Degree Thesis
Bachelor’s level
Predicting Road Rut with a Multi-time-series LSTM Model

Author: Henrik Backer-Meurke, Marcus Polland
School: Dalarna University
Supervisors: Roger Nyberg, Elin Ekman
Examiner: Rikard Land
Subject/main field of study: Recurrent Neural Networks
Course code: GIK28T
Credits: 15 hp
Date of examination: 26th of May 2021

At Dalarna University it is possible to publish the student thesis in full text in DiVA. The publishing is Open Access, which means the work will be freely accessible to read and download on the internet. This will significantly increase the dissemination and visibility of the student thesis.

Open Access is becoming the standard route for spreading scientific and academic information on the internet. Dalarna University recommends that both researchers as well as students publish their work Open Access.

I give my/we give our consent for full text publishing (freely accessible on the internet, Open Access):

Yes ☒
No ☐

Dalarna University – SE-791 88 Falun – Phone +4623-77 80 00
Abstract:

Road ruts are depressions or grooves worn into a road. Increases in rut depth are highly undesirable due to the heightened risk of hydroplaning. Accurately predicting increases in road rut depth is important for maintenance planning within the Swedish Transport Administration. At the time of writing this paper, the agency utilizes a linear regression model and is developing a feed-forward neural network for road rut predictions. The aim of the study was to evaluate the possibility of using a Recurrent Neural Network to predict road rut. Through design science research, an artefact in the form of a LSTM model was designed, developed, and evaluated. The dataset consisted of multiple-multivariate short time series where research was limited. Case studies were conducted which inspired the conceptual design of the model. The baseline LSTM model proposed in this paper utilizes the full dataset in combination with time-series individualization through an added index feature. Additional features thought to correlate with rut depth was also studied through multiple training set variations.

The model was evaluated by calculating the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE) for each training set variation. The baseline model predicted rut depth with a MAE of 0.8110 (mm) and a RMSE of 1.124 (mm) outperforming a control set without the added index. The feature with the highest correlation to rut depth was curvature with a MAE of 0.8031 and a RMSE of 1.1093. Initial finding shows that there is a possibility of utilizing an LSTM model trained on multiple-multivariate time series to predict rut depth. Time series individualization through an added index feature yielded better results than control, indicating that it had the desired effect on model performance.

Keywords:

Multiple-multivariate time-series, Multi-time-series LSTM model, Recurrent Neural Networks, Machine Learning, Road rut forecasting

Acknowledgements

During the research process a few employees within the Swedish Transport Administration has been of great help. We attribute thanks to Johan Lang who provided us with invaluable information for the background based on his extensive knowledge in the field. We also attribute special thanks to Fredrik Lindström and Johan Näslund gave us crucial insights and expertise into the PMSv3 system and road rut predictions within the agency. We would also like to thank Hannes Sahlin and Andreas Hägg who facilitated the collaboration with the agency, the creative freedom to choose our research topic and support during the process. Last but not least we would like to thank our supervisors Roger Nyberg and Elin Ekman who supported us from start to finish with guidance and feedback during our academic research.
Concept List

**Artificial Intelligence (AI)**
Artificial intelligence is the concept of a machine expressing human intelligence.

**Artificial Neuron**
A node in a neural network that takes an input and produces an output if active.

**Artificial Neural Network (ANN)**
A Neural network is a machine learning algorithm that learns in a similar way to the neurons in the human brain.

**Backpropagation**
The algorithm that lets the neural network learn by adjusting its weights.

**Feature**
Individual independent variables that act as input to machine learning architectures.

**Hyperparameters**
A set of parameters used to control the learning process during machine learning.

**Long Short-Term Memory (LSTM)**
A RNN architecture that can store memories further back than a traditional RNN.

**Multivariate Time Series**
Time-series with a target variable and one or more features.

**Noise**
Data that contains more meaningless than useful information.

**Recurrent Neural Network (RNN)**
A machine learning architecture that can remember sequence data.

**Road rut**
Road ruts are depressions or grooves worn into a road.

**Signal**
Real patterns in the data.

**The Swedish Transport Administration (STA)**
Swedish agency responsible for long-term planning of public transport infrastructure.

**Time Series**
Sequential data taken at equally spaced points in time.
Table of Contents

1 Introduction ........................................................................................................... 1
   1.1 Background ........................................................................................................ 1
   1.2 Problem Definition ............................................................................................. 3
   1.3 Aim ..................................................................................................................... 3
   1.4 Research Question .............................................................................................. 3
   1.5 Scope and Limitations ......................................................................................... 3
   1.6 Partners .............................................................................................................. 3

2 Literature Review .................................................................................................... 4
   2.1 Road Maintenance .............................................................................................. 4
       2.1.1 Fundamentals of Road Construction ........................................................... 4
       2.1.2 Road Rut ..................................................................................................... 5
       2.1.3 Pavement Profiler ...................................................................................... 7
   2.2 PMSv3 .............................................................................................................. 8
       2.2.1 Description of Measurement Data ............................................................... 8
   2.3 Climate Data ....................................................................................................... 9
       2.3.1 Description of Climate Data ....................................................................... 10
   2.4 Artificial Intelligence ......................................................................................... 10
   2.5 Artificial Neural Networks ................................................................................. 11
       2.5.1 The Artificial Neuron ................................................................................ 13
       2.5.2 Backwards Propagation of Errors .............................................................. 14
       2.5.3 Over- and Underfitting ............................................................................. 16
   2.6 Recurrent Neural Networks .............................................................................. 17
       2.6.1 Vanishing and Exploding Gradient ............................................................ 17
       2.6.2 Long Short-Term Memory ........................................................................ 18
   2.7 Case Studies in Multi-time-series Conceptual Design ....................................... 19
   2.8 Machine Learning Technologies ...................................................................... 20
       2.9 Data Processing ............................................................................................. 21
           2.9.1 Normalization ......................................................................................... 21

3 Methodology ........................................................................................................... 22
   3.1 Research Methodology ...................................................................................... 22
       3.1.1 Resource Collection .................................................................................. 22
       3.1.2 Design and Creation ................................................................................. 22
       3.1.3 Ethical Considerations .............................................................................. 23
   3.2 Data Collection .................................................................................................. 24
       3.2.1 Feature Choice ......................................................................................... 24
1 Introduction

1.1 Background

Roads have been a vital part of society’s development and since the introduction of the automobile, their conditions and how to maintain their functionality have been studied in increasing detail. In 1863 Sweden’s first technical book for building roads was written “Handbok i vägbyggnadskonst” and by the early 1900s even better models for building roads were being used, as described by H. N. Pallin in the book from 1915 “Handledning i byggnad och underhåll”. In parallel with the technical advancements in road construction the issue of maintenance arose. Road maintenance was mentioned by R. Ekwall in his book “Till Vägfrågan” in 1923 where he questioned the old techniques for building and maintaining roads by comparing the quality of Swedish roads to other European countries. By this time, the automobile had increased in popularity and old trails and paths were turned into asphalted highways (Peter Andren, 2019). The increased use of vehicles had a negative impact on the road surface and concepts such as road rut and road deformation became a topic of discussion.

Road ruts are depressions or grooves worn into a road and is highly undesirable for a multiple of reasons. According to Trafikverket (2012) road rut can negatively affect a vehicle’s ability to steer. During rain and snowfall, water collects in the grooves causing risk of hydroplaning. The road ruts also cause difficulties when snow plowing which in turn causes an increase in the use of road salts.

Depending on vehicle flow rate the Swedish road network is surveyed using a measuring vehicle every or every other year (Ramböll Sverige AB, 2014). The use of measuring vehicles with lasers started according to Johan Lang (5 April 2021. Personal interview) by “vägförvaltningen” in Örebro län in 1984 and was used nationally by 1990 (Ramböll Sverige AB, 2014). The measuring vehicle used 11 lasers that measured properties such as megastructure, curvature, and cross-profile. Data collected was used to calculate rut depth, rut area and the distance between rut bottoms (Ramböll Sverige AB, 2014).

Enhancements to the measuring vehicle were introduced in 1996 when they increased the laser count to 17 and is still in use today with minor changes (Lang, J., 5 April 2021, Personal interview).

The Swedish transport administration (STA) has specified maximum road rut depths for paved roads which is dependent on the road’s flow rate and speed. These road requirements regulate when paved roads are resurfaced (Trafikverket, 2012). The task of maintaining and repairing the road network containing 98 500 kilometers of state roads has been given to the STA by the Swedish government (Honauer, U., & Ödeen, S., 2020).

The agency has a 12-year national infrastructure plan for the Swedish transportation system. The latest plan from 2018 has a total estimated cost of road maintenance of 161,800mkr (Trafikverket, 2018). The infrastructure plan contains an operational level plan concerning the next four years of planned road network maintenance. The goal of this plan is to increase the ability to proactively conduct maintenance instead of fixing urgent issues that disturb traffic flow and other activities (Honauer & Ödeen, 2020).
During the construction of the national plan, prediction models are used to prognosticate the future needs of the road network. The models are used as an objective decision support tool in combination with the local knowledge possessed by regional leaders. The local knowledge in combination with the objective prediction models are essential when trying to find the true road conditions (Lang, J., 5 April 2021, Personal interview).

The use of prognosis models was introduced in 1984 when they launched a system called SABU or “styrning av beläggningsunderhåll” that used a matrix printer to print graphs of the road condition. During the late 80’s the agency started migrating their collected data from paper and file cabinets to electronically stored systems and introduced a support system called PUB “planering av underhåll av belagda vägar” that was based on the statistical software called SAS (Lang, J., 5 April 2021, Personal interview).

