Comparison of Machine Learning Methods for Evaluating Pavement Roughness Based on Vehicle Response

Philippe Nitsche[1]; Rainer Stütz[2]; Michael Kammer[3]; and Peter Maurer, Ph.D.[4]

Abstract: The roughness of a road pavement affects safety, ride comfort, and road durability. A useful indicator for evaluating roughness is the weighted longitudinal profile (wLP). In this paper, three machine learning models are compared for estimating the wLP when vehicle response information, i.e., accelerometer and wheel speed data is collected from common in-vehicle sensors. A multilayer perceptron, support vector machine (SVM) and random forest were applied for testing their effectiveness in estimating the key indices of wLP, namely, range and standard deviation. These models were trained from a set of features extracted from vehicle response simulations on accurate replications of roads with various roughness problems. In contrast to other research, the authors validated models with measurements collected with a probe vehicle. The results show that roughness phenomena can be accurately detected. The SVM produced the best results, although the models achieved rather similar performance. However, differences were found regarding the model robustness when reducing the size of the training feature set. The proposed method enables road network monitoring to be achieved by conventional passenger cars, which can be seen as a practical supplement to the prevalent road measurements with cost-intensive mobile devices. **DOI: 10.1061/(ASCE)CP.1943-5487.0000285.**

Author keywords: Roughness; Road pavement; Support vector machine; Random forest; Neural network; Weighted longitudinal profile; Machine learning; In-vehicle sensors; Vehicle dynamics; Simulation.

Introduction

Road roughness is a pavement surface parameter that affects ride comfort and road safety. Moreover, uneven surfaces can influence the durability of roads, since increased dynamic axle loads of vehicles can occur. Roughness is commonly measured with profilers that are able to capture variations in the pavement profile relative to a reference level. In this paper, the focus is on longitudinal road profiles, because they mainly influence vehicle response and ride quality. Modern profile instruments consist of lasers to measure the profile on a longitudinal line of the road. A suitable algorithm must then be applied to summarize the measured values to a so-called roughness index such as the international roughness index (IRI) (cf. Awasthi et al. 2003). A detailed summary of common roughness indices is given by Sayers and Karamihas (1997).

Most profilers require expensive installations and maintenance. However, continuous measurement of road networks is not practical for most administrations. Therefore, the utilization of in-vehicle sensors available in common passenger cars was studied. For example, the authors derived roughness information from a response profile measured by accelerometers, as proposed by Gonzales et al. (2007), Hugo et al. (2008), Harris et al. (2010), and Yamabe et al. (2010). In most research, the vehicle response is calculated from accelerations of the vertical axes. This can either be achieved by using real-car measurements (cf. Yamabe et al. 2010) or simplified vehicle models for simulation environments (cf. Gáspár and Nádai 2007; Harris et al. 2010).

The measured or simulated vibration response, typically given by accelerations, needs to be further processed to derive roughness information. Finding an appropriate method has been subject to research in the past years. While some use machine learning models, others focus on physical parametric and mathematical computation models. Ngwangwa et al. (2010) used a Bayesian-regularized nonlinear autoregressive exogenous model (NARX) neural network for reconstructing road profiles and achieved satisfactory results. A recurrent neural network was used by Guarneri et al. (2008) that was trained with outputs from simulations of a tire-suspension model in order to predict dynamic behavior. A more detailed study on the effect of learning rate and momentum-term in the backpropagation algorithm was conducted by Attoh-Okine (1999), who aimed at predicting the IRI as pavement performance indicator.

This paper is built upon the findings of studies that use artificial neural networks (ANN) for estimating road roughness. In contrast to other works, the authors trained three machine learning models with simulation data and measured roughness values as training instances, but validated with real in-vehicle measurements carried out on test tracks. A multilayer perceptron served as reference model. Furthermore, random forests and support vector machines were considered. To the authors' knowledge, the latter two model types have not been applied to this particular estimation problem. The goal was to identify strengths and weaknesses of different models regarding this specific purpose. Standard linear regression models have been disregarded, because of problems with correlated predictors and their inability to capture nonlinear relationships efficiently. The weighted longitudinal profile, which was taken as indicator to be estimated, is described in the following section. The authors proceed by illustrating the methodology, explaining the steps from acquiring training and validation data to feature

[1]M.Sc., AIT Austrian Institute of Technology, Mobility Dept., Giefinggasse 2, 1210 Vienna, Austria (corresponding author). E-mail: philippe.nitsche@ait.ac.at

[2]Diplom-Ingenieur, AIT Austrian Institute of Technology, Mobility Dept., 1210 Vienna, Austria. E-mail: rainer.stuetz@ait.ac.at

[3]Diplom-Ingenieur, AIT Austrian Institute of Technology, Mobility Dept., 1210 Vienna, Austria. E-mail: michael.kammer.fl@ait.ac.at

[4]Associate Professor, AIT Austrian Institute of Technology, Mobility Dept., 1210 Vienna, Austria. E-mail: peter.maurer@ait.ac.at

Note. This manuscript was submitted on June 26, 2012; approved on December 26, 2012; published online on December 29, 2012. Discussion period open until August 4, 2014; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801/04014015(8)/$25.00.

extraction and model design. The estimation results are presented in the last section, before concluding this paper with final remarks and future research recommendations.

Weighted Longitudinal Profile

In this work, road pavement roughness is characterized by the weighted longitudinal profile (wLP), which was first introduced by Ueckermann (2005) and further evaluated by Ueckermann and Steinauer (2008). This evaluation method is capable of distinguishing between three different phenomena of roughness (see Fig. 1) by amplifying single irregularities and pseudo-periodical road profiles. Moreover, the wLP is independent of vibration behavior and speed.

