A scalable, self-supervised calibration and confounder removal model for opportunistic monitoring of road degradation

Wout Van Hauwermeiren1 | Karlo Filipan2,3 | Dick Botteldooren1
Bert De Coensel1,2,†

1 WAVES, Ghent University, Ghent, Belgium
2 Roads, ASAsense BV, Brugge, Belgium
3 Acoustics Lab, Catholic University of Croatia, Zagreb, Croatia

Correspondence
Dick Botteldooren, WAVES Lab, Ghent University, Technologiepark 126, 9052 Ghent, Belgium
Email: Dick.Botteldooren@ugent.be

1 Died September 15, 2021.
[Correction added on 25 March 2022 after first online publication: “Confonder” is changed to “confounder” in article title]

Funding information
Research Foundation Flanders, Grant/Award Number: G0D5215N

Abstract
Assessing road degradation typically requires specialized hardware (such as laser profilometers) or labor-intensive visual inspection. To facilitate large-scale, timely inspection of road surfaces, opportunistic sensing is proposed: Sound and vibration measurements are obtained from vehicles that are on the road for other purposes than measuring road quality. Prior work has addressed the problem of calibration and measurement noise removal from this abundance of measurements for a small number of measurement vehicles that drive on the same roads. However, as the deployment of opportunistic monitoring progresses, the applied techniques suffer from scalability. Here, a scalable self-supervised calibration and confounder removal (SCCR) algorithm is introduced. It allows to self-calibrate even if the data collection is done in distinct geographic areas and is capable of generalizing to vehicles not encountered during the training phase. Several model design alternatives are explored. After the application of SCCR, supervised training on a small subset of roads allows to predict observations made by standardized techniques also in areas where the latter have not been performed. The approach is tested and validated with 41 cars driving on 23,000 km of roads.

1 | INTRODUCTION

Road authorities decide which road pavements to refurbish or replace through pavement network maintenance and rehabilitation programs while taking into account budget constraints, user disturbance, and desired minimum quality (Sun et al., 2020). Preventive maintenance strategies lead to life cycle cost reductions and reduction in carbon emissions (Zulu et al., 2020). As an example, a bituminous mixture can be put on top of a lightly degraded asphalt road surface (BRRC-a, 2020). In this way, water resistance and skid resistance increase, beginning cracks are countered, and road evenness is restored. In Belgium, the total length of local roads surpasses the total length of main arteries by a factor of 10, therefore efficient quality monitoring strategies must be put into place (Belgian Government–FOD mobiliteit, 2010).

A road profile is defined as “a two-dimensional slice of the road surface, taken along an imaginary line” (Saywers & Karamihas, 1998). Several metrics are available to characterize the longitudinal profile of pavements. The road administrators of countries, cities, and regions...
have adopted different standards. International standards include ISO, 13473-1:2019 (mean profile depth), ISO, 8608:2016, and EI926 (ASTM, 2015; international roughness index (IRI)). Country-specific standards include: unevenness coefficient (EC; vlakheidscoëfficiënt;nl, coefficients de planéité;fr) in Belgium, NBO (notes de bandes d’ondes;fr) in France and weighted longitudinal profile (WLP) in Germany and Austria. The 2019 standard of the European Committee for Standardization describes calculation methods for IRI, root-mean-square, and WLP. Visual inspection includes the pavement condition index, which classifies roads on a numerical scale between 0 and 100 (ASTM D6433-20, 2020). These quantifiers can be integrated into a pavement management system as demonstrated by Pasindu et al. (2020) with IRI for low-volume roads.

There are many instruments suitable for measuring a road profile (Sawyers & Karamihas, 1998). The following parameters are of great importance: the reference height, the width of the profile, the chosen line (e.g., the center of the road or along the right or left wheel track), and the sampling interval (e.g., 1 mm for texture measurements). Among instruments, there are static methods (rod level method) and dynamic methods (inertial profiler). The latter is mounted on a vehicle that moves at a constant velocity. In turn, this enables the sampling of entire road networks. The reference height is calculated from accelerometer measurements and the profile elevation is usually measured by a non-contact method such as a laser. From the measured profile, an index such as EC, NBO, or IRI can be calculated.