According to Johan Lang (5 April 2021. Personal interview) a new prognosis system was developed in 1995 when they migrated to the use of personal computers with windows NT as the operating system. The new system still used SAS, however a new user interface was introduced and developed in visual basic. Improvements to the system continued throughout the 2000’s with added variables and UI until 2011, when the development of a new system called PMSv3 started.

PMSv3 is a decision support system that was developed to support decision makers throughout the STA. The system relies on data gathered by the agency to build models for information visualization and prognostication. Measurement data is exportable as either 20-meter road sections or as 100-meter sections created by combining five 20-meter sections. During the time of writing this paper a multiple linear regression model is used to predict road conditions for the coming years. One strength of the prediction model is the ability to extrapolate road wear for roads that have little, or no data based on local conditions (Eriksson & Andrén, 2019).

According to Johan Näslund (24 Mars 2021. Personal interview) a new system called PMSv4 is going to be released in mid-2021. Within PMSv4 a new machine learning prediction model based on the Artificial Neural Network (ANN) feed forward architecture will be introduced alongside the previous model. The new model is superior compared to its predecessor.

The ANN architecture is inspired by the human brain and uses thousands of interconnected nodes that are simple processors. The networks are fed input data in the form of training sets with a target variable and additional independent variables called features. The network learns patterns in the data and the trained model is used for predictions (Hardesty, 2017). Unlike Feed Forward Networks, a Recurrent Neural Network (RNN) considers each training example as dependent on previous input making it particularly useful when processing time sequence data. This property distinguishes an RNN from ANNs due the “memory” component within the network (IBM Cloud Education, 2020).
1.2 Problem Definition
Prediction models used by the Swedish Transport Administration at the time of writing this paper are effective when predicting longer road sections. During our interviews with experts within the agency they explained the importance of shorter sections where road rut depth increases rapidly. These local rapid increases in rut depth forces the agency to pave sections of roads where the mean rut depth is still within the specified maximum values. The models in use by the agency are trained on 100-meter road sections derived from combining five 20-meter sections. Since a Recurrent Neural Network considers the history of the individual road section it should in theory predict each section more precisely if trained on the shorter 20-meter sections. Research on Recurrent Neural Networks is most often based on datasets containing one long continuous time series. At the time of writing this paper, research on multiple-time-series RNN models is extremely limited. Problems in model design arise since each road section is measured once every or every other year resulting in thousands of short time series.

1.3 Aim
To evaluate how a recurrent neural network model can be developed and trained on multiple-multivariate short time series to predict road rut. Through extensive research on the topic a model will be designed, developed, and optimized on a dataset provided by PMSv3. The instantiated model will be used to evaluate which features from the dataset yields the best prediction accuracy.

1.4 Research Question
RQ1: How can a recurrent neural network model be developed and trained on multiple-multivariate short time series to predict road rut?
RQ2: Which features yield the best road rut prediction accuracy?

1.5 Scope and Limitations
The research is restricted to the road E16 between Borlänge and Falun. The case studies are restricted to the Long-Short term memory architecture used for time series data.

1.6 Partners
The Swedish Transport Administration.
2 Literature Review

2.1 Road Maintenance

The Swedish Transport Administration (STA) is responsible for the maintenance and care of the public road network. The annual cost of maintaining the road network is approximately 9.5 billion SEK where road maintenance accounts for a third of the cost (Trafikverket, 2020a). The condition of a paved road can be described with many different parameters where the most important are the evenness of the top layer, road rut, pavement defects, and the load bearing capability of the road (Trafikverket, 2020b).

According to Trafikverket (2019) the main reason for road resurfacing is due to road rut. To understand what causes road rut, it is crucial to understand how roads are built.

2.1.1 Fundamentals of Road Construction

Roads consist of several different layers with their own specific functionality. The layers combined shall ensure that the road construction remains safe to travel on, capable of bearing heavy loads and durable during years of degradation from traffic and climate. The fundamental principle of road construction is to ensure that the layers protect the sublayer from deformation while ensuring that the pressures do not exceed the capacity of the individual layers (Wiman & Tholén, N.D).

Figure 1

The fundamentals of road embankment.

Note: Recreated from Wiman & Tholén (N.D).

Figure 1 illustrates the fundamentals of road embankment which is formed by the natural soil that the road is built upon. The subgrade surface is the boundary between the pavement and subgrade and it is crucial that the surface is even with the right crossfall. Early in the embankment excavation process ditches and drainage pipes are dug and put in place to ensure proper drainage (Trafikverket, N.D).
Figure 2 illustrates the fundamental principles of pavement layering. The wear layer is the top layer and its function is to provide a safe and comfortable surface to drive on. The optional binding layer is an intermediate layer that is used when the material difference between the wear and bounded bearing layer is too large. The bounded and unbounded bearing layer’s primary purpose is to spread the load of the traffic to prevent deformation of the subgrade. The reinforcement layer is also responsible for load spreading, unlike the bearing layer the reinforcement layer is constructed with coarse aggregates which ensures drainage of the pavement. The optional protection layer is used to protect the road from frost heaving (Wiman & Tholén, N.D).

2.1.2 Road Rut

According to Dawson and Kolisoja (2004) permanent road rutting is highly undesirable for reasons such as the increase in fuel consumption and heightened risk of skidding due to water and ice buildup in the ruts. Road ruts also encourage water to collect in the subgrade which in turn reduces the load bearing capabilities of the road, further increasing the rate of deformation.
Rutting on paved roads is fundamentally due to three types of mechanisms;

**Figure 3**
*Mode 0 - Rutting caused by compaction.*

![Diagram of Mode 0 rutting](Image)

*Note. Recreated from Dawson and Kolisoja (2020).*

**Compaction of non-saturated materials - Mode 0 rutting**
During road construction the aggregates used on the roads are compacted using road rollers which is normally considered sufficient to prevent further compaction during trafficking. However, if the aggregates are not fully saturated, rutting may occur by further compaction of aggregates (See *figure 3*). Rutting caused by compaction is however self-stabilizing as further rutting is hindered due to the stiffening of material and better load spreading. In regions affected by frost penetration, the frost in combination with moisture causes heaving in the subgrade and in turn decompaction of the aggregate layers. When the warmer weather in spring starts the thawing process, subgrade compaction becomes possible. Variable heaving along the road causes variable amounts of deformation hence variable road rut depth increases (Dawson & Kolisoja, 2004).

**Figure 4**
*Mode 1 - Rutting caused by subgrade deformation.*

![Diagram of Mode 1 rutting](Image)

*Note. Recreated from Dawson and Kolisoja (2020).*

**Mode 1 rutting - Subgrade deformation**
Subgrade deformation is the deformation of the subgrade in conjunction with the aggregate layers. The pressures on the road surface cause the soil beneath the road to deform and is often characterized by a broad rut with a slight heave remote from the wheel path (See *figure 4*). The soil beneath the aggregates pushes the granular materials upwards causing the heave (Dawson & Kolisoja, 2004).
Mode 3 rutting - Surface abrasion
Rutting caused by abrasion leaves thinner tracks in the surface layer and is primarily caused by the use of studded tires (See figure 5). The studded tires abrade the surface chipping away particles over time (Dawson & Kolisoja, 2004).

2.1.3 Pavement Profiler
Every year the Swedish Transport Administration plans and orders the construction and maintenance of new and existing roads in the Swedish road network. To achieve desired results the agency needs reliable measurement data. The measurement data is used during the lifespan of a road network, from the specification and planning of new roads and as decision support for the yearly maintenance operations (Ramböll Sverige AB, 2014).

The road network is measured to evaluate a road's condition and the perceived comfort of its users. These measurements have historically been carried out with simple measuring techniques such as measuring sticks or with subjective judgements based on experience. Since the mid 1980’s the agency has been measuring road conditions using a laser pavement profiling system (Ramböll Sverige AB, 2014).

The laser pavement profiler is a testing system that measures and collects data at normal traffic speeds to a large extent independently of the vehicle speed variations. The vehicle measures several different surface properties without touching the test object which has benefits as wear and tear on the test equipment is minimal and there is no need for fine calibration (VTI, n.d).

The measurements are used to calculate parameters such as crossfall, hilliness, curvature, and road damage properties such as International Roughness Index (IRI) and road rut (Ramböll Sverige AB, 2019).
2.2 PMSv3

PMSv3 or pavement management system version 3 is a web-application built by the Swedish Transport Administration. The aim of the system is to provide internal and external users access to pavement data from Swedish roads through a user interface. PMSv3 allows the users to export data from measured road conditions such as rut depth, IRI, crossfall, pavement data and general road data such as road speed and width. The desired parameters of a road, or multiple roads can be selected in the interface and exported to excel (Trafikverket, 2019).

Measured road conditions are exportable as sections of 20 respectively 100 meters where the 100-meter sections are derived by combining five 20-meter sections. The pavement profiler measures the roads continuously, however, properties are stored and exportable as either the mean or the max value of the section. Properties such as curviness, hilliness, crossfall and road rut are grouped by the section which reduces the need for manual data processing (Trafikverket, 2019, September 17).

2.2.1 Description of Measurement Data

Starting Point

The starting point for each road section is referred to as StartingPoint (StartLopandeLangd in PMSv3) in this paper. It is the given a value based on how far it is in meters from the starting point of the road (Ramböll Sverige AB, 2014).

