	0.8161	0.8008	0.3109	2048	[5,32,7,2], [32,32,7,2], [32,32,5,2], [32,16,3,2]	0.2	[112,256,150]
Six Years	0.9029	0.8806	0.8736	0.2006	2048	[6,32,7,2], [32,32,7,2], [32,16,5,2]	0.4	[240,256,150]

Table 5: Best model's configuration by years

It is obvious from Table 5 that the best result witnessed is 0.8736 for R^2 for CNN model, this is the final model with model configuration as follows.

- **Batch Size**: 2048

  - This indicates that during training, 2048 samples will be processed in each batch.

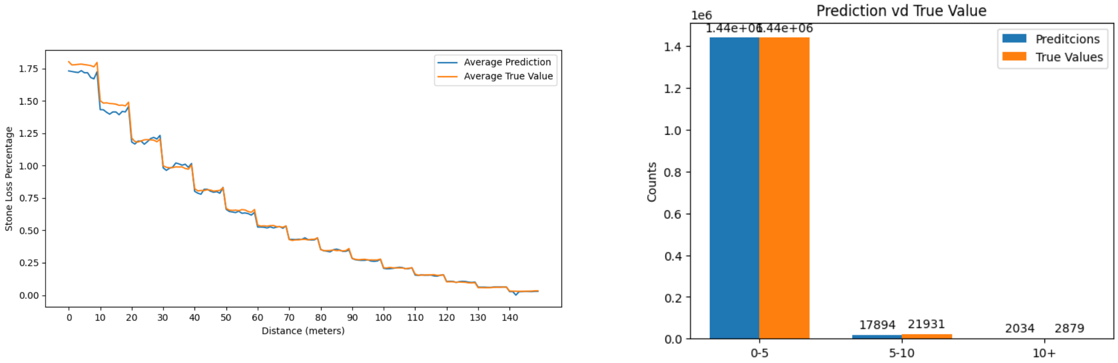
- **Convolutional Layers**:
  - **Layer 1**: 6 input channels, 32 output channels, kernel size of 7, stride of 2.
  - **Layer 2**: 32 input channels, 32 output channels, kernel size of 7, stride of 2.
  - **Layer 3**: 32 input channels, 16 output channels, kernel size of 5, stride of 2.
  - These layers apply convolution operations to the input data, with specified kernel sizes and strides, transforming the spatial dimensions and depth of the input tensor.

- **Fully Connected Layers**:
  - **Layer 1**: 240 units
  - **Layer 2**: 256 units
  - **Layer 3**: 150 units
  - These dense layers are applied after the convolutional layers to perform the final transformation and classification of the data.

- **Dropout Rate**: 0.4
  - During training, 40% of the units in the layers will be randomly set to zero to prevent overfitting by encouraging the model to learn more robust features.

- **Batch Normalization**: Enabled
  - Batch normalization will be applied after each convolutional and fully connected layer. Normalizing the inputs of each layer aids in stabilizing and expediting the training process.

As we can see in the left figure of Fig 16, the results for predicting stacked hectometers are averaged, and the two lines for predicted and true results are very close to each other, with only a slight difference compared to the smaller difference shown in Fig 14. As we can see on the right figure, where X-axis presents the stone loss percentage interval, now the model has a better capability to predict high stone loss percentage with 2034 counts in the 10+ interval compared to only 872 counts before. Similar for the 5-10 stoneloss interval, which increased by 4332 counts in this interval.



Overall prediction

Prediction Results vs True Value in intervals

Figure 16: Best Results for stacking the segments

As shown in Fig 17 below, the model demonstrates a strong capability to predict high stone loss percentages, as indicated by the comparison between the yellow line (representing real data for the corresponding year) and the red line (representing predictions). Although there are some instances where the gap between the predicted and actual values is relatively large, the model consistently predicts high values or their nearby sessions, effectively providing early warnings for maintenance.