Several published studies report the use of inertial measurement units (IMUs) in passenger cars as an alternative for expensive hardware such as laser profilometers. In some studies, a physics-based model of the vehicle is used to train an inverse model which estimates the road. The inverse model may consist of a recurrent neural network (Kim et al., 2021; Ngwangwa et al., 2010; Z. Zhang et al., 2018), or a convolutional neural network (CNN; Jeong et al., 2020), or an augmented Kalman filter (Xue et al., 2020; Zhao et al., 2019). In literature, examples of car dynamics modeling include a quarter-car, a half-car, and a linear pitch plane model among others. These models predict motion using masses, springs, and spring dampening. The difficulty arises in measuring the parameters of the cars without human intervention as the parameters vary by vehicle type and brand. In Ngwangwa et al. (2010), the parameters that are used were experimentally estimated for a single haul-truck in Haul (2005). In Zhao et al. (2019), this problem is approached by driving over a hump of known size: car parameters of three cars are then estimated using a genetic algorithm. In Xue et al. (2020), driving at a constant velocity on an unknown stretch of road (without any distinct defects such as potholes) is sufficient to fit the parameters of the half-car model. In Jeong et al. (2020), ground-truth IRI measurements reconstruct the road profile in 10-cm intervals using the power spectral density (PSD) function-based method as described by Tyan et al. (2009). In addition, a velocity profile of the trajectory is recorded during several actual passages by a variety of 60 vehicles (Heydinger et al., 1999). The authors proposed to train a CNN on the inverse of the simulated vehicle responses to the road and velocity profile. Features of the inverse model are the time series of instantaneous acceleration, velocity, and angular velocity at a virtual location on the dashboard of the car. However, the study only demonstrates the technique by simulating the accelerometer and gyroscope responses of different car makes.

On the other hand, a study by Alhasan et al. (2016) extracts wavelet-based features from vertical acceleration samples. The authors explored the design of wavelets inspired by the quarter-car model, to assess overall profiles and localized roughness features. The main advantage is the ability to assess short road profile segments (smaller than 7.62 m). Additionally, Mirtabar et al. (2020) calculate IRI using double integration of the vertical acceleration measured by an Arduino sensor board embedded in a vehicle. A 0.82 coefficient of determination and an average error of 15% is obtained, compared with a reference profiler method.

Yang et al. (2020) use many different smartphones of different models that have accelerometers with different sampling rates. This study quantifies the ride quality with the road impact factor (RIF) transform. RIF takes into account the vertical acceleration and the horizontal velocity of the car. In addition, RIF has a linear relationship to the IRI quantifier under identical observation conditions. The authors have observed that when RIF is measured by different devices under the same driving conditions, the resulting RIF is normally distributed, however, with a different mean value per device. Hence, calibration with the reference mean is proposed.

Several other papers focus on classifying potholes and other distinct road features (e.g., speed bumps) using supervised learning. Varona et al. (2020) monitor potholes and identify road surface types using smartphone accelerometer sensors. The authors compare reservoir computing models, CNN, and long-short term memory neural networks, which are directly applied to the time-series data. In addition, stability events are identified: opening and closing of the doors, driving over speed bumps, answering a call on the phone, and so forth. On the other hand, Baldini et al. (2020) extract a spectrogram from the time window of the gyroscope and accelerometer samples, and a CNN is used to classify them. The authors validate their approach with 12 distinct vehicles (six among
them are Fiat Panda cars) and 20 loops per car on a 2-km test track. An accuracy of 97.1% has been reported in detecting potholes, cracks, transverse cracks, rumble strips, and speed bumps. Although the reported collected dataset is quite large, it is limited to the test track with many passages passing on the same physical road feature. Similarly, Bello-Salau et al. (2018) apply a wavelet decomposition to noisy vertical accelerometer time windows and obtain different scale lengths using a method called wavelet transformation scale-space filtering. These features are used to classify potholes that result in high precision and low false-positive rate.

Menegazzo and Von Wangeheim (2020) investigate multiple models to distinguish between asphalt roads, dirt roads, or cobblestone roads using accelerometer, gyroscope, and global positioning system (GPS) speed data. The paper has considered different drivers and cars. To apply the traditional machine learning models such as support vector machines, k-nearest neighbors, and k-means clustering, the authors produced features with mean, variance, and standard deviation of the $x\gamma z$-accelerometer and gyroscope and the mean of the GPS speed. For deep learning models, they have used LSTM and CNN on the time-series data. They concluded that the CNN approach was the most stable with $F_1$-score of 90.85% for dirt roads, 85.84% for cobblestone roads, and 98.96% for asphalt roads. The study by El-Wakeel et al. (2018) focuses on classifying potholes, dents, railway crossings, manholes, cracks, speed bumps, and paved roads. They used a GPS, a gyroscope and an $x\gamma z$-accelerometer. A wavelet packet denoising was applied to the raw data, then several features were extracted. The SVM model reached a true positive rate of 90%.