Road rut

Figure 6

*Cross Section Created with 17 Lasers from the Road Surface Tester.*

The pavement profiler has a laser system mounted on the front bumper. Each laser measures the distance down to the pavement, the data collected from the lasers are used to create the cross section. The cross section is created by drawing a line through each data point illustrated by the blue line in *figure 6*. To calculate the rut depth of the pavement, a line is drawn by placing a stretched string from the point farthest to the left (A) to the point farthest to the right (C) illustrated by the red line in *figure 6*. “Spårdjup max” is calculated as the maximum distance between the two lines of a section and the mean is the average depth of a section. The method can also be used to calculate the right and left rut depth by using 60% of the cross section on each side (Ramböll Sverige AB, 2014).
Rut bottom distance
The rut bottom distance (Spårbottenavstånd in PMSv3) is the distance in millimeters between two deepest points of each rut on a stretch of road. High rut bottom distance values indicate that the ruts are formed by deformation caused by commercial heavy traffic. Low values indicates that the ruts are formed by wear from studded tire use (Ramböll Sweden AB, 2014).

Crossfall
Crossfall (Tvärfall in PMSv3) on a road is crucial for the safety of its users and the ability for the water to run off the road surface. Crossfall is the gradient of the road surface in relation to the horizon and is recorded as a percentage value. During road measuring, crossfall is calculated by combining the vehicle gradient with the cross profile (Ramböll Sweden AB, 2014).

Hilliness
Hilliness (Backighet in PMSv3) describes the road's longitudinal inclination and is measured using a tilt sensor which registers the pavement profiler’s gradient. The hilliness value is calculated by dividing a stretch of roads length with the difference in height between the highest and lowest value. An uphill is defined with a positive value while a downhill has a negative value (Ramböll Sweden AB, 2014).

Curvature
Curvature (Kurvatur in PMSv3) describes how the road turns and the sharpness of the turns. The radius of the curve, r, is measured using a sensor in the pavement profiler. The road curvature value is calculated by 10 000/r and a mean is derived per 20 meters. Straight roads are given a value of 0 and a 400 meter curve is given a curvature value of 25. The curvature value is positive in right turns and negative in left turns (Ramböll Sweden AB, 2014).

AADT vehicles
AADT vehicles (ÅDT fordon in PMSv3) is the average annual daily traffic of all vehicles (Trafikverket, 2013, June 13).

AADT heavy
AADT heavy (ÅDT tung in PMSv3) is the average annual daily traffic of trucks (Trafikverket, 2013, June 13).

2.3 Climate Data
The Swedish Meteorological and Hydrological Institute (SMHI) is a Swedish expert agency responsible for measuring and displaying weather, water, and climate data. The agency offers tailored services and products that are used nationally and internationally as decision support tools. From general forecasts and warnings to climate studies and research assignments. The agency publishes yearly statistics from their ground-based observation stations where the collected data has been structured and processed (SMHI, 2020).
2.3.1 Description of Climate Data

Accumulated precipitation

Precipitation data is available through the yearly published data sheets from the agency. The data is available as monthly or yearly accumulated precipitation in millimeters for individual weather stations. The number of days with precipitation is also available and is defined as the number of days where accumulated precipitation is above 0.1 millimeters (SMHI, 2002).

Air temperature

Air temperature data is also published by the agency providing yearly temperature statistics from individual weather stations. The datasheets include yearly mean temperature, mean min and max data and number of days with a temperature below the freezing point (SMHI, 2002).

2.4 Artificial Intelligence

According to Andrén (2019) the Swedish government took control of the public road network in 1944. Six years later a researcher named Alan Turing wrote a paper called “Intelligent Machinery”. Instead of an organized machine that was mechanically programmed he proposed an unorganized machine able to learn in a similar way to a human infant (Turing, 1948). Two years later Turing published “Computing machinery and intelligence” in the British journal Mind. This is the publication that contains the famous “Imitation Game” otherwise known as the Turing test which is said to be the starting point of artificial intelligence (Turing, 1950).

Artificial intelligence can be explained by looking at two issues, one the representation of knowledge and the other the ability to search. Knowledge in AI involves the machine's ability to manipulate the entire spectrum of knowledge to express intelligent behavior. Search is a technique to solve problems by exploring the different problem states, like the possible moves on a chess board. To gain knowledge on a specific domain an expert can be used together with an AI specialist to create a system that is effective and intelligent in its behavior. This combination is normally referred to as expert systems and is widely used in the fields of medicine, geology, and chemistry. Another way AI system could gain knowledge is by learning itself, this is commonly referred to as machine learning. Machine learning programs modelled on the evolving patterns found in nature are called genetic algorithms and the algorithms that mimic the neurons in the human brain are called neural networks (Luger, 2005).

In 1957 Frank Rosenblatt invented the perceptron based on the human brain. The proposal was that a system of signal generating neurons would form a network which he called the “perceptron”. This feed forward neural network creates an output if it received a suitable input from either the environment or other neurons. The invention of the perceptron started a boom of new AI research and devices. However, the increased focus on AI also brought with it opposition and newspapers started talking about Frankenstein Monsters being built (Frank Rosenblatt, 1961).

Since the invention of the Perceptron a wave of optimism erupted over the AI field during the 1960s driven by new AI funding and research. In 1969 two authors Minsky and Pepert published a book called Perceptrons. This publication looked at the Perceptron architecture more rigorously and concluded that the computational abilities of the perceptron could not meet the requirements of the best suited applications. The lack of basic theory behind the research on perceptrons started the so-called AI-Winter, a period from mid-70’s to the mid 80’s. Funding
for AI decreased and research on learning machines based on the perceptron came to a standstill (Minsky & Papert, 2017).

Even though funding was withdrawn some groundbreaking research still took place and in 1974 Paul John Werbos published his “Beyond regression” thesis. In his thesis Werbos lays the foundation for back-propagation applied to a neural network. In the thesis he attempted to create a model for predicting a country's political future based on the Deutsch-Solow model of nationalism (Werbos, 1975).

A few years later Dr John Hopfield created the first recurrent neural network known as the Hopfield network. This is the first neural network to have time sequence retention based on the human mind's ability to store memory. Hopfield proposed that given content-addressable memory and an algorithm for time evolution based on asynchronous parallel processing, a system could retrieve its previous states in a time series. Hopfield also demonstrates that his algorithm stores “memories” that can be adjusted. However, he encountered problems with storage capacity when the system tried to store new memories (Hopfield, 1982). The same year as the “Recurrent neural network” was developed by Hopfield, another Neural network architecture “neocognitron” was proposed by the Japanese researcher Kunihiko Fukushima. The multilayered network implemented had the ability to classify and recognize patterns according to their shapes without an expert. Fukushima was able to train his network with input patterns at different angles and distortions. This was done with the use of deep learning, where the deepest layers were not affected by the distortions in the input patterns (Fukushima & Miyake, 1982). What Fukushima presented is today called Convolutional Neural Networks or CNN’s and was famously used in 1989 to recognize handwritten zip codes with an error rate of only 1.1% (LeCun et al., 1990). The CNN’s had problems during this period; datasets were limited in size to be useful and when scaling from prototype to real world applications, they became too computationally heavy (Minsky & Papert, 2017). The RNN architecture was also computationally heavy due to the back propagation training algorithm. Back propagation was coined by Williams, Hinton and Rumelhart (1985) and lets a neural network learn by a process known as gradient descent that adjusts the weights of its layer. However, back propagation for RNN networks suffered from the vanishing gradient problem where the actual descent is so small for each iteration that the computational time to train the RNN was not practical (Hochreiter, 1991). Josef Hochreiter also published a solution to the vanishing error problem with backpropagation for RNN’s together with Jürgen Schmidhuber. In an article published in Neural Computation 1997 they presented long short-term memory (LSTM). A RNN architecture able to bridge and learn long time interval sequences, outperforming and solving complex AI tasks that were not possible before (Hochreiter & Schmidhuber, 1997).

2.5 Artificial Neural Networks

Unlike traditional computer programming, machine learning and neural networks differ in their approach to get desired outcomes. In traditional programming we write and execute business rules and heuristics to get an outcome we anticipate. In machine learning the dataset is used as the fundamentals of rules and heuristics (Baig et al., 2020, p. 3). Machine learning techniques are used to automatically find underlying complex patterns in data. The hidden patterns and the machines knowledge about the problem can be used to predict future events. The patterns within the dataset commonly referred to as the training dataset is found by training a model, also known as fitting a model. During training, the model tries different rules and continually evaluates the changes by evaluating the performance. The variable that the machine learning algorithm is trying to predict is called the target variable while relevant extra data called features are used
to help the machine learning algorithm find patterns. The learning can either be supervised or unsupervised. In supervised learning the model is fed both the input variable and the dependent output variable to predict. In unsupervised learning the model is fed only the input and finds patterns and rules in the input data (Edwards, 2018).

Neural networks are a subset of machine learning and an extension of the research made starting in the early 50s. According to Schmidhuber (2014) they are networks of connected neurons (See Figure 7) with each neuron being a simple processor that produces sequences of outputs according to an activation function. Neurons can behave differently depending on their layer in the network. An input neuron can be activated by input from a sensor connected to the environment while a neuron in a hidden layer can be activated by the weighted connection from another activated neuron. Neurons in the output layer can output back to the environment and in turn trigger new events. The goal of the neural network is in the end to find weights between the connections in the network so that it exhibits the wanted behavior. According to Minsky & Papert (2017) the use cases for neural networks have exploded in 2012 with the invention of GPU based processing together with big image dataset generation. They conclude that they did not imagine in 1969 that the passage of time would eventually tilt the balance in neural networks favor.