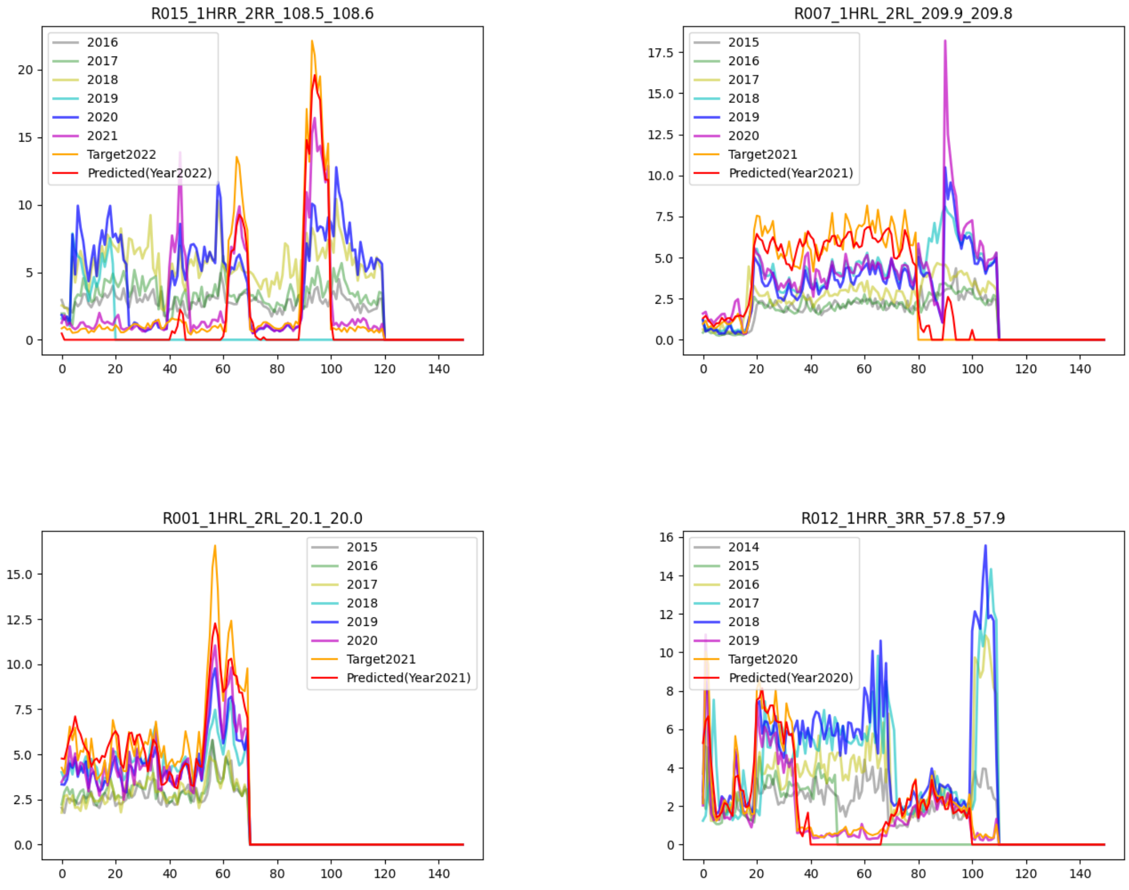


Figure 17: Prediction for places that exists high stone loss percentage

7.3 Discussion of Results

Hereby, the best performance for both traditional Machine Learning models and Deep Learning models is compared in Fig 18 below.