Another approach of measuring the roads is driving with an IMU and simultaneously calibrating the device with a reference instrument (such as a 1D or a 2D laser profilometer). In Park et al. (2020), two IMUs per wheel (one on the knuckle and one at the top of the damper) and a 2D laser profilometer were installed in a passenger car. A fully connected multi-layer neural network transforms the PSD of the accelerometer data into ISO 8608 roughness classes for a 110-m road segment. Velocity is set at a fixed 40 km/h.

In addition to the previously discussed methods, one can also rely on image processing and machine learning approaches to assess the texture and roughness of the surface. For instance, Valikhaneni et al. (2021) quantify the surface roughness of concrete structures using images with sufficient resolution with the help of wavelets and machine learning. The authors reported an accuracy of more than 93% of the estimation of the concrete surface roughness. To avoid replacing the entire structure, degraded concrete structures (e.g., bridges, buildings) are repaired. On the other hand, Bang et al. (2019) use dashboard images in moving vehicles to obtain pixel-level road crack labels.

A CNN with transfer learning (ResNet-152) is found to achieve the best performance with recall 71.98%, precision 77.68%, and the intersection of union equal to 59.65%. Then, Hsiesh et al. (2020) propose a CNN model to assess jointed plain concrete pavement. They report an average precision of 85.42% on multi-label slab condition classification. Finally, A. Zhang et al. (2017) propose CrackNet, a CNN model without downsampling layers, to identify pixels in 3D laser images, which correspond to cracks in asphalt. The authors claim a precision (90.13%), recall (87.63%), and $F$-score (88.86%).

Opportunistic monitoring of road surfaces has previously been studied at Ghent University, where a fleet of cars was equipped with sound and vibration sensor boxes. The collected data have already been used to measure the influence of pavement characteristics on rolling noise on roads (Van Hauwermeiren et al., 2021). In this work, a denoising autoencoder (DAE) (Goodfellow et al., 2016), is introduced for calibrating noise measurements made by a fleet of cars and for removing modifiers and confounders (vehicle speed, acceleration, and temperature). One can categorize the DAE methodology under solutions for the blind sensor calibration problem (as introduced by Balzano & Nowak, 2007, in which a set of not-calibrated sensors need to be calibrated in an unsupervised manner, without ground truth). In Van Hauwermeiren et al. (2019), a vibration feature expected to be correlated to texture has been introduced: a 1/3-octave filter bank is applied on vertical accelerometer samples, then this spectrum is shifted to a corresponding spatial wavelength according to the vehicle speed. This feature is hypothesized to resemble the PSD of the road surface height. To account for different driving speeds, the proposed feature was corrected by a generalized additive model (GAM). Laser profilometer data have been used to calibrate this texture feature by linear rescaling.

This paper builds upon the earlier work presented in Van Hauwermeiren et al. (2019, 2021). The work presented here extends the proposed methodology to a scalable approach that allows to include a large number of measurement vehicles into the proposed road monitoring system.

The methodology used in the presented work is introduced in Section 2. In particular, in Section 2.3, adaptations are presented, which allow handling cars that drive on the non-overlapping sets of roads. Moreover, a method for generalizing the calibration using a set of typical measurements is proposed to account for the cars that the model did not encounter during training. Variations of the neural network architecture are introduced in Section 2.4, and a texture feature calculation is presented in Section 2.6.

The dataset obtained for this work is presented in Section 3. Furthermore, results are shown and discussed in
Section 4. In particular, this part investigates several critical elements of the calibration and confounder removal model: generalization to unseen combinations of cars (Section 4.3) and a generalization to cars not encountered during training (Section 4.4). Finally, Section 4.6 validates the effectiveness of the self-calibrated measurements in predicting the ground-truth measurement of the EC. In Section 5, conclusions based on the presented results are given.