Figure 7

A Three-layered Neural Network with Connected Neurons.

Note. $x_1$ = input, $w$ = weights, $b$ = bias, $\sum$ = sum of $(w_1 \cdot x_1)$, $F$ = activation function
2.5.1 The Artificial Neuron

According to (Luger, 2005) Frank Rosenblatt (1958) was the first person to use McCulloch-Pitts mathematically model of a human neuron in his learning algorithm called the Perceptron. It is according to Minsky & Papert (2017) the simplest learning machine based on a single layer neural network and is reported by Almásı et al. (2016), that the artificial neuron used in the perceptron is still the artificial neuron used today in much larger neural networks.

The artificial neuron (See figure 8) has five components which are the inputs $x$, weights $w$, bias $b$, net input function, and activation function (Baig et al., 2020, p. 50). As stated in Luger (2005) the weights represent connection strengths and are a set of real value weights. The input is scaled by the weights for each input line. The sum of the weighted inputs called the “net input function” are passed into the activation function which determines if the neuron is in an active state or not. Depending on this state being either active or inactive the neuron either sends an output or does not.

In the context of a neural network the term learning refers to is the adjustment of weights for each input combined with the biases in the network to get better predictions through error feedback. The weights associated with an input dictate how much influence the input feature should have in computing the next node (Baig et al., 2020).

According to Baig et al., (2020, p. 51) the net input function can be described as the sum of the products of the input and their weights plus the bias $b$ which is also called the additive bias. The net input function is shown below:

$$y = (w_1 \cdot x_1 + w_2 \cdot x_2 + \ldots + w_n \cdot x_n) + b$$

Note. inputs $x$, weights $w$, bias $b$, $\sum =$ net input function, $F =$ activation function, recreated from Luger (2005, p. 456).
2.5.1.1 Activation Functions

The activation function receives input from the net input function and converts the input to a new output range based on the chosen activation function. According to Babs (2020) neurons make decisions depending on the chosen activation function \( f(x) \). Some common activation functions are shown below.

**Rectified Linear Units (ReLU)**

Activation function that ensures the output doesn't go below 0.

\[ f(x) = \max(0, z) \]

**Hyperbolic Tangent (Tanh)**

Activation function returns the hyperbolic tangent of \( z \) between (-1 and 1).

\[ f(x) = \tanh(z) \]

**Sigmoid**

Activation function that gives an output between (0 and 1).

\[ f(x) = \frac{1}{1 + e^{(-1 \cdot z)}} \]

The input data, initial weights \( w \) and biases \( b \) generate an output through the activation function and forwards the output. This is commonly referred to as forward propagation. To actually train the model to become better at predicting a concept called backward propagation needs to be introduced (Baig et al., 2020).

2.5.2 Backwards Propagation of Errors

According to Baig et al., (2020, p. 58) an epoch is each time a neural network processes a training set. During each epoch the network does forward propagation where input data is fed to the neurons and through the network converges to the predicted output. The output is compared to the true label and the error or loss as it is commonly referred to is calculated.

The error or loss is calculated by the loss function which calculates the error between the true and predicted label. Loss functions are categorized per the desired output of the model. There are classification loss functions used when classifying examples and regression loss functions used to predict a real-valued quantity. The latter is relevant in this context (Baig et al., 2020).

2.5.2.1 Loss Functions / Accuracy Metrics

**Root Mean Square Error (RMSE)**

Root Mean Squared Error (RMSE) is a widely used metric for evaluating and training models in machine learning (Munoz-Organero, 2020). Since the errors are squared the RMSE metric penalizes larger prediction errors where an error of 10 is 100 times worse than an error of 1. The penalization of larger errors makes RMSE particularly useful when large prediction errors are considered more important than small deviations (Kampakis, nd).
RMSE can be calculated using the following equation:

\[
RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\text{Predicted}_i - \text{Actual}_i)^2}
\]

**Mean Absolute Error**

Mean Absolute Error or (MAE) is the absolute difference between the true and predicted value. The metric gives a measure of how far the predictions were from the actual value. Since the error is made positive it does not give indications if the model is under or over predicting the data (Mishra, 2018; Parmar, 2018).

MAE can be calculated using the following equation:

\[
\text{MAE} = \frac{1}{n} \sum_{j=0}^{n} |y_{ij} - \hat{y}_j|
\]

**Mean Absolute Percentage Error (MAPE)**

The Mean Absolute Percentage Error (MAPE) is the mean of the absolute percentage errors of predictions. Since it provides the error in terms of percentages it is easily understood.

MAPE can be calculated using the following equation:

\[
\text{MAPE} = \left( \frac{1}{n} \sum_{i=0}^{n} \left| \frac{\text{Actual}_i - \text{Forecast}_i}{\text{Actual}_i} \right| \right) \times 100
\]

**2.5.2.2 Gradient Descent**

For the model to minimize error in the neural network, the model through backpropagation calculates the error in reverse order until it reaches the input layer. As it propagates backwards it adjusts the weights and biases in the network through gradient descent (Baig et al., 2020, p. 59). Gradient descent has been used to optimize the loss function in machine learning algorithms during the past decade. It has advantages of being computationally inexpensive, simple and in the right context can guarantee convergence. However, it is limited in that the user needs to manually adjust the learning rate (Ravaut & Gorti, 2018).

**Learning Rate**

Choosing the right learning rate presents a challenge to researchers. If the learning rate is too small, the network can take a long time to converge and if the learning rate is too high, then the network might not converge at all. This is because during training the learning rate determines the gradient to be applied to the current weights to reduce loss. Getting the right learning rate decreases the amount of time it takes to train the model. Some optimizers have adaptive learning rates meaning manually tweaking the learning rate is not needed (Yu et al., 2020). According to Baig et al., (2020) the algorithm used when updating the weights of the network to minimize loss is called an optimizer.

**2.5.2.3 Optimizers**

New optimizers have been developed that use either first-order gradient descent or second-order gradient descent. First-order optimizers approximate the first order derivatives of the loss function while second order approximates the second order derivatives, the effect on the gradient descent can be seen in figure 9. Stochastic gradient descent (SGD) is an optimizer that iterates over batches of input data to minimize the loss function. The SGD method has been
combined with momentum to get even better optimization results, and lately the learning rate has been dynamically optimized by the ADAM, ADAGRAD and ADADELTA optimizers (Ravaut & Gorti, 2018).

Figure 9
Illustration of first-order vs second order gradient descent.

Stochastic Gradient Descent (SGD)
Stochastic gradient descent is a popular optimization technique. It is optimized to process randomized input batches (red line in figure 9) and has proven to be efficient even for non-convex loss functions, meaning there could be many local minimums (Sutskever et al., 2013).

Adaptive learning rate method (ADADELTA)
This optimizer dynamically adapts as time passes by using first order information. Manual adjustment of the learning rate is not needed (Zeiler, 2012).

Adaptive Gradient Algorithm (ADAGRAD)
The ADAGRAD optimizer uses previously learned information of the data structure to perform better gradient descent learning (Sutskever et al., 2013). According to the developers of ADAGRAD the adaptive learning technique is better than online learning methods for sparse gradient vectors (Duchi, Hazan & Singer, 2011).

Adaptive moment estimation (ADAM)
The ADAM optimizer is a memory efficient optimizer appropriate for very noisy and sparse gradients. According to its authors it only requires first-order gradients to compute individual adaptive learning rates for input parameters. It is based on ADAGRAD and RMSProp methods for stochastic optimization (Kingma & Ba, 2014).

2.5.3 Over- and Underfitting
Neural networks learn the relationships in the input data. These relationships are modeled inside the network. If a dataset is too small many of the relationships created by the network will be noise that only exist in the training data but not in real test data. This is called overfitting (Srivastava et al., 2014). According to Van der Aalst et al (2010) underfitting is the opposite where a network over-generalizes the behavior relationships meaning the model allows
behavior without strong support from the input data. Controlling the balance between overfitting and underfitting is important. A technique to prevent overfitting is called dropout.

### 2.5.3.1 Dropout

Dropout is a technique to handle overfitting by randomly dropping neurons from the network. This temporarily removes the neuron and its connections. The benefits of temporarily dropping neurons are that the network trains on a new network each time so that it does not converge to one goal (Srivastava et al., 2014).

### 2.6 Recurrent Neural Networks

Traditional feed forward neural networks consider inputs and outputs as independent of each other where each example in the training set is processed without consideration of the previous input. Unlike feed forward networks, a Recurrent Neural Network (RNN) considers each training example as dependent on previous input making it particularly useful when processing time sequence data. This property distinguishes an RNN from ANNs due the “memory” component within the network (IBM Cloud Education, 2020).

![A Recurrent neural network moving through different steps in a time-sequence.](image)

*Figure 10*

A Recurrent neural network moving through different steps in a time-sequence.

*Note.* Recreated from Phi (2018).

The process for a simple RNN is better understood if it is unfolded through time $t$. So that every neural layer has a time subscript $x_t, h_t - 1, h_t$ as illustrated in figure 10. The input vector $x_t$ and the previous activation of the hidden layer $h_t - 1$ are adjusted by their weights from their respective weight matrices and added together. The newly formed vector is added a bias and then propagated through an activation function shown as a blue circle in figure 10. The network generates an output and stores the current activation of the hidden layer $h_t$ to be used in the next step as the previous activation of the hidden layer $h_t - 1$. (Guo, 2013)

### 2.6.1 Vanishing and Exploding Gradient

One of the big challenges when training feedforward deep neural networks and recurrent neural networks is the vanishing gradient problem. As the model backpropages through the network updating the weights the layers close to the output node will be updated accordingly. By the time the error propagates to the initial layers its value will have diminished greatly (Baig et al., 2020, p. 267). According to Hochreiter & Schmidhuber (1997) the reason the error signals either vanish or explode is the size of the weights.
Hochreiter & Schmidhuber (1997) and Sherstinsky (2018) found that the “vanishing gradient” and “exploding gradient” problem occur more frequently when the magnitude of the sample set is large, and the training process must unroll a long range of the sequence dependencies.