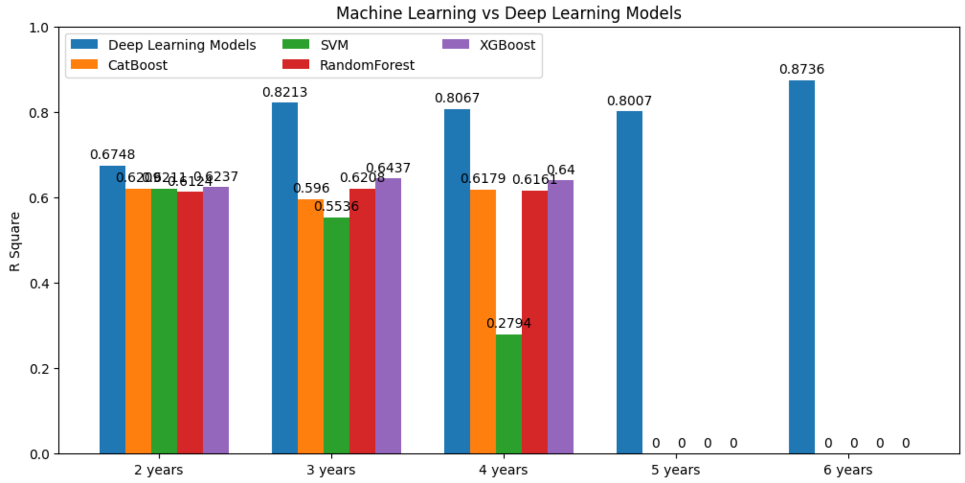


Figure 18: Best performance for each model, 0 means experiment missing

It is evident that Deep Learning models significantly outperform traditional Machine Learning models. Moreover, as the number of years for prediction increases, the performance of Deep Learning models continues to improve. In contrast, the performance of traditional Machine Learning models does not show similar improvement over time. This is primarily because Convolutional Neural Networks (CNNs) is used as the deep learning approach. CNNs excel at capturing the spatial features of time series data. Consequently, with more years of historical data, the model can learn and identify trends more effectively.

Additionally, the architecture of CNNs allows them to handle complex patterns and dependencies in data, which traditional Machine Learning models struggle to do. As the dataset grows with each passing year, the CNN's ability to generalize and make accurate predictions enhances significantly. This continual improvement is less pronounced in traditional Machine Learning models, which often reach a performance plateau regardless of the increasing amount of historical data.

In summary, the superior performance of Deep Learning models, particularly those employing CNNs, is attributed to their advanced capability to capture intricate patterns in extensive datasets. This advantage becomes increasingly apparent as the dataset expands over the years, leading to progressively better predictive accuracy compared to traditional Machine Learning models.

Code

All of our code can be downloaded via this link. With this link you can access the code.

8 Conclusion

In our project, both traditional Machine Learning and Deep Learning models are implemented to predict stone loss percentage. Initially, it is obvious that ML models yielded unsatisfactory results and struggled to forecast severe damage, even with techniques like one-hot encoding or using multiple years for predictions.

Transitioning to Deep Learning, it is witnessed that both Multilayer Perceptron and Convolutional Neural Network models initially outperformed the ML models. Further experiments using stacked architectures and CNN models did not significantly enhance performance. However, when oversampling techniques applied to address severe damage instances, the results dramatically improved to an impressive 0.87, demonstrating the model's capability to predict severe damages accurately.

For the sub questions in this project. In comparison to ML models, Deep Learning models showcased superior ability to capture temporal and spatial data trends and perform well, especially in predicting severe damages.

Moving onto years, as observed in Tables 3 and 5, there is a clear trend showing that prediction accuracy improves with the increase in the number of years of historical data. This indicates that the more years of historical data that are incorporated, the higher the observed prediction accuracy.

Averaging over multiple meters in specific segments of the road improves the results, as shown in Table 2. Consequently, averaged meter data file is used in Deep Learning models to enhance performance. Finally, stacking nearby segments did not improve prediction accuracy. However, when data indicating severe damages is oversampled, the accuracy increased dramatically, which means the model could predict next year's stone loss percentage effectively.

9 Future Work

Even though our model effectively predicts stone loss, there are additional steps we can take to further improve its performance. Implementing these enhancements will help refine predictions and provide more accurate and actionable insights.

- Hyper parameter fine-tuning: Hyperparameter tuning is a critical aspect of training deep learning models. It involves adjusting various parameters of the model, such as learning rate, batch size, number of network layers, number of filters, etc., to enhance the model's performance. Even if we modified some parameters manually(batchsizes, number of layers, dropout rate), However, manually tuning these parameters can be time-consuming and may not yield the best results. Therefore, automated hyperparameter optimization methods, such as genetic algorithms, become particularly important. Genetic algorithms are heuristic optimization methods that mimic the process of natural selection. It is a commonly used optimization method that involves a directed random search approach [54]. This is particularly advantageous for intricate optimization problems with a large number of parameters, where obtaining analytical solutions is challenging [55]. It finds the optimal set of parameters through operations like crossover, mutation, and selection. Since we didn't implement this approach in this project, we believe that the performance could be improved if this approach is implemented.

- Additional Road Features: Since only a few road features(Weg, Baan, Strook, Bps_start) are considered, apparently other features such as traffic flows, road conditions, environmental factors and etc could provide more comprehensive insights into road usage and safety. By incorporating these additional features, the analysis can be enhanced and more informed decisions can be made. For example traffic intensities data from INWEVA contains detailed traffic information that can be integrated with our current LCMS data. By incorporating this data, valuable road features can be added, and the importance of various traffic patterns can be captured. It could also enhance the model's performance and capture the critical aspects of ravelling.