The true road profile $z(x_i)$ that is measured with a profiler is segmented into sections with a length of 50 m with a sampling distance of $\Delta x = 0.1$ m. In order to obtain a sample size of 2^n for a Fourier transformation, the sections are enhanced to 2,048 profile points x with overlapping segments before and after. As a first step, the segmented profile must be transformed into the frequency domain. As proposed by Ueckermann and Steinauer (2008), the transformed profile $\underline{z}(k)$ is then filtered by a fourth order Butterworth high-pass filter H with a cutoff wavelength L_c of 50 m and maximum wavelength $L_s = 204.8$ m, which can be written as

$$\underline{Z}(k) = \underline{z}(k) \cdot H(k)$$

The weighted spectrum is calculated as

$$\underline{Z}_w(k) = \underline{Z}(k) \cdot W(k)$$

where the weighting function is given by

$$W(k) = \begin{cases} \tilde{W}(k) & \text{for } 0 \le k < 1025 \\ \tilde{W}(2048 - k) & \text{for } 1025 \le k < 2048 \end{cases}$$

and

$$\tilde{W}(j) = \sqrt{\frac{(j\frac{L_c}{L_s})^{\alpha-1}}{2^{\frac{\alpha-1}{2}} - 2^{-\frac{\alpha-1}{2}}}}$$

The purpose of the weighting function is to amplify single as well as pseudo-periodical irregularities in the profile. The road's waviness, which is a measure of the amplitude ratio between short and long wavelengths of a profile, is denoted as α. Typical values of α are between 2.0 and 2.8, depending on road network characteristics. By modifying α and therefore the weighting function, one can give special emphasis on short or long wavelengths. Basically, the higher the proportion of shortwave irregularities on a pavement, the higher the waviness should be chosen. Since the test tracks have a high proportion of concrete pavements and therefore mostly shortwave irregularities, a value of 2.6 is used.

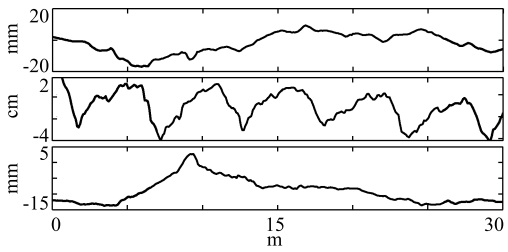

Fig. 1. True road profiles showing general roughness, pseudo-periodical roughness, and single irregularities

The weighted spectrum $\underline{Z}_w$ is then separated into 10 octaves, where each of these octave bands is separately retransformed into the space domain. The overlaps of each single profile are cut off to obtain the original section length of 50 m. Consequently, there are 10 weighted profiles $z_{w,j}$ with $1 \le j \le 10$, whose weighted sum results in

$$\text{wLP} = \sum_{j=1}^{N=10} \frac{\sigma_{w,j}}{\sigma_{w,\text{all}}} \cdot z_{w,j}$$

where σ_w is the standard deviation. From the weighted profile, two characteristic values are calculated: the range Δ_{wLP} and standard deviation σ_{wLP} of the wLP. These can be compared to the respective threshold values given by the national road maintenance guidelines. We considered both values as the target variables for the machine learning methods described in this paper. The estimation of road roughness parameters presents the special difficulty that the data contains outliers, which are not to be discarded but rather might indicate dangerous sections along a road, where, e.g., the pavement is damaged. Such single irregularities are indicated by a higher Δ_{wLP}, while periodical roughness produces a higher σ_{wLP}.

Methodology

The proposed methodology for estimating the roughness of a road (given by Δ_{wLP} and σ_{wLP}) is based on the analysis of dynamic response of the vehicle traveling on it (see Fig. 2). Vibration response and varying wheel speeds act as indicators for determining road roughness. Therefore, we created a simulation framework to calculate the response of a full-car vehicle model traversing different road models with specific roughness characteristics. The vehicle-infrastructure interaction simulation (VIIS) comprises a realistic model of a passenger car. The Dymola was used, which is a commercial implementation of the object-oriented Modelica modeling language that emphasizes on multidomain modeling and simulation. The authors further employed the vehicle dynamics library (VDL) (Andreasson 2011) for multibody vehicle dynamics simulations. For this research, 25 test tracks of the Austrian high-level road network were accurately replicated by creating three-dimensional road models, which are based on laser measurements from the mobile laboratory RoadSTAR. The RoadSTAR system is comprehensively described by Maurer (2007). The VIIS computed

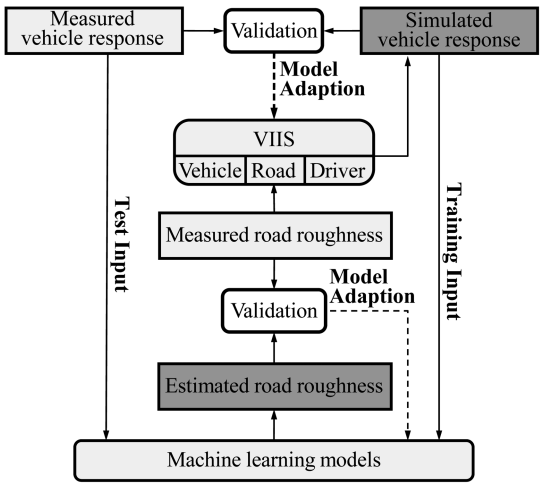

Fig. 2. Methodology for estimating road roughness

the vehicle response that was further validated with measurements taken with a probe vehicle. Then, the simulation models were calibrated to achieve realistic outputs.