2.6.2 Long Short-Term Memory

Sepp Hochreiter and Jurgen Schmidhuber developed the LSTM architecture to handle the vanishing gradient problem when backpropagating through time (BPTT). According to the creators, LSTM is able to bridge time steps of over 1000 by using a grading-based-algorithm. The algorithm is capable of enforcing a constant error flow meaning it neither explodes nor vanishes (Hochreiter & Schmidhuber, 1997).

The LSTM architecture replaces the artificial neuron with a memory block composed of gateways that control the flow of information through an activation function, usually sigmoid. The block also has a memory cell unit which is a linear self-connected unit known as the constant error carousel (CEC). The CEC is how LSTM can avoid the vanishing gradient problem of a simple RNN, this is because when BPTT the local error is constant inside the CEC. The activation of the CEC is the cell state (López et al., 2018).

The core concept of an LSTM is the cell state. The cell state acts as the long-term memory of the network where the relevant information is carried through in the network. The cell state is updated within each cell where gates regulate what is kept, forgotten, or added.

![Components of an LSTM Cell.](image)

*Note. Recreated from Phi (2018).*
As seen in figure 11 the cell state travels from the previous cell in the line at the top. What is forgotten or added to the cell state is determined by the gates in the cell (Phi, 2018).

In the original paper called Long Short-Term Memory by Hochreiter & Schmidhuber there was no way to handle input streams that were not previously segmented in a way so that they explicitly flagged ends of series. This caused problems with networks breaking down since a network could not reset leading to the cell state growing indefinitely. To overcome this problem a “forget gate” was added so that the LSTM cells that could learn to reset itself when necessary (Gers, Schmidhuber & Cummins, 1999).

The forget gate is responsible for deciding what should be forgotten or kept and does so by passing the previous hidden state and the current input through a sigmoid function. The forget gate output \( f_t \) is a value from 0 and 1 which is multiplied point by point with the previous cell state \( c_{t-1} \). This pointwise multiplication dictates how much of the previous state should be forgotten or kept. Multiplication with 0.5 for instance means that 50% of the previous state would be “forgotten” (Phi, 2018).

Updating the cell state is done through the input gate where the hidden state and current input is passed through a sigmoid function and a tanh function separately. The points get multiplied together and added to the cell state by pointwise addition. The updated cell state is passed through to the next cell. The cell input \( x_t \) and previous hidden state \( h_{t-1} \) is passed through a sigmoid function in the output gate and is pointwise multiplied with the newly updated cell state after passing a Tanh activation function. The new hidden state \( h_t \) is passed on as output from the cell and is used for predictions (Phi, 2018).

2.7 Case Studies in Multi-time-series Conceptual Design

This section presents two previous research papers that use RNN machine learning models that are trained on multi-time series dataset and has a focus on individualizing the prediction results.

**Case 1: Deep physiological model for blood glucose prediction in T1DM patients.**

Researcher Munoz-Organero (2020) researched machine learning models for predicting Blood Glucose (BG) levels of previous researchers before creating his own hybrid RNN LSTM model for prediction of BG levels. According to the author, deep learning methods outperform the shallow models for BG level predictions when measuring root mean squared deviation (RMSE) accuracy. The best RNN models had a RMSE of 7.55mg/dL and used a RNN with LSTM cells where training was done on both Continuous Glucose Monitoring (CGM) data and insulin data.

The hybrid model created consists of 4 models with an input layer, LSTM cell and a dense layer. The outputs of all layers are concatenated into a final Dense layer that outputs the prediction for Blood Glucose (BG). Two datasets were used, the first set consisted of 40 different patients generated artificially while the second set contained 9 real patients with type 1 diabetes. Validation and training on the data was done by splitting datasets into training and validation having distributions of 70/30% and 80/20%. RMSE results achieved on the simulated patients with the hybrid RNN LSTM model were 3.45(ml/dL) on the 70/30% split, beating all 14 models researched. According to the author one reason for achieving such an accuracy was by using more input signals then previous researchers (Munoz-Organero, 2020).

According to Gu et al (2017) blood glucose predictions are vital for people who take preventative measures in time against health risks. Previous models used have been designed specifically for each person with handcrafted features which has resulted in low accuracy due to limited training data and poor feature representation. The paper proposes a multi-time-series deep LSTM model to accurately predict blood glucose concentrations. The model shares data among multiple users and trains individually on single users. The model consists of three components which is feature representation through feature embedding, multi time series structure learning utilizing the full dataset and individual model training. Feature embedding eliminates noise and results in faster training and better generalization. Structure learning is done through an LSTM network where the different users’ data are merged using a multi-task framework. Considering blood glucose concentrations are individual specific the researchers use an individual training layer to train personalized output. The activation function used in the individual training layer is the ReLU activation function and hyperparameters are optimized using Bayesian Optimization. Preliminary testing shows that the MT-LSTM outperforms both classic regression models and the conventional LSTM. The model is evaluated using RSME where the model got an RMSE of 0.78 (mmol/l) compared to the normal LSTM model at 1.52 (mmol/l) (Gu et al, 2017).

2.8 Machine Learning Technologies

Kaggle (2020) evaluated the most common technologies used in machine learning. The survey gathered responses from 20,036 members. The statistics presented are based on the 13% of responders who are employed as data scientists. The survey showed that 74.1% of responders used JupyterLab as their IDE for machine learning leading over visual studio code at 33.2% of usage.

The Kaggle (2020) survey concludes that python-based tools continue to dominate the machine learning frameworks. Scikit-learn is the most popular framework in use as 82.8% of responders use it.

Scikit-learn is a python machine learning framework that provides simple and efficient tools for predictive data analysis. The framework can be used for preprocessing of data, regression, classification and model evaluation (scikit-learn, n.d.)

Miniconda is a minimalistic python package management tool that lets its user install over 720+ conda packages from the anaconda repository (Conda, n.d.).

JupyterLab is a web-based development environment for Jupyter notebooks, data and code. The environment supports a wide range of workflows in machine learning and data science. Jupyter notebook documents serve as complete computational records of a session where executable code and explanatory documentation are saved (Jupyter, n.d).

Darts is a python library used for easy manipulation and forecasting of time series. The library contains models for predictions from classics such as ARIMA to deep neural networks. Darts support both univariate and multivariate time series and the neural networks can be trained on multiple time series (unit8, n.d).

The Pandas library is a python tool for data analysis and manipulation. The development started in 2008 and has become widely used in an array of domains such as finance, statistics, analytics,
and other data science fields. Some key functionality in the pandas library include: Time-series manipulation, dataset merging and joining, data reading & writing, and its data frame object for manipulating and indexing. (Pandas, n.d)

2.9 Data Processing

2.9.1 Normalization

Normalization is a technique used to process input data to transform the raw inputs to a better form more suited for network training. Without normalization training the neural network would be slow and there is a risk that the network would be biased towards one or more features with a higher range of values. Min-Max normalization rescales the features or outputs from one range to a new range of values. Often this range is between 0 and 1 or from -1 to 1 and an advantage of min-max normalization is the preservation of all exact relationships in the data (Jayalakshmi & Santhakumaran, 2011).
3 Methodology

3.1 Research Methodology

3.1.1 Resource Collection

According to Oates (2006) there is a wide range of sources used in literature reviews where some of sources are considered more worthy in research. Books are often used as introductory sources and explain a field and the main theories and approaches in it. Books are however often outdated and are often not in line with the current thinking. Journal articles often provide the most current thinking and research within an area of interest and a literature review is often composed of mostly journal articles. Reports are often difficult to obtain and are often behind a paywall, they are however often produced to a high professional standard. The reports are not usually reviewed by objective outsiders and therefore should be treated with caution (Oates, 2006).

Unstructured interviews have been conducted with key employees within the Swedish Transportation Administration for background, support, and assessment of the model performance. Unstructured interviews are defined by Oates (2006) as interviews conducted by introducing a concept and letting the interviewed subject develop their ideas speaking freely about events or belief. The voluntary interviews were conducted in online conference calls. Each call was recorded and transcribed with the interviewee’s approval. However, regarding personal integrity and company secrets the transcriptions are not included in the appendix.

Reports and resources connected to the Swedish Transport Administration, road measuring and construction have mainly been provided by the agency contact or searched for on their website.

Sources for possible technical solutions or background information on the processes, methods, architectures, and algorithms used have mainly been found on search engines and databases. These databases and search engines are mainly targeted towards academic research and is considered by Oates (2006) as valuable tools. The search terms that have mainly been used for resource collection are; Road construction, Road rut, Artificial intelligence, Machine Learning, Neural Networks, Recurrent Neural Networks, Long-Short-Term Memory, Multiple-Time Series RNN, Multi time series LSTM, Short time series RNN, Data processing, Normalization.