2 | METHODOLOGY

2.1 | Opportunistic data collection

The opportunistic approach implies that no dedicated measurement campaigns are being performed. Rather, it delivers measurements as a side effect of normal driving activity. To obtain a good coverage of the road network, it is advantageous to select the vehicles to equip with a sensor box carefully: rental vehicles, police cars, cars of salespeople, and small delivery vehicles typically cover a larger portion of the road infrastructure than the car of a daily commuter. Heavy vehicles and special-purpose cars such as fire trucks or garbage vans are avoided to limit the range of vibration transfer functions and driving conditions and thus to relax the requirements of the model.

The sensor box contains a GPS, an accelerometer, and a microphone and continuously sends its measurements to the server. On the server side, the GPS trace is map-matched to an OpenStreetMap road network on road segments of length 20 m (Trogh et al., 2020). As the GPS is not accurate enough to detect the lateral lane position, this is estimated on the basis of the lateral acceleration, $a_y(t)$, by identifying lane-changing motions. Figure 1 presents the general layout of the approach.

2.2 | Problem statement and model approach

Compared to well-controlled measurements, opportunistic measurements suffer from a few shortcomings:

1. Although the measurement equipment on itself is well-calibrated, the way it is installed in a variety of measurement vehicles may differ, and hence a calibration of the whole system including the so-called car transfer function of the vehicle is needed.
2. Measurement conditions are not well-controlled; this may influence the relationship between the observation and the underlying road quality (e.g., vehicle speed, tire temperature) or non-related noise may be picked up (e.g., engine vibration, music system).

The abundance of repetitions of the measurement with a variety of vehicles will need to be explored to remove these problems. Calibration (1) may be solved by deploying all vehicles on the same trajectory under the same driving and weather conditions, yet this approach is time-consuming and not scalable. Simple averaging or slightly more evolved statistical analysis may eliminate some of the dependencies (2) yet once more this approach does not scale well as it will require many passages on all roads considered, preferable under representative driving conditions and in all-weather situations. Hence, a modeling approach is proposed.

The model consists of two parts that are often found in modern machine learning approaches. The first step is an self-supervised step that solves the problems of mutual calibration and confounder removal and the second step relates the measurements to an indicator that has been agreed upon in (local) standards or common practice. The latter can be seen as a recognition and labeling problem. Using this combined self-supervised and supervised approach, the need for labeled data, reference measurements, is significantly reduced.

2.3 | Scalable self-supervised calibration and confounder removal for opportunistic measurements

To tackle the two issues mentioned in Section 2.2, the self-supervised model relies on the following assumptions:

1. Two measurements of the same quantity should give the same result. In particular, two mobile measurements of road quality are made under the same conditions at the same location within a reasonable time.
interval—where the probability of new damage or aging is small.

2. Known dependencies such as the driving speed and temperature—identified by a field expert—are independent of the local differences in road quality, at least within a certain category of pavement;

3. (Optional) On average, each measurement system performs the same range of measurements, that is, all cars statistically speaking drive on the same type of pavements with the same average aging and damage characteristics.

These assumptions are only valid in general terms and may fail in specific cases. Assumption (1) will generally hold for overall aging effects (e.g., stone loss) and mild defects (e.g., alligator cracks) while strong local defects (e.g., potholes) may be avoided by some cars and not by others. Hence, any training and model evaluation should allow for such outliers.

Several variants of the proposed scalable self-supervised calibration and confounder removal (SCCR) model are investigated and compared in this publication.

SCCR1 (Figure 2a) relies solely on assumptions (1) and (2) and extends previous work (Van Hauwermeiren et al., 2021) to many cars. It implements a modified use of a DAE neural network.

A DAE is typically trained to map a set of noisy observations obtained under non-standard conditions, \( X(t_1) \), to clean observations at a reference condition, \( Y(t_2) \). To facilitate the task of the DAE, an identifier of the car that measured \( X(t_1) \), and the known or expected dependencies that will finally define the reference conditions, \( CC_1 \), are added.

The deep neural network layout is given in Figure 2a. Note that information on the target car \( C_2 \) and the target conditions \( CC_2 \) are added to the bottleneck node information.

For the application at hand, no clean reference data are available for training (note that this is the self-supervised part of the model); hence, the typical encoding-decoding architecture of a DAE is kept while its application is significantly modified. Henceforth, the approach shall be referred to as SCCR model.