Three different machine learning models were applied for estimating road roughness. They were trained with a set of features taken from the outputs of the vehicle response simulation. The roughness estimation was validated by comparing the results with the true road roughness measured by the RoadSTAR system.

Acquisition of Training Data from Simulations

The key parts of the VIIS environment are road, vehicle, and driver models, which are briefly described in the following sections.

3D Road Model

In Austria, the high-level road network is periodically monitored using the high-performance measuring vehicle RoadSTAR. The most important surface properties and road geometry parameters are measured under normal traffic conditions at measuring speeds between 40 and 120 km/h (standard speed 60 km/h). A modified Stuttgart skiddometer is used for the measurement of the skid resistance. Texture and evenness are measured with laser scanning systems, while the road geometry parameters are derived from the inertial measurement unit (IMU). In combination with a differential global positioning system (GPS) unit these parameters can be referenced to their position.

The presented road model is an adapted version of the model described in Nitsche et al. (2011). For an implementation of a road model in the VDL, the interface specifications for ground models was considered. To construct a 3D road model from RoadSTAR measurement data in the VDL, the parameters shown in Table 1 provide essential inputs.

Using the origin $p_0 = (0, 0, 0)^T$ of the world coordinate system, the road centerline is calculated according to the equation

$$p_i = p_{i-1} + \begin{pmatrix} \psi_i \cos(\phi_i \pi/180) \\ \psi_i \sin(\phi_i \pi/180) \\ \sin(\psi_i) \end{pmatrix}$$

where $i \in \{1, \ldots, n-1\}$ and n is the number of measurements. The quantity $\psi_i = \arctan(s_i/100)$ is the angle of gradient. At every point of the road centerline, lateral profiles are added to the road model. These profiles have crossfall q_i and are perpendicular to the tangent vector of the centerline. To model rut depth, the transverse profile is measured by 23 lasers covering a lateral width of 3.3 m. The corresponding measurement values $H_{1,i}, \ldots, H_{23,i}$ are superimposed on the lateral profiles. The measurements of the longitudinal profile are inserted between corresponding points of the lateral profiles. Finally, these points are used to generate triangular or quadrilateral surface patches with a lateral resolution of 15 cm and a longitudinal resolution of 10 cm. For the visualization of the road model in the simulation environment, these patches are represented as 3DFACES in a resulting DXF-file (drawing exchange format). For the simulations in Dymola, another file is required, which contains the coordinates of the surface patches along with the corresponding friction values μ. By specifying arbitrary functions for the parameters in Table 1, we are also able to generate artificial road segments with adaptive resolution (e.g., road bumps).

Table 1. Road Geometry and Condition Parameters

Parameter	Resolution (m)
Pavement skid resistance μ	5
Pavement slope s (%)	1
Pavement crossfall q (%)	1
Heading angle ϕ (°)	1
Transversal pavement profile $H_1, \ldots, H_{23}$ (10^{-4} m)	1
Longitudinal pavement profile $P_1, \ldots, P_{10}$ (mm)	0.1

Vehicle Model

The vehicle model used in the simulations was a generic front wheel driven car. This model was adapted to match the parameters of a BMW X3 model, because the authors used this car as test vehicle for evaluating the estimation outputs. For this purpose, the authors referred to data from the BMW technical information system (TIS) to enrich the vehicle model; the main adjusted properties are listed in Table 2. Tire models are one of the key components for vehicle dynamics simulations. In this study, the authors used the semiphysical tire model TMeasy (Hirschberg et al. 2007, which relies on measured and observed force-slip characteristics.

Driver Model

The instruction driver model included in the VDL was adopted as the default driver model for all simulations. It takes a list of instructions, each consisting of a condition and an action. The driver reads the list sequentially from the first instruction, so that one instruction is always active. When an instruction is active, the corresponding action is performed when the condition becomes true. An instruction timer keeps track of the time from when the currently active instruction was activated. The instruction driver starts in lateral and longitudinal (including gear changing) closed-loop tracking mode with the reference trajectory from the ground. A closed-loop action is interrupted by any open-loop action on any of the involved driver controls. For example, if speed hold is active and a brake ramp or gear select is applied, then the speed hold is interrupted, and the accelerator pedal is set to default state. Lateral closed-loop tracking can not be restarted after an open loop steering action.

Acquisition of Validation Data from Measurements

Measurements to collect data for roughness estimation were carried out. This data includes road profiles measured by the RoadSTAR system as well as vehicle dynamics measurements with a probe vehicle.

Probe Vehicle

A BMW X3 2.0i equipped with a CAN (controller area network) reader software was used as a probe vehicle. An additional accelerometer was mounted in the center console inside the vehicle. During the measurements, the following sensor data were recorded:

Longitudinal acceleration: An important variable is the acceleration measured in the longitudinal direction of the vehicle.

Lateral acceleration: The acceleration in the lateral direction of the vehicle is essential for vehicle dynamic control systems such as the ESP (electronic stability program). For the estimation of longitudinal road profiles, the lateral acceleration is of minor interest.

Table 2. BMW X3 Parameters Used in the Vehicle Model

Parameter	Value
Track width (front)	1,538 mm
Track width (back)	1,556 mm
Wheel base	2,795 mm
Height	1,495 mm
c_w-value	0.35
Weight	1,665 kg
Tire dimensions	235/55/R17

Vertical acceleration: The acceleration in the vertical direction is not provided by the CAN bus and therefore captured by an additional accelerometer. It is useful for roughness estimation, since it gives the vertical deviations of the vehicle.