- Stack additional years: In current analysis, data from two to six years of historical records is stacked. The findings indicate that incorporating more years of data can lead to improved results. Therefore, one of the key future directions is to extend the timeframe and stack additional years of historical data. This enhancement has the potential to significantly boost model performance and provide deeper insights.

Acknowledgments

Firstly, big thanks to Greet Leegwater, the leader of The Asphalt and Road Construction team, without her kind help in terms of onboarding session and daily supervision the project could not run so smoothly. To Shang Jen Wang who patiently guide me during the whole period. There serious and meticulous teaching will give me a very deep understanding of deep learning and data analysis. At the same time, I really appreciate Andrius Bernatavicius's kindness help. During our weekly meetings, he consistently provides insightful guidance on the future steps we need to undertake. His constructive feedback, delivered in a gentle and respectful manner, helps me identify and rectify my mistakes, fostering my professional growth and development. During my injury, he also helped me handle some school matters (contacting supervisors, graduation registration, etc.). Without his selfless assistance, this project could not have progressed so smoothly.

References

[1] T. Spek, "Pavement distress classification in 3d pavement measurements using convolutional neural networks," Ph.D. dissertation, TNO, 2019.

[2] M. Miradi, A. A. Molenaar, and M. F. van de Ven, "Knowledge discovery and data mining using artificial intelligence to unravel porous asphalt concrete in the netherlands," in *Intelligent and Soft Computing in Infrastructure Systems Engineering: Recent Advances*. Springer, 2009, pp. 107–176.

[3] A. Shtayat, S. Moridpour, B. Best, A. Shroff, and D. Raol, "Dynamic monitoring of asphalt pavement using mobile application," in *26th World Road CongressWorld Road Association (PIARC)*, 2019.

[4] TNO, "Asphalt in the netherlands," 2022. [Online]. Available: https://www.tno.nl/en/sustainable/safe-sustainable-living-environment/infrastructure/asphalt/

[5] M. Huurman, L. Mo, M. Woldekidan, R. Khedoe, and J. Moraal, "Overview of the lot meso mechanical research into porous asphalt raveling," in *Advanced Testing and Characterization of Bituminous Materials, Two Volume Set*. CRC Press, 2009, pp. 523–534.

[6] A. Shtayat, S. Moridpour, B. Best, S. Rumi *et al.*, "An overview of pavement degradation prediction models," *Journal of Advanced Transportation*, 2022.

[7] W. v. Aalst, G. Derksen, P.-P. Schackmann, P. Paffen, F. Bouman, and W. v. Ooijen, "Automated ravelling inspection and maintenance planning on porous asphalt in the netherlands," in *International Symposium Non-Destructive Testing in Civil Engineering (NDTCE 2015). Berlin*, 2015.

[8] W. Van Ooijen, M. Van den Bol, and F. Bouman, "High-speed measurement of ravelling on porous asphalt," in *Symposium on Pavement Surface Characteristics [of Roads and Airports], 5th, 2004, Toronto, Ontario, Canada*, 2004.

[9] TNO, "Discover tno," 2024. [Online]. Available: https://www.tno.nl/en/

[10] J. Nicholls, M. Jacobs, E. Schoen, D. van Vliet, S. Mookhoek, N. Meinen, G. van Bochove, J. De Vissche, A. Vanelstraete, F. Hammoum *et al.*, "Development of a ravelling test for asphalt," in *Bituminous Mixtures and Pavements VII*. CRC Press, 2019, pp. 144–152.

[11] J. Nicholls, "Drat–development of the ravelling test compendium of sites and the extent of ravelling," 2016.

[12] W. van Aalst. Automated raveling inspection and maintenance planning on porous asphalt in the netherlands.

[13] E. Schlangen, "Other materials, applications and future developments," in *Self-healing phenomena in cement-based materials: state-of-the-art report of RILEM technical committee 221-SHC: self-healing phenomena in cement-based materials*. Springer, 2013, pp. 241–256.

[14] D. Brown and E. Johnson, "Traffic load effects on pavement performance: A comparative analysis," *Transportation Research Part B: Methodological*, vol. 74, pp. 112–126, 2019.