To train the model, pairs of passages at a certain road segment are formed. Passages on the same road segments should contain the same hidden, spatial information. However, different observation conditions apply: for example, acceleration, deceleration, high and low velocity. Moreover, properties of the car such as the weight distribution, current gear, and tire pressure influence the measurements. The SCCR is expected to transform measurements to account for these conditions, (1) by explicitly correcting for the dependencies described in \( CC_1 \); (2) by encoding only common factors related to the only spatial dependence: the road surface, in the bottleneck nodes. In other words, with this alternative training approach, it is expected that \( c_1 \) and \( CC_1 \) are decoded and their dependence is removed in the encoder part while \( c_2 \) and \( CC_2 \) dependencies are reintroduced in the decoder part.

The underlying spatial information can change in time due to road degradation or road maintenance; hence, a time limit between the passages is added: \( |t_2 - t_1| \leq \Delta t \). This avoids that the model learns and removes changing road conditions. There is a trade-off in the choice of \( \Delta t \): between having more training data (pairs of measurements made by different devices) or avoiding picking
### TABLE 1  Chosen parameters for the different flavors of SCCR

| Configuration                      | SCCR1 | SCCR2a-Dense | SCCR2a-ReLU | SCCR2a-Gated | SCCR2b |
|------------------------------------|-------|--------------|-------------|--------------|--------|
| Number of hidden layers in encoder | 3     | 3            | 3           | 3            | 3      |
| Activation function encoder/decoder| Tanh  | ReLU         | ReLU        | None         | None   |
| Number of bottleneck nodes ($N_b$) | 40    | 40           | 40          | 40           | 40     |
| Activation function bottleneck     | Tanh  | None         | None        | None         |        |
| Number of hidden layers in decoder | 3     | 3            | 3           | 3            | 3      |
| Number of hidden nodes per layer   | 600   | 600          | 400         | 400          |        |
| Skip connections with device       | No    | No           | Yes         |              |        |
| Device gate                        | No    | No           | Yes         |              |        |
| Learning rate                      | 5x10^{-5} |        |              |              |        |
| Dropout                            | 0.1   |              |              |              |        |
| (De)normalize globally             | All features | All context variables |              |              |        |
| (De)normalize by device            | None  | All input features and target features |              |              |        |
| Maximum time difference between observations on training ($\Delta t$) | 3 months |              |              |              |        |
| Input features                     | $T(f,t)$, $L_{EQ}(f,t)$, $dl_{min}(f,t)$, $dL_{95}(f,t)$, $L_{50}(f,t)$, $dL_{95}(f,t)$, $dL_{max}(f,t)$ |              |              |          |
| Context variables                  | $v(t)$, $a_x,min(t)$, $a_x,max(t)$, $\mu(a_x(t))$ |              |              |          |
| Target features                    | $T(f,t)$, $L_{EQ}(f,t)$, $dl_{min}(f,t)$, $dL_{95}(f,t)$, $L_{50}(f,t)$, $dL_{95}(f,t)$, $dL_{max}(f,t)$ |              |              |          |
| Identification of device           | One-hot encoded device number |              |              | 2D-latent space |
| Reference conditions               | $v_{ref}$ | Top 85th percentile of speed at that location |              |              |        |
| $\mu(a_x)$, $a_x,min$, $a_x,max$  | $0 \pm \sigma$ |              |              |              |        |
| Temperature                        | 15°C  |              |              |              |        |

observations with changing road conditions. In Table 1, the SCCR parameters are listed. Detailed information on hyperparameter tuning and choice of the neural network layout can be found in Van Hauwermeiren et al. (2021).

This basic model has the advantage of simplicity and it does not rely on assumption (3). However, it does not scale well to multiple cars and larger geographic areas, as it requires that all vehicles occasionally drive on the same road segments. For larger geographic regions, a few cars may occasionally drive on the same road but without careful balancing of the training dataset, these very few occasions will not affect the training of the SCCR. Hence, the basic model in this publication, SCCR1, already includes a balanced training set. It is classically obtained by relying on the statistics of overlapping trajectories and oversampling rare combinations. A threshold is applied to avoid oversampling a single accidental overlapping trajectory for cars usually deployed in distinct regions.

The second group of models, SCCR2, relies in addition on assumption (3) to alleviate the restriction of overlapping trajectories for self-calibration. In a first approach, labeled SCCR2a, the feature batch normalization that is typically used in deep learning to improve convergence is adapted: Normalization is applied for each measurement device separately. In this way, each measurement device only performs relative measurements, that is, relative to the surfaces it has encountered. If two sets of cars would drive on completely disjoint road infrastructure (e.g., on a different continent) and this road infrastructure on average also has a different quality, measurements could no longer be compared after the self-supervised calibration step; hence, a ground-truth measurement (see Section 2.6) would be needed in each cluster.