Steering angle: The steering angle sensor is mounted on the steering shaft in the interior of the vehicle and provides real-time measurements. By recording data of two potentiometers offset by 90 degrees, it can be determined how far and how fast the steering wheel has been turned. Turning the wheel clockwise produces a negative angle, counterclockwise a positive angle value in degrees.

Wheel speeds: The speed of the rotations of all four wheels is measured and sent to the control module, which is connected to the CAN bus. Wheel speed sensors for automobiles typically use an indexing disc mounted on a wheel as well as a pickup that detects the passage of marker elements carried by the disc as the wheel turns. Safety systems like ABS (anti-lock braking system) or ESP strongly depend on the data provided by the wheel speed sensors.

Brake/clutch state: The binary state of the clutch and the brake is provided by the CAN bus. The states *active* and *inactive* are available.

Additionally an external GPS receiver was mounted behind the front windshield, which sent an NMEA (National Marine Electronics Association) message every second. It recorded values for time, position, speed, satellite number, dilution of precision (DOP) values, and heading. In order to produce merged data sets, the CAN reader software synchronized CAN and NMEA messages.

Test Tracks

In order to choose representative test tracks, it was necessary to cover a wide range of different road profiles. Road sections with bad roughness were considered, as well as very smooth tracks. Every test track was segmented into road stretches of a certain length of maximal 2,000 m to optimize calculation time of postprocessing tasks. In total, 25 stretches comprising 30 km of Austrian motorways were thus available. These stretches were simulated in the simulation environment. The longitudinal profiles of the test tracks were also measured using the RoadSTAR system in order to provide some reference information for the CAN data.

Measurement Procedure

Measurements were only carried out while driving with steady speed. When traversing a rough road, the driver may unintentionally push the accelerator pedal. This could cause undesired variations in the engine speed, wheel speed, and longitudinal accelerometer signals. By activating the cruise control system, these variations could be minimized. During the measurements without cruise control, the driver tried to keep the speed at a constant level. Furthermore, unnecessary steering movements were avoided. During the measurements, it was necessary to record markers for specific events. These markers were used in the postprocessing phase in order to crop the recorded measurement files. Precise GPS information was necessary to match CAN measurements with RoadSTAR measurements in postprocessing. Therefore, the authors carried out each measurement run in the same lane and ideally in the same trajectory that the RoadSTAR had driven.

Feature Extraction and Selection

In order to use the proposed machine learning models effectively, one of the most important steps is to extract expressive features from the collected data. The sources for these features should be closely related to the estimated variable as well as be available from standard in-vehicle sensors. For example, access to this information can be gained via the bus systems of modern cars, in cooperation with the manufacturer. For these reasons, we chose the longitudinal, lateral and vertical acceleration (a_x, a_y, a_z) as well as the angular speed $(\omega_{fl}, \omega_{fr}, \omega_{rl}, \omega_{rr})$ of the four wheels (front/rear, left/right) for further processing. All of these signals are mostly determined by the road geometry in combination with the driver's actions and are therefore suited to reconstruct these underlying factors.

Each road stretch is divided into overlapping segments (windows), which consist of 600 samples in succession, comprising a length of 60 m with a step length of 10 m between two subsequent segments. Some windows include special driving maneuvers such as strong acceleration actions, gear changes and steering movements, all of which influence vehicle dynamics. Therefore, the data within these windows does not solely reflect road roughness. To eliminate this influence of the car's driver, additional data sources are monitored to identify such segments. Only those windows in which the brake and clutch pedals are inactive, and those in which the steering wheel angle does not exceed $\pm 20°$, are presented as training input to the machine learning models. In total, 1,030 road segments fulfill these requirements and comprise the set of data available for further processing.

The statistics computed for each remaining window are the mean $\hat{\mu}$, standard deviation $\hat{\sigma}$, range $\hat{\Delta}$, and the mean of the short time energy $\hat{\mu}_{\mathrm{STE}}$ for each variable. The latter is primarily used in the field of speech and voice recognition for distinguishing voiced and unvoiced parts of a signal (Rabiner and Schafer 2007). Furthermore, it includes the power spectral densities f_i for the frequency bands $0.1(i-1)\,\mathrm{Hz} \le f \le 0.1i\,\mathrm{Hz}$ for $1 \le i \le 5$ computed with Welch's method. Consequently, 63 raw features are obtained.

The final choice of features for training was based on the similarity of the statistics derived from simulated and measured data. The distribution of the frequency band f_1 and the means $\hat{\mu}$ of the windows showed significant differences, likely due to inaccuracies in the simulation. These features were thus excluded. Additionally, to reduce correlations within the training data, only the features for the rear wheels were chosen. In total, this leads to 35 features as input for the machine learning models, which include the $\hat{\sigma}$, $\hat{\Delta}$, $\hat{\mu}_{\mathrm{STE}}$ statistics as well as the power spectral densities f_2 to f_5 for a_x, a_y, a_z, ω_{rl} and ω_{rr} respectively.

In a final processing step, the remaining input features and target variables are standardized by computing the t-statistic $[X - \hat{\mu}(X)]/\hat{\sigma}(X)$ for each variable X, mapping the mean to zero and variance to one. This provides a roughly uniform scaling between the variables and concentrates the feature vectors in an area around the origin of the high-dimensional space containing the data.