3.1.2 Design and Creation

According to Oates (2006) the design and creation research strategy focuses on developing new IT artefacts. Artefacts are either constructs, models, methods, or instantiations. Constructs are concept or vocabulary used in a particular IT domain. Models is the combination of constructs and is used to understand or develop solutions for problems. Methods is guidance on the models to be produced and processes to follow to solve problems. Instantiation is a working system that demonstrates that ideas, theories, constructs, or models can be implemented.
Figure 12 shows the overarching research methodology. Awareness is according to Oates (2006) the recognition and articulation of a problem. Suggestion is a tentative suggestion of how the problem might be addressed. Development is where the design idea is implemented and how it is done depends on the type of IT artefact proposed. Evaluation examines and assesses the IT artefacts worth and deviations from expectations. Conclusion involves the documentation of results from the design process and the knowledge gained is identified.

According to Oates (2006) what distinguishes design and creation from normal systems development is the focus on the use of academic qualities such as analysis, explanation, justification, and critical evaluation. The IT artifact developed in this research is considered an instantiation where a prototype is developed based on a case study. The developed artifact can be considered a tangible end product but where the focus is on the development process. For this reason, the conceptual design, implementation, and evaluation of the developed artefact is described in detail. The knowledge gathered from the case studies and the design and development was used to answer the first research question. The instantiated model was used to evaluate different training sets with different features. The results from the evaluations were used to answer the second research question.

3.1.3 Ethical Considerations

This paper is part of the advancements in Artificial intelligence which can have ethical and societal consequences that goes beyond what the authors of this paper intended. There are controversies surrounding the use of AI since the system will do what it is assigned to do without moral and ethical considerations. This could lead to increased inequalities in society and possible other side effects which should be considered during research. These aspects have led to calls for international oversight and regulation along with ethical approaches to the development of Artificial Intelligence (Olbios, 2019). During this research ethical and moral aspects has been considered by the authors. The research efforts in this paper are directed towards positive development of AI since the findings may help facilitate better road conditions.
3.2 Data Collection

According to Oates (2006) a scientific research paper must include a method for data generation. The data collection should result in either a qualitative or quantitative data analysis. In this paper a quantitative data analysis is made based on data collected from the PMSv3 system.

3.2.1 Feature Choice

Parameters were chosen after researching causes for road rut and in consultation with PMSv3 experts from the Swedish Transportation Administration.

Rut depth right H17 (20m) is the mean of the right-side rut for a 20m section and was chosen as the target variable for predictions.

Crossfall and accumulated precipitation were chosen as precipitation is a factor for mode 0 rutting. Considering that crossfall describes a road's ability to get rid of water it is also relevant as a feature (See section 2.2.1).

Since mode 0 rutting can be caused by frost heaving the number of frost days and mean yearly temperature were also chosen (See section 2.1.2).

Curvature and hilliness were chosen as it was thought to have an impact on rut depth. The hypothesis was that curvature and hilliness impacts rut depth as the weight shifts in the vehicles (mode 1) and through increased abrasion from studded tires (mode 3) (See section 2.1.2).

AADT heavy was chosen since heavy traffic has an impact on mode 1 rutting. AADT vehicle was chosen as studded tires abrade the surface in mode 3 rutting which is why the overall flowrate on a road is interesting as a feature. (See section 2.1.2)

Rut bottom distance was chosen after Johan Lang (5 April 2021. Personal interview) described the parameter as interesting in the context of road predictions.
3.2.2 Road Selection

Figure 13
Map with highlighted road section between Falun and Borlänge.

Note. Illustration is from the PMSv3 system where the green highlighted section is the road E16 starting from Borlänge (green marker) and ending in Falun (red marker).

As seen in figure 13 above the following road section on E16 between Falun and Borlänge was chosen as the specified time series 2006-2015 showed no signs of road repair on the section’s analysis page. The parameters chosen and additional obligatory metadata were exported to excel as the raw dataset.

3.2.3 Data Cleaning

Figure 14
Raw data from the PMSv3 system.

<table>
<thead>
<tr>
<th>Matdatum exakt</th>
<th>StartLopandelangd</th>
<th>SlutLopandelangd</th>
<th>Längd</th>
<th>Backighet (20m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006-07-11</td>
<td>168143</td>
<td>168163</td>
<td>20</td>
<td>0,60</td>
</tr>
<tr>
<td>2006-07-11</td>
<td>168153</td>
<td>168183</td>
<td>20</td>
<td>0,60</td>
</tr>
<tr>
<td>2006-07-11</td>
<td>168183</td>
<td>168203</td>
<td>20</td>
<td>0,50</td>
</tr>
<tr>
<td>2006-07-11</td>
<td>168223</td>
<td>168228</td>
<td>15</td>
<td>0,60</td>
</tr>
<tr>
<td>2006-07-11</td>
<td>168238</td>
<td>168243</td>
<td>5</td>
<td>0,60</td>
</tr>
<tr>
<td>2006-07-11</td>
<td>168243</td>
<td>168263</td>
<td>20</td>
<td>0,80</td>
</tr>
<tr>
<td>2006-07-11</td>
<td>168263</td>
<td>168283</td>
<td>20</td>
<td>0,80</td>
</tr>
<tr>
<td>2006-07-11</td>
<td>168283</td>
<td>168302</td>
<td>20</td>
<td>0,40</td>
</tr>
</tbody>
</table>

The raw excel file provided by PMSv3 (See figure 14) has the chosen parameters as its columns and each road section (20m in our case) as its rows. The dataset is ordered by road section ascending for each selected year.
The filter function in excel was used to filter the raw data by StartingPoint (StartLopandelangd) which formatted the raw excel file into time series. Each time-series consisted of one 20-meter road section for a time period of 10 years spanning from 2006 to 2015 (See figure 15). By splitting the data irregularities were exposed connected to the starting point for some of the 20m road sections. As can be seen in figure 16 below there is a mismatch between starting points for some road sections, when looking through the whole dataset this problem seemed to occur more frequently in the year 2011. To manage this problem a decision was made to remove all the time series which did not have 10 rows with the same StartingPoint in an effort not to distort the data quality.

Note. The row containing data from 2011 has a different starting point (168651) than other points inside time-series (168663).
Missing data for curvature between 2006 and 2008 was generated by calculating the mean of the following years, 2009 to 2015. This was done since the curvature value should in theory remain the same over the years on the same road section (Lindström, F., 4 May 2021, Personal interview).

Crossfall data was missing from the raw exported file and was provided by the Swedish Transport Administration separately from their database and appended to the dataset.

Accumulated precipitation, mean yearly temperature and frost days was extracted from the data sheets published by SMHI for the Falun-Lugnet weather station. The extracted data was appended to the dataset.

The filtered and cleaned dataset consisted of 2390 data points or 239 time-series with 10 features and 1 target parameter.

### 3.3 Design and Development

In case study 2 the authors mention the improved performance of an LSTM model trained on multiple patients for blood glucose predictions instead of training a patient specific model. The authors attribute the success to the increased amount of training data in combination with the individual patient specific training performed in the individual training layer. The model has better feature representation considering the larger dataset while keeping the core benefits of an LSTM which is to learn the individual time series yielding better predictions (See section 2.7).

Inspired by case study 2 the proposed design was an LSTM model that was created using the Darts library and fed an array consisting of individual time series. The time series was split into training and validation sets where the first 7 years (2006-2012) in the time series were used to train the model (See section 2.7, case 2). The validation set was exclusively used to compare the actual values to the predictions for 2013 and 2014. Considering the extremely short time sequences for each road section a benefit of training on all training time series is that the model had 1673 data points instead of 7 to train on. The baseline model had the StartingPoint as an added feature which is the index value for each time series. The hypothesis was that passing a time series specific index to the model would allow the cell state to remember time series specific rut depth progression over time. This was hypothesized to yield similar benefits to the individualized training proposed by case 2 in the MT-LSTM model.

#### 3.3.1 Model Development

Python was the programming language used for model construction. This decision was made due to the overwhelming popularity of the python-based libraries. The choice of IDE was a web based Jupyter Notebook in a mini-conda environment (See section 2.8). Drivers for Nvidia CUDA were installed and set up to utilize parallel computing when training the model.

The formatted and cleaned csv file was read using Pandas library (See section 2.8). The reading process resulted in a pandas array where column names were appended separately. The date column was dropped before normalizing the data since the date is categorical. The target Right Rut Depth (20m) was also dropped as it was scaled by dividing each row with a factor of 10. This was done to retain the true values for ease of rescaling the output to the same scale, allowing for better interpretation of the predictions. The Scikit-learn library was then used to normalize the dataset with a Min-Max scaler between 0 and 1 (See section 2.9.1).
The Darts library is specifically built for time series data and has classes and functions to simplify the creation of time series, training of models and evaluating the model performance. For this reason, the Darts library was used as the primary library for model creation, training, and predictions (See section 2.9). The Darts LSTM model takes a timeseries object as input which is why each time series was split and processed into time series objects. The dataset was fed into a loop where a time series object was created every tenth row in the dataset resulting in individual time series objects for each road section. Each time series was split into training and validation series and appended to separate python arrays. The additional features called covariates within darts were appended as time series in separate arrays. The loop resulted in four arrays with 239 timeseries each.

3.3.2 Model Optimization

The model was trained on the baseline dataset where the hyperparameters were tweaked to find the best model. The parameter hidden size is the number of neurons in the LSTM layer. Each neuron should be matched to a pattern detected by the network and activated if there would be a strong enough relationship to the input data (See section 2.5.1). By manually incrementing the value by 1 and finding where the validation loss was lowest, an optimum value was reached. The parameter n_rnn_layers is the number of hidden layers in the network and was incremented by 1 in the same way as hidden size.