[15] Y. R. Kim *et al.*, "Compaction degree and its effects on asphalt mixture performance," *International Journal of Pavement Research*, vol. 34, no. 2, pp. 89–98, 2012.

[16] J. Sullivan *et al.*, "The impact of polymer-modified asphalt on road durability," *Journal of Road Engineering*, vol. 45, no. 3, pp. 123–135, 2015.

[17] Z. Wang, P. Krishnakumari, K. Anupam, J. van Lint, and S. Erkens, "Spatial-temporal analysis of road raveling and its correlation with traffic flow characteristics," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 3609–3616.

[18] Z. Cheng, S. Zheng, N. Liang, X. Li, and L. Li, "Influence of complex service factors on ravelling resistance performance for porous asphalt pavements," *Buildings*, vol. 13, no. 2, 2023. [Online]. Available: https://www.mdpi.com/2075-5309/13/2/323

[19] O. Sirin, D. K. Paul, E. Kassem *et al.*, "State of the art study on aging of asphalt mixtures and use of antioxidant additives," *Advances in Civil Engineering*, vol. 2018, 2018.

[20] A. Abouelsaad and G. White, "Review of asphalt mixture ravelling mechanisms, causes and testing," *International Journal of Pavement Research and Technology*, pp. 1–15, 2021.

[21] J. De Visscher and A. Vanelstraete, "Ravelling by traffic: Performance testing and field validation," *International Journal of Pavement Research and Technology*, vol. 10, no. 1, pp. 54–61, 2017.

[22] Y.-c. Wu and J.-w. Feng, "Development and application of artificial neural network," *Wireless Personal Communications*, vol. 102, pp. 1645–1656, 2018.

[23] N. N. Eldin and A. B. Senouci, "A pavement condition-rating model using backpropagation neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 10, no. 6, pp. 433–441, 1995.

[24] M. Jalal, I. Floris, and L. Quadrifoglio, "Computer-aided prediction of pavement condition index (pci) using ann," 10 2017.

[25] A. Issa, H. Samaneh, and M. Ghanim, "Predicting pavement condition index using artificial neural networks approach," *Ain Shams Engineering Journal*, vol. 13, no. 1, p. 101490, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2090447921002264

[26] N. Karballaeezadeh, F. Zaremotekhases, S. Shamshirband, A. Mosavi, N. Nabipour, P. Csiba, and A. R. Várkonyi-Kóczy, "Intelligent road inspection with advanced machine learning; hybrid prediction models for smart mobility and transportation maintenance systems," *Energies*, vol. 13, no. 7, 2020. [Online]. Available: https://www.mdpi.com/1996-1073/13/7/1718

[27] J.-D. Lin, J.-T. Yau, and L.-H. Hsiao, "Correlation analysis between international roughness index (iri) and pavement distress by neural network," in *82nd Annual Meeting of the Transportation Research Board*, vol. 12, no. 16, 2003, pp. 1–21.

[28] A. Ashrafian, "Classification-based regression models for prediction of the mechanical properties of roller-compacted concrete pavement," *Applied Sciences*, vol. 10, no. 11, p. 3707, 2020.

[29] S. Inkoom, J. Sobanjo, A. Barbu, and X. Niu, "Prediction of the crack condition of highway pavements using machine learning models," *Structure and Infrastructure Engineering*, vol. 15, pp. 1–14, 03 2019.

[30] S. Inkoom and Sobanjo, "Pavement crack rating using machine learning frameworks: Partitioning, bootstrap forest, boosted trees, naïve bayes, and k-nearest neighbors," *Journal of Transportation Engineering, Part B: Pavements*, vol. 145, no. 3, p. 04019031, 2019.

[31] G. Leegwater, W. van Aalst, W. Courage, M. Moenielal, S. Wang, H. Böhms, and G. Luiten, *Doorontwikkeling levensduurvoorspellend asfaltmodel, Asfalt-Impuls - LAM Fase 5 - WP2*. TNO, 2023, no. TNO 2924 R10278.

[32] W. Van Aalst, P. Piscaer, B. Vreugdenhil, F. Bouman, M. van Antwerpen, G. van Antwerpen, and J. Baan, "Raveling algorithm for pa and sma on 3d data," in *Roads and airports pavement surface characteristics*. CRC Press, 2023, pp. 382–392.