Multilayer Perceptron

The first machine learning model studied in this section is a multilayer perceptron (MLP), a special kind of feedforward ANN. The vast amount of literature about ANNs and the varied applications demonstrate that they are a well-established tool for the modern engineer (surveys are given in e.g., Huang and Zhang 1993 or Zhang 2000). Despite the drawbacks of MLPs and ANNs in general, such as being a *black-box* model and the difficulty of choosing network properties (hidden layers, transfer functions), their advantages make them suitable for the purpose of estimating the road roughness as demonstrated in, e.g., Ngwangwa et al. (2010). This includes the ability of MLPs to approximate any function to arbitrary accuracy (see Hornik et al. 1989; Cybenko 1989). In particular, they can serve as a model for the nonlinear relations ubiquitous in nature. No assumptions about the underlying distribution of the data are necessary. For these reasons, the MLP was

included as a reference for comparisons with the other machine learning models. The networks were implemented using the *MATLAB* neural network toolbox (Version 7.0.2/R2011b).

Two separate network architectures were employed to estimate the target variables Δ_{wLP} and σ_{wLP}. This has the advantage that their topology and training parameters can be tuned individually, based on the target variable. Additionally, it allows a fair comparison with the other models, which can only handle a single target variable. Thus, the output layer of the networks comprises a single neuron corresponding to the estimation based on the input to the network. The inputs of the MLPs were selected as described above, resulting in 35 input neurons. After manual tuning and testing, network architectures with two hidden layers were chosen. For the estimation of σ_{wLP} these layers contain 8 neurons and 1 neuron, respectively (35-8-1-1 topology). The second target Δ_{wLP} is in general less smooth and requires a more complex topology 35-15-10-1 to yield a similar estimation accuracy. Both network structures feature fully connected layers as well as hyperbolic tangent transfer functions in the hidden layers and a linear transfer function in the output layers.

During the networks' training the regularized mean squared error (MSE) of the network outputs and the target values is minimized. The regularization term helps to make the model less prone to overfitting to the set of training patterns. Error minimization is achieved using backpropagation (Rumelhart et al. 1986), with an optimization method based on the Levenberg-Marquardt algorithm (Levenberg 1944; Marquardt 1963). It is implemented as the trainlm function of the neural network toolbox. The authors manually tuned the damping parameters, which control whether the algorithm is closer to the Gauss-Newton method or gradient descent. The final networks were trained with $\mu = 0.001$ (initial damping), $\mu_{\text{inc}} = 10$ (iterative increment of damping during optimization step) and $\mu_{\text{dec}} = 0.1$ (final decrement of damping after finished optimization step) for σ_{wLP} respectively $\mu = 0.01$, $\mu_{\text{inc}} = 20$, $\mu_{\text{dec}} = 0.1$ for Δ_{wLP}.

However, as with many other nonlinear optimization algorithms, the Levenberg-Marquardt algorithm is not guaranteed to find a global minimum for the performance function. To account for that, a network architecture was trained multiple times on a fixed set of feature vectors. The random initialization of the weight and bias values of the neurons provides different starting points on the high-dimensional error surface for each run, thus making it possible to search various regions for local minima. This simple ensembling technique yields a number of slightly different estimator models, from which the best are chosen and their outputs averaged to provide a more robust estimation. This averaged network structure is further denoted as aMLP.

In this work, 20 to 30 networks were trained and evaluated using the mean absolute difference, the mean relative error, and the correlation coefficient of the network output and target data. A crossvalidation was applied to estimate the performance of a trained network given new input data. Subsequently, the MLPs were chosen according to their results on the validation set, with emphasis on the correlation coefficient, to build the final aMLPs. For both target variables, these consist of 10 networks each. For the crossvalidation, the set of feature vectors was randomly split into two disjoint sets. The training set, comprising 85% of the data or 876 cases, was used to train the MLPs. To avoid the occurrence of overfitting to this set, the performance on the remaining 15% or 154 cases provided an early stopping criterion for the training procedure. If the MSE on this set of data grows or stagnates, the training was stopped before reaching the maximum number of allowed iterations (1,000 in this work).

Support Vector Regression

Support vector regression (SVR) (Vapnik et al. 1997) is an extension of support vector machines (SVM) (Schölkopf and Smola 2002; Abe 2005), which can be used for various function approximation and time series prediction applications. They are closely related to the previously described MLP networks. For the latter, an unconstrained, nonconvex optimization problem is solved to obtain the weights, while SVMs are based on a quadratic objective function with linear constraints.

In the so-called ε-SVR, the aim is to find a function $f(x)$ that is both as smooth as possible and deviates maximally from the targets y_i by ε. For this purpose the input space is mapped into a high-dimensional feature space, where an optimal hyperplane defined by

$$f(x) = w^T h(x) + b$$

is determined. Here, h is the mapping function into the d-dimensional feature space, $w \in \mathbb{R}^d$ denotes a weight vector and b is the bias term. An ideal estimation is achieved if

$$|y_i - f(x_i)| \leq \varepsilon, \quad \forall\ i \in \{1, \ldots, n\}$$

To allow model errors, nonnegative slack variables $\xi_{+,i}$ and $\xi_{-,i}$ are introduced. This yields the quadratic optimization problem

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_{+,i} + \xi_{-,1}) \rightarrow \min$$

subject to the constraints

$$y_i - w^T h(x_i) - b \leq \varepsilon + \xi_{+,i}$$

$$w^T h(x_i) + b - y_i \leq \varepsilon + \xi_{-,i}$$

$$\xi_{+,i}, \xi_{-,i} \geq 0$$

for the so-called L_1 soft-margin SVR. The constant $C > 0$ is a tradeoff parameter between the training error and model complexity. The SVR is expressed by a convex quadratic optimization problem and thus, if feasible and bounded, has a global optimum. The minimization problem can be solved by quadratic programming techniques.