Dropout is the number of neurons to be deactivated during backpropagation through gradient descent and is used to avoid the overfitting problem in machine learning (See section 2.5.3). Dropout was optimized by incrementing the value from 0 in 0.1 increments until the value with the lowest validation loss was found (See section 2.5.3.1). Batch size was optimized by incrementing the value by a factor of seven as each series consists of seven time-steps. The number of epochs was set to 700. Since the historically best model was extracted by Darts load_from_checkpoint function the number of epochs did not matter since the model had already converged at one point.

The four most common optimizers used for RNN models were tested. The SDG optimizer uses a random batch of inputs and has proven benefits for non-convex loss functions. The ADAGRAD, ADADELTA and ADAM optimizers are extensions of the SDG optimizer but dynamically adapts their learning rate as time passes (See section 2.5.2.3). The best optimizer was found by comparing validation loss of all four algorithms. ADAM outperformed the other optimizers and was therefore chosen. The random state is the seed value used to replicate the training and validation of the model and was set to 22. Considering the time restraints, the possibility of cross testing each parameter was not a feasible option. The finalized LSTM model can be seen in figure 17 below.
Figure 17

*Final LSTM Model.*

```python
optimizer = optim.Adam
optimizer.lr=0.001

rut_model = RNNModel(
    model="LSTM",
    input_chunk_length=1,
    output_chunk_length=2,
    hidden_size=10,
    n_rnn_layers=1,
    dropout=0.2,
    batch_size=105,
    n_epochs=700,
    optimizer_cls=optimizer,
    model_name='rut_rnn',
    log_tensorboard=True,
    random_state=22)
```

*Note.* `rut_model` = Instantiation of the LSTM model, `optimizer` = the ADAM algorithm for model training.

### 3.3.3 Model Evaluation

Each time series and matching covariates were used as inputs to the Darts model predict function in a loop to evaluate each section's predictions.

Figure 18

*Python code showing the predict function in model.darts class.*

```python
rut_pred = best_model.predict(n=2,series=timeseries[x],covariates=covariates[x])
```

`rut_pred` is an array (See figure 18) consisting of rut depth predictions for years 2013 and 2014. The actual and predicted values for section X were multiplied by 10 to scale the predicted and actual values back to the original scale retaining the true ratio between them. Absolute Error (AE) was calculated for each prediction from the actual and predicted rut depth. The actual, predicted and AE for each rut depth prediction was appended to a data frame. After looping through the whole dataset, the data frame consisted of 478 rows, one row for each predicted timestep.
The Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) was calculated and appended to the data frame by the help of Scikit-learn functions (See figure 19 & section 2.5.2.1).

Figure 19

*Prediction Results Output from the Baseline Model in Darts.*

<table>
<thead>
<tr>
<th></th>
<th>Actual</th>
<th>Predicted</th>
<th>Absolute Error</th>
<th>Root Mean Squared Error</th>
<th>Mean Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8.5</td>
<td>8.261498</td>
<td>0.238502</td>
<td>1.124027</td>
<td>0.810963</td>
</tr>
<tr>
<td>1</td>
<td>9.4</td>
<td>9.557172</td>
<td>0.157172</td>
<td>1.124027</td>
<td>0.810963</td>
</tr>
<tr>
<td>2</td>
<td>8.6</td>
<td>8.496930</td>
<td>0.113062</td>
<td>1.124027</td>
<td>0.810963</td>
</tr>
<tr>
<td>3</td>
<td>9.5</td>
<td>9.789923</td>
<td>0.289923</td>
<td>1.124027</td>
<td>0.810963</td>
</tr>
<tr>
<td>4</td>
<td>7.0</td>
<td>7.198662</td>
<td>0.198662</td>
<td>1.124027</td>
<td>0.810963</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>473</td>
<td>9.3</td>
<td>8.955623</td>
<td>0.644372</td>
<td>1.124027</td>
<td>0.810963</td>
</tr>
<tr>
<td>474</td>
<td>5.8</td>
<td>8.096577</td>
<td>2.296577</td>
<td>1.124027</td>
<td>0.810963</td>
</tr>
<tr>
<td>475</td>
<td>9.0</td>
<td>8.94470</td>
<td>0.05522</td>
<td>1.124027</td>
<td>0.810963</td>
</tr>
<tr>
<td>476</td>
<td>5.3</td>
<td>8.593436</td>
<td>3.293436</td>
<td>1.124027</td>
<td>0.810963</td>
</tr>
<tr>
<td>477</td>
<td>9.3</td>
<td>9.400744</td>
<td>0.100744</td>
<td>1.124027</td>
<td>0.810963</td>
</tr>
</tbody>
</table>

478 rows x 6 columns

*Note. Actual = Actual value, Predicted = Predicted value by our model.*

To evaluate which features yielded the best prediction results, 11 datasets were derived from the original dataset. Each training set variation had the target variable combined with an additional feature and in one set with all features. Each training set variation was trained and evaluated in the same way as the baseline.
4 Results

The conceptual design knowledge gathered from the case studies and the development of the model is presented in chapter 4.1. The prediction results from the instantiated model, including all training set variations is presented in chapter 4.2.

4.1 Conceptual Design Findings

There are a few approaches to developing a multiple multivariate time series LSTM model where case 1 proposes a hybrid model to predict blood glucose (BG) levels. The hybrid model consists of four individual LSTM models, one for each feature in the dataset. The output from the four models is concatenated into a final dense layer for BG predictions. The model was trained on data from 40 artificially generated patients resulting in an RMSE of 3.45( ml/dl) beating the 14 models researched (See section 2.7). The MT-LSTM model proposed by case 2 utilizes three individual model components to successfully train an LSTM model to predict BG levels. Feature representation is done through feature embedding which eliminated noise and allowed for faster training. Generalized multi-time-series structure learning was done through an LSTM network where the individual users’ data was merged to utilize the full dataset. Patient specific training was done through an individual training layer resulting in personalized predictions. Preliminary results show that the proposed model outperforms classic regression models and the conventional LSTM architecture. The proposed model got an RMSE of 0.78( mmol/l) compared to 1.52( mmol/l) for the LSTM model. Both authors attributed the results to the increased amount of training data compared to patient specific models trained on a single individual.

The model proposed in this paper was inspired by the case studies carried out in the literature review. To utilize the full dataset an LSTM model was created using the Darts library and trained on the full training dataset which combines 239 individual time series into a python array. The array of timeseries allowed for an LSTM model to be trained due to the increased amount of training data since a road section specific model would only have seven points to train on. The base model proposed utilized a time series specific index for the cell state to hopefully remember time series specific rut depth increases (See section 2.6.2).
4.2 Prediction Results

Table 1 shows The Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) for each training set displayed in millimeters rounded to four decimals. The first column contains the names of each training set and the second column the added features.

<table>
<thead>
<tr>
<th>Training set</th>
<th>Target</th>
<th>Features</th>
<th>Root Mean Squared Error (RMSE)</th>
<th>Mean Absolute Error (MAE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>Rut depth</td>
<td>StartingPoint</td>
<td>1.124</td>
<td>0.811</td>
</tr>
<tr>
<td>Set 1</td>
<td>Rut depth</td>
<td>Hilliness</td>
<td>1.1706</td>
<td>0.8427</td>
</tr>
<tr>
<td>Set 2</td>
<td>Rut depth</td>
<td>Rut Bottom Width</td>
<td>1.1767</td>
<td>0.8506</td>
</tr>
<tr>
<td>Set 3</td>
<td>Rut depth</td>
<td>Crossfall</td>
<td>1.1241</td>
<td>0.8293</td>
</tr>
<tr>
<td>Set 4</td>
<td>Rut depth</td>
<td>Curvature</td>
<td>1.1093</td>
<td>0.8031</td>
</tr>
<tr>
<td>Set 5</td>
<td>Rut depth</td>
<td>AADT_vehicles</td>
<td>1.1489</td>
<td>0.8254</td>
</tr>
<tr>
<td>Set 6</td>
<td>Rut depth</td>
<td>AADT_heavy</td>
<td>1.1245</td>
<td>0.811</td>
</tr>
<tr>
<td>Set 7</td>
<td>Rut depth</td>
<td>Mean_temperature</td>
<td>1.2388</td>
<td>0.9367</td>
</tr>
<tr>
<td>Set 8</td>
<td>Rut depth</td>
<td>Frost_days</td>
<td>1.2324</td>
<td>0.9227</td>
</tr>
<tr>
<td>Set 9</td>
<td>Rut depth</td>
<td>Precipitation</td>
<td>1.3078</td>
<td>1.0177</td>
</tr>
<tr>
<td>Set 10</td>
<td>Rut depth</td>
<td>All features above</td>
<td>1.1494</td>
<td>0.87</td>
</tr>
<tr>
<td>Control</td>
<td>Rut depth</td>
<td></td>
<td>1.1968</td>
<td>0.876</td>
</tr>
</tbody>
</table>

Nearly half of predictions was under 0.5 millimeters and two thirds was below 1 millimeter (See figure 20). The base model generated a mean absolute error of 0.8110 (mm) outperforming the control at 0.8760 (mm) where the same model was trained without covariates (See Table 1). The results show that the time series specific index may have the desired effect on the model. To get a sense of the performance of the model, the Mean Average Percentage Error (MAPE) for baseline was calculated (See section 2.5.2.1). The baseline MAPE was 10.87% showing the average percentage error for each point.
Curvature yielded the best overall MAE of 0.8031 (mm) indicating that it correlates with rut depth. The results validate to some extent the hypothesis that increased abrasion (mode 3) from studded tires or weight shifting (mode 1) occur at curvy sections (See section 2.1.2).