In the above models, a unique car identifier, $c$, is introduced in the model. Because of this, each time a measurement vehicle is added to the pool, a re-training part of the model becomes necessary. This re-training is not very intense for the central processing unit, but still, this approach is hardly scalable to massive deployment. Moreover, the possible advantage for cross-calibration between disjunct areas in case cars of the same make and model belong to both pools is not explored.

In SCCR2b, a solution is proposed to identify cars by their typical measurements rather than by a unique number. To this end, 100 measurements of features, $X$, and matching conditions, $CC$, are randomly selected. The order of these measurements is irrelevant for the classification task at hand, which leads us to build a model based on the
set2set model (Vinyls et al., 2016) to predict the unique car identifier (Figure 2c). The 2-node bottleneck of this car identifier model can be interpreted as a latent space representation of the car. The resemblance between cars can be detected in this latent space. The point in this latent space corresponding to a car is used as an input to the main part of the SCCR (Figure 2b). This point replaces the one-hot encoded label, c.

### 2.4 Neural network architectures

Different flavors of SCCR2a are explored (see Table 1): SCCR2a-Dense, SCCR2a-Gated, and SCCR2a-ReLU. SCCR2a-Dense has the same architecture as SCCR1. The device identifier is simply added as an input to the first hidden layer. During experimentation, saturation was apparent in the bottleneck nodes due to tanh activations and the dimensionality reduction. To this end, SCCR2a-ReLU is added to the comparison. Here, the bottleneck has got no activation function and a rectified linear unit (ReLU) is applied after every hidden layer. Last, SCCR2a-Gated is introduced. This model is inspired by multiplicative attention mechanisms (Vaswani et al., 2017). Every hidden layer $h_t$ in SCCR2a-Dense is multiplied by an attention gate $\sigma_c : h_t(X; CC,c) \ast \sigma_c$. The gate is a fully connected layer with sigmoid activation heads. This results in a vector $\sigma_c$ and produces a 0 to 1 value for every hidden node, based on the car identifier. In addition, the hidden layer multiplied with the gate is concatenated again with the device identifier (either input device $c_1$ in encoder and output device $c_2$ in decoder). The gate acts as a gradual On–Off switch, which has the power to disable certain parts of the neural network.

### 2.5 Averaging pseudo-measurements

The neural networks described above, when fully trained, allow to transform a measurement made by one device, $c_1$, under particular conditions, $CC_1$, to a pseudo-measurement by any of the other devices in the pool, $c_2$, at reference conditions, $CC_r$. More specifically, due to the way the networks are trained, the component of the measurement connected to location is transformed. Suppose $Y'$ is the transformation of model SCCRx. In general, averaging over multiple observations reduces measurement error, but—under mild assumptions—the measurement error reduces as the square root of the number of observations. With the proposed opportunistic approach, smaller roads will typically only be sampled by very few cars, and hence one cannot rely on averaging.

Therefore, mapping $Y'$ is introduced to transform each measurement $X(t)$ made by a car $c(t)$ under conditions $CC(t)$ at a location $x$, to a set of pseudo-measurements made by a group of leader cars $C_L$ and average over this group. This average over the pseudo-measurements can be referred to as the calibrated measurements $Y''(t)$, which is given by:

$$Y''(t) = 1/C_L \sum_{c_2 \in C_L} Y'(X(t), c(t) \rightarrow c_L, CC(t) \rightarrow CC, (x))$$

(1)

The reference conditions depend on the road segment $x$ as the reference vehicle speed is taken as $v_{85}$ on that segment, the speed exceeded by 85% of the cars driving on that segment. Finally, the calibrated measurements on a single location are averaged out:

$$Y_x(x) = \frac{1}{N(x)} \sum_{t \in X} Y''(t)$$

(2)

In small datasets, $C_L$ could contain all measurement devices. However, this approach does not scale well. Hence, devices $C_L \subset C_L$ are selected, such that the mean-squared error (MSE) between predictions based on these cars of any measurement made by another car is minimal when transforming to the neighboring cars:

$$1/N_n(c_L) \sum_{c_2 \in \text{neighbours}(c_L)} \text{MSE} \left(Y'(X_L, c_L \rightarrow c_2, CC), X_2 \right)$$

(3)

$N_n(c)$ is the amount of neighbors of $c$, where neighbors are defined as cars that drive at least on one common street. This means they hold enough information on the underlying properties of the road.