To avoid the computation of all the required scalar products in the high-dimensional feature space, a kernel function

$$K(x_i, x_j) = h(x_i)^T h(x_j) \tag{1}$$

is evaluated in input space, which is known as the *kernel trick*. A popular choice for the kernel is the radial basis function (RBF) kernel

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\gamma}\right)$$

with a kernel parameter $\gamma > 0$.

The accuracy of an SVR model is largely dependent on the selection of the model parameters, such as ε, C and γ for RBF kernels. We conducted the analysis of the SVR models in the statistical computing environment R (R Development Core Team 2012) using the add-on package kernlab (Karatzoglou et al. 2004). The hyperparameters C and γ were tuned using a 10-fold cross-validated grid search. As final parameters we used $\varepsilon = 0.1$, a kernel bandwidth $\gamma = 0.016$ and for the regularization parameter $C = 0.5$ for the estimation of Δ_{wLP} and $C = 1$ for σ_{wLP}, respectively.

Random Forests

Random forests (RF) (Breiman 2001) are an advancement of single classification and regression trees (CART) (Breiman et al. 1984), which follow a simple nonparametric regression approach. The feature space is recursively partitioned into rectangular regions. Observations with similar response values are grouped and a constant value is predicted within each region. Appropriate predictors variables and splitting points are selected by calculating some measure of node impurity. The variable that leads to the largest reduction of impurity is chosen. CART that are too large are susceptible to overfitting. This issue is usually addressed by pruning methods.

RF are so-called ensemble methods, because an ensemble of CART are aggregated for prediction (majority vote for classification and average for regression). One crucial point is that the trees are constrained to be very simple. Each tree is grown to maximal depth and no pruning of the trees is applied. For each tree a random subset of the original data, e.g., a bootstrap sample, is used for training. Furthermore, at every split, only a subset of m predictors is considered for use. RF can handle a large numbers of explanatory variables effectively without overfitting, even in the presence of complex interactions. The value m and the number of trees in the forest are tuning parameters of this method. However, it is not uncommon to perform well with default values, i.e., without additional parameter tuning (Liaw and Wiener 2002). To investigate the performance of RF models, the R-package randomForest (Liaw and Wiener 2002) was used. For both target variables, the main tuning parameter $m = 5$ was chosen.

Table 3. Performance Indicators for Simulated Input from the Entire Training Data Set

Output	Model	MAD	MRE	ρ
σ_{wLP}	aMLP	0.615	0.104	0.955
	SVM	0.600	0.097	0.949
	RF	**0.313**	**0.051**	**0.987**
Δ_{wLP}	aMLP	8.950	0.181	0.896
	SVM	8.095	0.148	0.880
	RF	**4.861**	**0.095**	**0.958**

Note: Best results are highlighted in bold.

Results

Given one of the target variables σ_{wLP} or Δ_{wLP} in a vector y of length n and a corresponding estimation $\hat{y}$, the three models were compared by the following statistics: the mean absolute difference (MAD) $(1/n)\sum_{i=1}^{n}|\hat{y}_i - y_i|$ given in millimeters, the mean relative error (MRE) $(1/n)\sum_{i=1}^{n}|\hat{y}_i - y_i|/y_i$, and the correlation coefficient (ρ) of $\hat{y}$ and y.

To show training efficiency of the machine learning models, the performance indicators on the training data set are given in Table 3. The results indicate that the models perform better on estimating σ_{wLP}. This is not surprising, considering that Δ_{wLP} includes very sharp peaks and is much less smooth in contrast to σ_{wLP}. For the latter target variable, the RF achieves the highest correlation and lowest MRE and MAD overall. In comparison to the other two models, the RF also shows the best performance for estimating Δ_{wLP}. A possible explanation is that a MLP and a SVM both use smooth functions in their estimation, while the RF provides a piecewise constant regression. Hence, the latter is better suited to capture the highly irregular target variables during training.

To evaluate the performance of the employed machine learning models concerning unknown data, we completely excluded one road stretch with a length of about 1,000 m from the training procedures. Its middle and end sections include a very rough profile, which demonstrates the difficulty of the estimation of the wLP. The models were tested both on data from simulations and measurement data collected by the probe vehicle. The model results for this road are depicted in Fig. 3. Overall, the characteristics of the target variables are captured by all three of the models. The peaks indicating pavement irregularities are detected, although not reproduced exactly. This is especially apparent for measured input data. The SVM and RF systematically underestimate σ_{wLP} on this stretch of road. The performance indicators for the validation stretch are given in Table 4. The various models yield similar results. For simulated input, the aMLP gives the best estimation of σ_{wLP} overall, but much worse performance for Δ_{wLP}. However, the other models are able to estimate both target variables with similar accuracy, with the SVM being the best model for Δ_{wLP}.