Results show that hilliness did not correlate as well as curvature with a MAE of 0.8427 (mm) and a RMSE of (1.1706).

Since crossfall has an effect on water runoff (See section 2.2.1) it was considered an important feature for mode (0) rutting. Results indicate that crossfall correlates with rut depth since it yielded a better MAE than control. The traffic flows of a road are seen as important features since rutting occurs due to abrasion from studded tires and pressures on the road surface. AADT_heavy resulted in a MAE of 0.8110 (mm) matching baseline. Rut bottom distance is considered a categorical variable (See section 2.2.1) and had a small impact on rut predictions.

Climate data had a negative impact on the rut predictions compared to control adding noise to the relationships in the model (Note that this is discussed in section 5). Precipitation had a particularly strong negative correlation to rut depth with a MAE of 1.0177 (mm) (See section 2.2.1).

The dataset with all features added, showed mild improvements in MAE with 0.87 (mm) over control 0.876 (mm). The added features also yielded a better RMSE than six other sets indicating that it was better at predicting outliers (See section 2.5.2.1).
5 Discussion

5.1 Base Model Results

Recurrent neural network models can be developed and trained on multiple-multivariate short time series in different ways. The model proposed in this paper utilizes the full training set with a time series specific index that yielded better prediction results compared to the univariate model with target only. According to Fredrik Lindström (4 May 2021, Personal Interview) the model shows promising results and potential use cases for the RNN architecture within the agency. He thought that evaluating shorter road sections was extremely interesting since this is something that the agency wants to do. Research is limited on multiple-multivariate-time series RNN models and comparable use cases were hard to find. The main field where LSTM models trained on multiple-multivariate time series have been developed and evaluated in the medical research field. Case 1 proposes a hybrid model that showed significantly better results than the models it was compared to. The MT-LSTM model proposed by case 2 with generalized feature training and an individual training layer outperformed the conventional LSTM model it was compared to. Our results when using an index feature were similar to the two case studies in that all showed improvements by utilizing some sort of individualization. All three models used the LSTM architecture but had different designs. The model proposed in this paper used the index as a feature so that the network could learn the relationship for the individual road sections. The other two researchers addressed the problem with either an individual training layer or model component (See section 2.7).

The model proposed achieved better than expected results with a MAPE of 10.87% and a MAE of 0.81(mm). According to Johan Näslund (24 mars 2021. Personal interview) a mean error below 1 millimeter are considered good in the context of rut predictions. Even though the results were promising, there were limitations in data quantity and quality in the training dataset. The decision to remove distorted time series resulted in the loss of nearly half of the dataset. According to Fredrik Lindström (4 May 2021, Personal Interview) The Swedish Transport Administration is using 100-meter sections derived from the 20-meter sections where the mismatch in StartingPoint has been adjusted and matched correctly. Outliers considered unreasonable in terms of progression are also filtered out to increase data quality. The decision to remove corrupt time series was made due to the lack of a documented method for adjusting the time series without distorting the data quality.
The training set used had some issues with data quality where the rut depth decreased for some time series. These decreases are impossible for the model to anticipate (See figure 21), especially since most of the decreases have been observed in the validation set. According to Fredrik Lindström (4 May 2021, Personal interview) the time series where this phenomenon occurred are most likely due to irregularities when measuring the road. An example he mentioned is that a driver can for instance evade obstacles on the road. The time series with these kinds of issues affected the overall prediction results as may be indicated in the difference between the RMSE and the MAE in the baseline model. This is because the RMSE penalizes big prediction errors. According to Fredrik Lindström (4 May 2021, Personal interview) the release of PMSv4 will improve the quality of the data where misalignments of StartingPoint will be addressed. This will increase the possibility of utilizing a Recurrent Neural Network for predictions (See section 2.6.2).

Figure 22 shows the plotted graphs of two of the best predictions. The difference in progression between the graphs may show the ability of the model to predict time series individually. Since most of the rapid increases in our dataset was contained in the validation set it is still hard for the model to anticipate these rapid increases. To further validate the ability to predict rapid increases in rut depth it would be interesting to train the model on longer time series with rapid increases in rut depth. This would give the model a fair chance of predicting future rapid increases. Since the model can be considered a black box interpretation of the model’s ability to individualize predictions based on the added index are hard to validate.
5.2 Feature Analysis

The prediction result from all training variations show that curvature had the strongest impact when predicting rut depth with a MAE of 0.8031 (mm). The fact that curvature yielded better predictions than the baseline can be attributed to the fact that curvature is also time series specific with added effect since the curvature has real impact on rut depth. The initial findings show that the vehicle flow rate is of great interest when predicting road rut. AADT_vehicles resulted in a MAE of 0.8254 (mm) which is better than control. AADT_heavy resulted in an impressive MAE of 0.8110 matching baseline. Crossfall yielded similar predictions as AADT_vehicles. Hilliness and the categorical variable rut bottom width seems to have low impact on rut depth predictions considering the outcome. The climate related variables added noise to the predictions and resulted in worse predictions than the control. This could be due to the fact that each time series has the same data for each year resulting in generalization instead of individualization of the time series predictions. The dataset with all features added, showed mild improvements in MAE over control. This validates the effectiveness of an LSTM network's ability to filter out noise since it still performed better than control and three other combinations (See section 2.6.2). The added features also yielded a better RMSE than six other sets indicating that it was better at predicting outliers.

5.3 Method Critique

The results from the comparisons are interesting for the agency and could be considered indications of which features correlate with rut depth. Since the control model had the same hyperparameters as the baseline with covariates there is a possibility that the univariate control was not fully optimized. Due to time restraints the optimization of the model was not as structured as it could have been. Creating separate models for control would have added increased uncertainty of the results due to the possibility of deviations in the way the two models would have been optimized. There are methods used to find the best hyperparameters which could have been used to generate a univariate and multivariate model. Future researchers could validate the results of the feature comparison by utilizing a more formal optimization strategy.

The literature review resulted in usable knowledge in the form of model design inspiration and background to the topic. The two cases in the case study were recently published which is important in the field of AI since development is rapid. Older sources are often outdated in terms of the techniques and frameworks used today. The majority of references in the literature review where journal articles that are considered trustworthy and relevant in this context. The importance of journal articles in scientific research is mentioned by Oates in her book “Researching Information Systems and Computing” (See chapter 2.2).

The design and creation approach as described by Oates has its limitations. There is a lot of responsibility on the individual researchers to apply the methods described on their individual research. Design and creation is described as a perspective on applied research where focus is applied to documentation of every stage of development. The balance of documenting and over documenting the model design, creation and instantiation is hard to find. The Design and Creation methodology was a suitable method for answering the research questions. The ability to validate design choices based on case studies through the instantiation of a model further strengthened the validity of design choices. The quantitative approach to research gave stronger and more interpretable evidence for this research topic compared to evaluation through a qualitative approach. However, a quantitative approach requires a controlled and strict environment to facilitate the repeatability of the model results. For this reason, an emphasis was made on the documentation and explanation of the model design, development, and evaluation.
5.4 Future Research

The possibility exists that features that had moderate effect on rut predictions could in combination with other features yield exponentially better prediction results. This is due to the fact that they may correlate with each other and rut depth at the same time. These possible combinations are of great interest and could be validated by future research. Since an experiment with all combinations would have resulted in too many training sets, it was not evaluated in this paper.

Further validation on the individual index effect on the model’s ability to adjust the predictions are of great interest for future research. Fredrik Lindström (28 April 2011, Personal interview) also mentioned the fact that interpreting what causes rapid rut depth progression is hard when utilizing artificial intelligence. Finding out how the model interprets input, and which weighted relations that are responsible for prediction results are of great interest to better point to certain reasons for rut depth increases.
6 Conclusion

The model shows the effectiveness of the Recurrent Neural Networks in use cases where the target variable is time dependent. During the evaluation process, additional use cases for the agency was discovered. Being able to visualize the prediction results for each 20-meter section gives the agency a new method for performing analysis on individual rut depth progression. The ability to test which features correlates with rut depth is a possibility that is also highly desirable for the agency. The research discoveries in this paper can be considered as optimistic indications of the usefulness of a Recurrent Neural Networks in the field of road maintenance. Both the agency and the authors conclude that the research can be considered a success.

RQ1: How can a recurrent neural network model be developed and trained on multiple-multivariate short time series to predict road rut?

There are a few different ways of developing a recurrent neural network that can be trained on multiple-multivariate short time series. All models share the common attribute of time series individualization, either through an individual training layer or through an added index feature.

RQ2: Which features yield the best road rut prediction accuracy?

Curvature yielded the best prediction results tightly followed by AADT Heavy and StartingPoint which is the index feature for each road section. Crossfall yielded better predictions than control while the climate related variables had negative correlation with rut depth.
References

https://doi.org/10.1016/j.neucom.2015.02.092


https://medium.com/coinmons/the-mathematics-of-neural-network-60a112dd3e05

https://ebookcentral.proquest.com

https://docs.conda.io/en/latest/miniconda.html

C3. (n.d). What is Mean Absolute Percent Error (MAPE). Retrieved from: 
https://c3.ai/glossary/data-science/mean-absolute-percent-error/


https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0

http://urn.kb.se/resolve?urn=urn:nbn:se:vti:diva-13973

https://link.springer.com/chapter/10.1007/978-3-642-46466-9_18

https://digital-library.theiet.org/content/conferences/10.1049/cp_19991218


**Interviews**

Lang, Johan [Trafikverket Borlänge, Sweden] 5 April 2021. *Personal Interview*