[33] I. W. van Aalst. The current dutch highway pavement monitoring system.

[34] TNO, "Dutch HiRav Dataset (TNO-RWS Dutch Highway Raveling Dataset) v20230630," 2023.

[35] W. van Aalst. one meter dos/lcms data tno-rws dutch highway raveling dataset.

[36] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[37] P. Venkatesan and S. Anitha, "Application of a radial basis function neural network for diagnosis of diabetes mellitus." *Current Science (00113891)*, vol. 91, no. 9, 2006.

[38] I. Yilmaz and O. Kaynar, "Multiple regression, ann (rbf, mlp) and anfis models for prediction of swell potential of clayey soils," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5958–5966, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417410012649

[39] B. Boehmke and B. Greenwell, *Hands-on machine learning with R*. Chapman and Hall/CRC, 2019.

[40] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

[41] V. K. Chauhan, K. Dahiya, and A. Sharma, "Problem formulations and solvers in linear svm: a review," *Artificial Intelligence Review*, vol. 52, no. 2, pp. 803–855, 2019.

[42] L. Wu, Y. Peng, J. Fan, and Y. Wang, "Machine learning models for the estimation of monthly mean daily reference evapotranspiration based on cross-station and synthetic data," *Hydrology Research*, vol. 50, no. 6, pp. 1730–1750, 2019.

[43] G. Huang, L. Wu, X. Ma, W. Zhang, J. Fan, X. Yu, W. Zeng, and H. Zhou, "Evaluation of catboost method for prediction of reference evapotranspiration in humid regions," *Journal of Hydrology*, vol. 574, pp. 1029–1041, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022169419304251

[44] N. Bakhareva, A. Shukhman, A. Matveev, P. Polezhaev, Y. Ushakov, and L. Legashev, "Attack detection in enterprise networks by machine learning methods," in *2019 international Russian automation conference (RusAutoCon)*. IEEE, 2019, pp. 1–6.

[45] L. Diao, D. Niu, Z. Zang, and C. Chen, "Short-term weather forecast based on wavelet denoising and catboost," in *2019 Chinese control conference (CCC)*. IEEE, 2019, pp. 3760–3764.

[46] J. Fan, X. Wang, F. Zhang, X. Ma, and L. Wu, "Predicting daily diffuse horizontal solar radiation in various climatic regions of china using support vector machine and tree-based soft computing models with local and extrinsic climatic data," *Journal of Cleaner Production*, vol. 248, p. 119264, 2020.

[47] P. P. Ippolito, "Hyperparameter tuning: The art of fine-tuning machine and deep learning models to improve metric results," in *Applied data science in tourism: Interdisciplinary approaches, methodologies, and applications*. Springer, 2022, pp. 231–251.

[48] M. Wojciuk, Z. Swiderska-Chadaj, K. Siwek, and A. Gertych, "Improving classification accuracy of fine-tuned cnn models: Impact of hyperparameter optimization," *Heliyon*, 2024.

[49] Y.-S. Park and S. Lek, "Artificial neural networks: Multilayer perceptron for ecological modeling," in *Developments in environmental modelling*. Elsevier, 2016, vol. 28, pp. 123–140.

[50] W. H. Lopez Pinaya, S. Vieira, R. Garcia-Dias, and A. Mechelli, "Chapter 10 - convolutional neural networks," in *Machine Learning*, A. Mechelli and

S. Vieira, Eds. Academic Press, 2020, pp. 173–191. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128157398000109

[51] I. Koprinska, D. Wu, and Z. Wang, "Convolutional neural networks for energy time series forecasting," in *2018 international joint conference on neural networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[52] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," *arXiv preprint arXiv:1703.04691*, 2017.

[53] W. Van Aalst, M. van Antwerpen, G. van Antwerpen, and T. Wensveen, "Bewerking dos-lcms data ten behoeve van rafelingstrends zoab," 2024. [Online]. Available: https://filesender.surf.nl/?s=download&token=01e41e83-5f37-48ef-a6d2-a2c5a85985a2

[54] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[55] F. Leung, H. Lam, S. Ling, and P. Tam, "Tuning of the structure and parameters of a neural network using an improved genetic algorithm," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 79–88, 2003.