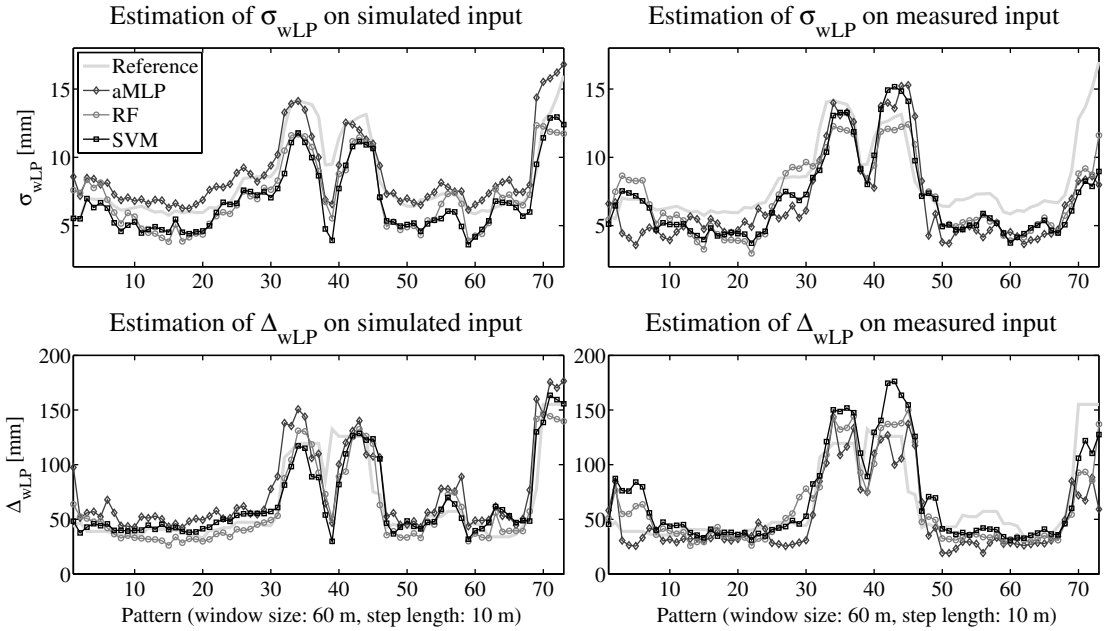

Fig. 3. Comparison of the employed machine learning models on the validation road stretch

J. Comput. Civ. Eng., 2014, 28(4): 04014015

Table 4. Performance Indicators for Simulated and Measured Input from the Validation Road Stretch

		Simulated input			Measured input		
Output	Model	MAD	MRE	ρ	MAD	MRE	ρ
σ_{wLP}	aMLP	**1.030**	**0.129**	0.896	2.072	0.255	0.798
	SVM	1.500	0.184	0.872	**1.674**	**0.209**	**0.843**
	RF	1.528	0.181	**0.910**	1.772	0.213	0.820
Δ_{wLP}	aMLP	17.032	0.304	0.865	18.147	0.276	0.768
	SVM	**10.918**	**0.176**	**0.895**	**16.135**	**0.252**	0.814
	RF	11.386	0.183	0.858	17.514	0.276	**0.833**

Note: Best results are highlighted in bold.

It is not surprising that the simulated input produces superior estimation results to the measured input. Real-world measurements usually underlie uncertainty or noise, which is not modeled in the simulation framework. Some inaccuracies are introduced by the approximation of the road, tire, and vehicle model. Another reason can be found in the driver behavior, i.e., speed choice or steering that differs from the ideal driver model.

Comparing the model outputs for simulated and measured inputs, the RF and especially the SVM seem to be robust models for estimating σ_{wLP}. They produce satisfactory results in both cases, in contrast to the aMLP, which suffers a significant performance drop. Regarding the estimation of Δ_{wLP}, the aMLP's performance does not decrease as much as for the two other models. These capture the characteristics of the simulated input better and thus yield less accurate estimations from measured input. Although the difference of the models' performances is small, the SVM produces the best results for measured input.

Accuracy aside, one must also note the differences in training and behavior of the three machine learning models. Choosing a set of representative features is important for each model, but the RF and SVM are found to be more stable than the aMLP with respect to the quantity of features. For example, a drastic reduction of features (as low as 5, chosen by recursive feature selection) does not significantly alter the results for the former two, while a neural network is not able to capture the structure of the data efficiently in this case.

Another comparative aspect is the parameter selection of the model and its training algorithm. For estimating the road roughness indicators as presented in this paper, tuning the settings of the RF and SVM has little impact on the performance. On the other hand, for MLPs there is no *standard* topology or optimization algorithm. Additionally, it relies heavily on randomization (splitting of the training set), similar to the RF. But the latter implements an ensembling technique, which minimizes the impact of this randomization, while the MLP's convergence and performance depend crucially on the random starting point on the error surface. The simple averaging procedure described above alleviates this problem. This leads to a longer total training time for the MLPs than for the other machine learning models, during which many configurations need to be tested.

It must be noted that this is an arbitrarily chosen validation set, which may not be representative. Future work will comprise a more comprehensive validation with a larger dataset of unknown road stretches.

Conclusions and Outlook

This paper discussed three machine learning models—MLP, RF, and SVM—for estimating road roughness in order to compare their performance and practicability. The estimation target is the wLP, which is a promising indicator for describing longitudinal roughness problems. Training data was derived from vehicle response simulations on different road tracks with a total length of 30 km. Therefore, the authors developed a simulation framework for creating realistic road, vehicle, and driver models. In total, 35 relevant features were extracted from the three axes of acceleration and the angular velocities of the vehicle's rear wheels. The authors then evaluated the machine learning models' estimation output with a road stretch that was independent to the training procedure. In general, all three models yielded similar satisfactory results. The SVM achieved the highest correlation and lowest mean absolute difference between estimated and true values. When reducing the number of features and therefore computation effort for training, the RF and SVM were more stable than the MLP.

Considering the many factors contributing to the results, such as parameters of the simulation, the road model as well as training the models, the estimations on the test road stretch show that the approach presented in this paper is feasible at constant vehicle speed. Single pavement irregularities as well as periodical roughness problems can be detected. In future work, we will simulate different vehicle models with varying speed to achieve a higher variability in training data. Moreover, a more detailed road model will be subject to future work. The proposed method enables road network monitoring done by conventional passenger cars, which can be seen as a supplement to prevalent road measurements with cost-intensive mobile devices.