For the strategies SCCR1 and SCCR2a, the leader cars are selected based on their unique identifier; for strategy SCCR2b, a few highly probable points are picked in the latent space. This procedure relies on one important assumption: car $c_1$ can be transformed into any car $c_2$ using $Y'$. In the results sections, this is further explored.

A relative assessment can also be performed, in case there is an interest in the properties of the road in comparison with other roads of the same designed target driving speed. As actual driving speed data are readily available, $v_{85}$ is used and a GAM $G(v_{85})$ (Servén et al., 2018) is fitted on $Y_x(x)$. The indicator relative to the average road is defined as:

$$\Delta Y_x(x) = Y_x(x) - G(v_{85})$$

(4)

### 2.6 Feature extraction

Careful feature extraction will allow a data-driven model to be trained more efficiently. Crafting features can be seen...
as a way to include expert knowledge in the data-driven approach.

All the features that are extracted from the sensor box are listed in Table 2. 3D accelerometer samples are collected at 100 Hz. The x-axis is located in the traveling direction, the y-axis on the lateral direction, and the z-axis on the vertical direction. A 1/3 octave band filter bank is applied to $a_z(t)$ between 1 and 25 Hz. This results in $a_z(t)$ (logarithmic units). GPS records are collected at 1 Hz and include longitude, latitude, and vehicles speed. Sound is collected in 1/3 octave bands at an interval of 0.02 s. The minimum, maximum, median and $L_{EQ}(f, t)$ are taken from every band once a second. The data is then transmitted to a central server through a 3G connection. $L_{EQ}(f, t)$ and $a_z(f, t)$ are concatenated into a single spectrum $SV(f, t)$.

Vehicles driving at different speeds over a spatial pattern will experience vibrations excitation at different frequencies even if the underlying unevenness has the same spatial spectrum. Shifting from one feature (vibration spectral component) to another based on a third feature (driving speed) is something a neural network, deep or otherwise, cannot easily learn. Hence, this expert knowledge is explicitly included by adding texture features.

Texture features are obtained by shifting the frequency axis proportional to the driving speed. A similar technique has been recently applied for the case of railway vibrations (Pallas et al., 2020). This relationship encodes the notion that the road sounds higher in pitch and triggers car vibrations higher in frequency when you drive faster. For a given device $c(t)$ and characteristic length of the road surface structure, $\lambda$ ($2.5 \text{ cm} \leq \lambda \leq 10 \text{ m}$), the approximated texture $T$ is given:

$$T(\lambda, t) = SV\left(\frac{v(t)}{\lambda}, t\right) - H_c\left(\frac{v(t)}{\lambda}\right)$$

Cars have varying mass distribution, spring systems, tires, and geometry. Therefore, at some frequencies in the $SV(f, t)$ spectrum, peaks can be found corresponding to resonance frequencies that affect the transfer function between the tire and the sensor box. While shifting the spectrum, the amplitude could therefore change; hence, a car transfer function $H_c(f)$ is subtracted from the measurement prior to shifting. It is approximated by taking the average of $SV(f, t)$ for 3 h of driving of every car $c$ at a speed, $v \in [30 \text{ km/h}, 125 \text{ km/h}]:$