References

Abe, S. (2005). *Support vector machines for pattern classification*, 1st Ed., Springer, London.

Andreasson, J. (2011). "The vehicle dynamics library: New concepts and new fields of application." *Proc., 8th Int Modelica Conf.*, Technical Univ., Dresden, Germany, 414–420.

Attoh-Okine, N. O. (1999). "Analysis of learning rate and momentum term in backpropagation neural network algorithm trained to predict pavement performance." *Adv. Eng. Softw.*, 30(4), 291–302.

Awasthi, G., Singh, T., and Das, A. (2003). "On pavement roughness indices." *J. Civ. Eng.*, 84, 33–37.

Breiman, L. (2001). "Random forests." *Mach. Learn.*, 45(1), 5–32.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*, CRC Press.

Cybenko, G. (1989). "Approximation by superpositions of a sigmoidal function." *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4), 303–314.

Gáspár, P., and Nádai, L. (2007). "Estimation of dynamical parameters of road vehicles on freeways." *5th Int. Symp. on Intelligent Systems and Informatics*, Institute of Electrical and Electronics Engineers (IEEE).

Gonzales, A., O'Brien, E., Li, Y., and Cashell, K. (2007). "The use of acceleration measurements to estimate road roughness." *Veh. Syst. Dyn.*, 46(6), 483–499.

Guarneri, P., Rocca, G., and Gobbi, M. (2008). "A neural-network-based model for the dynamic simulation of the tire/suspension system while traversing road irregularities." *IEEE Trans. Neural Network.*, 19(9), 483–499.

Harris, N., Gonzalez, A., O'Brien, E., and McGetrick, P. (2010). "Characterisation of pavement profile heights using accelerometer readings and a combinatorial optimisation technique." *J. Sound Vib.*, 329(5), 497–508.

Hirschberg, W., Rill, G., and Weinfurter, H. (2007). "Tire model TMeasy." *Veh. Syst. Dyn.*, 45(sup1), 101–119.

Hornik, K., Stinchcombe, M., and White, H. (1989). "Multilayer feedforward networks are universal approximators." *Neural Networks*, 2(5), 359–366.

Huang, S. H., and Zhang, H.-C. (1993). "Neural networks in manufacturing: A survey." *Electronic Manufacturing Technology Symp., 1993,*

15th IEEE/CHMT Int., Institute of Electrical and Electronics Engineers (IEEE), 177–190.

Hugo, D., Heyns, P., Thompson, R., and Visser, A. (2008). "Haul road defect identification using measured truck response." *J. Terramechanics*, 45(3), 79–88.

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004). "kernlab—An S4 package for kernel methods in R." *J. Stat. Software*, 11(9), 1–20.

Levenberg, K. (1944). "A method for the solution of certain problems in least squares." *Q. Appl. Math.*, 2, 164–168.

Liaw, A., and Wiener, M. (2002). "Classification and regression by randomForest." *R News*, 2(3), 18–22.

Marquardt, D. (1963). "An algorithm for least-squares estimation of nonlinear parameters." *SIAM J. Appl. Math.*, 11(2), 431–441.

Maurer, P. (2007). "Correlation of skid resistance values and braking deceleration of passenger cars." *ATZ Worldwide*, 109(5), 24–28.

Ngwangwa, H., Heyns, P., Labuschagne, F., and Kululanga, G. (2010). "Reconstruction of road defects and road roughness classification using vehicle responses with artificial neural networks simulation." *J. Terramechanics*, 47(2), 97–111.

Nitsche, P., Stütz, R., and Helfert, M. (2011). "A simulation-based concept for assessing the effectiveness of forgiving roadside treatments." *Adv. Transp. Stud.*, 2011, 87–98.

Rabiner, L. R., and Schafer, R. W. (2007). "Introduction to digital speech processing." *Found. Trends Signal Process.*, 1(1–2), 1–194.

R Development Core Team. (2012). "R: A language and environment for statistical computing." R Foundation for Statistical Computing, Vienna, Austria, ⟨http://www.R-project.org/⟩.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). "Learning representations by back-propagating errors." *Nature*, 323(6088), 533–536.

Sayers, M. W., and Karamihas, S. M. (1997). *The little book of profiling*, Univ. of Michigan Transportation Research Institute, Ann Arbor, MI.

Schölkopf, B., and Smola, A. J. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*, MIT Press, London.

Ueckermann, A. (2005). "Das bewertete Längsprofil." *Straße + Autobahn*, 56(1), 18–24.

Ueckermann, A., and Steinauer, B. (2008). "The weighted longitudinal profile—A new method to evaluate the longitudinal evenness of roads." *Road Mater. Pavement Des.*, 9(2), 135–157.

Vapnik, V., Golowich, S. E., and Smola, A. J. (1997). "Support vector method for function approximation, regression estimation, and signal processing." *Adv. Neural Inform. Process. Syst. 9—Proc. 1996 Neural Inform. Process. Syst. Conf. NIPS 1996*, MIT Press, Cambridge, MA, 281–287.

Yamabe, S., Hayashi, R., Nakano, K., and Suda, Y. (2010). "Estimation of road information from running vehicle." *Int. Symp. on Innovations in Intelligent Systems and Applications*, Istanbul, Turkey.

Zhang, G. P. (2000). "Neural networks for classification: A survey." *IEEE Trans. Syst. Man. Cybern. C Appl. Rev.*, 30(4), 451–462.