$$H_c(f) \approx \frac{1}{N} \sum_t a_z(f, t)$$

### Table 2: Set of features that are produced by the sensor box at an interval of 1 s ($\mu$ is the one-second average)

| Label | Definition | Reason for including this variable |
|-------|------------|------------------------------------|
| $a_{z,\text{min}}(t), a_{z,\text{max}}(t), \mu(a_z(t))$ | Vertical acceleration statistics | For example, potholes, bridge joints |
| $a_{y,\text{min}}(t), a_{y,\text{max}}(t), \mu(a_y(t))$ | Sideways acceleration statistics | Lane changing motions |
| $a_{x,\text{min}}(t), a_{x,\text{max}}(t), \mu(a_x(t))$ | Forward acceleration / deceleration statistics | Driving conditions |
| $L_{EQ}(f, t)$ | Equivalent sound level per 1/3 octave frequency band | Overall surface macro texture |
| $L_{\text{min}}(f, t), L_{05}(f, t), L_{50}(f, t), L_{95}(f, t)$ | Percentiles over 1 second of $L(f, t)$ | Changes surface macro texture |
| $a_{z}(f, t)$ | Spectral vertical acceleration in dB | Vibrations due to overall road surface waviness and macrotexture |
| $\lambda(t)$ | Spatial texture | See Section 2.6 |
| $c(t)$ | Device identifier | Calibration between devices |
| $v(t)$ | Speed | Driving conditions |
| $x(t)$ | Location from GPS | Mapping to road segments |
right-most lane. ARAN is equipped with three lasers and a vertical accelerometer to measure the road surface elevation. The relative reference height is computed by integrating the accelerometer readings two times over a set amount of time. Then, a low-pass filter is applied to the reference height (e.g., a moving average filter of $\lambda = 2.5$ m). The filtered reference height is subtracted from the laser signal. The EC, $EC_3$, is then given by integrating the remaining signal over a certain distance and dividing the result by a calibration constant $2^*$distance (BRRC-b, 2020). A lower value of $EC_3$ means a smoother road surface.

The ground-truth laser measurements are localized with GPS and aligned with the opportunistic measurements. A moving average filter of 200 m is applied to both $EC_3$ and the opportunistic measurements to remove laser measurement noise and to obtain a better match the opportunistic measurements. Due to the uncertainty on the actual location, the peak values of $EC_3$ could be misaligned with the opportunistic measurements (e.g., due to potholes). A GAM model is trained with a selection of features of $Y_\lambda$ and the target value $EC_3 (\lambda = 2.5 \text{ m}, \lambda = 10 \text{ m} \text{ and } \lambda = 40 \text{ m})$ at right ($r$) and left ($l$) wheel track.

Road administrators in Belgium put a threshold on EC to decide which roads are in bad condition and which roads are in good condition (BRRC-b, 2020). The thresholds depend on the maximum allowed velocity on the road: A lower threshold is set for high speeds. $EC_\lambda = 10$ m is only used for roads with a speed limit $\geq 40$ km/h, and $EC_\lambda = 40$ m is only used for roads with a speed limit $\geq 60$ km/h. In Flanders (Vlaams standaardbestek SB 250), the following thresholds are in place to indicate a good road surface:

- $EC_{\lambda=2 \text{ m}}: \leq 25 \leq 80 \text{ km/h}, \leq 40 \text{ (60 km/h)} \leq 80 \text{ km/h}, \leq 45 \leq 60 \text{ km/h)}$
- $EC_{\lambda=10 \text{ m}}: \leq 50 \leq 80 \text{ km/h}, \leq 80 \text{ (60 km/h)} \leq 80 \text{ km/h}, \leq 90 \leq 60 \text{ km/h})$
- $EC_{\lambda=40 \text{ m}}: \leq 25 \leq 80 \text{ km/h}, \leq 40 \leq (60 \text{ km/h)} \leq 80 \text{ km/h})$

2.8 Performance metrics

The performance of the overall model can be validated against a ground truth using standard metrics such as correlation coefficient and mean squared error. Standard recall and precision can be calculated for a classification task based on minimal requirements.

Validating the self-supervised part of the model is not as straightforward. A good measurement system should produce low measurement uncertainty while remaining sensitive to the characteristics of interest. To analyze the performance of calibration and de-noising, two metrics are introduced: $w(M)$, which measures the variability between repeated calibrated measurements $Y''$ and averages it over all measurement locations (road segments); $a(M)$, which measures the variability of average measurement results $Y_a$ over a wide range of locations (road segments). These metrics—which are introduced for rolling noise assessment in Van Hauwermeiren et al. (2021)—allow to obtain an indication on measurement uncertainty ($w$) and sensitivity ($a$) without knowing a ground truth. Note that the latter depends on the range of pavements that are assessed. Within this metrics system, the purpose of a calibration and de-noising step can then be formulated as an attempt to increase the ratio $r(M) = a(M)/w(M). M$ is either $a_x (f, t), a_{Z,DAE} (f, t), T(\lambda, t), \text{or } T_{DAE} (\lambda, t)$.

The above procedure, however, does not prevent drift. One could indeed imagine that on segments where measurements are performed by multiple devices, $w(M)$ remains small but if all measurement devices are no longer tested pairwise, a systematic error could still be accumulated.

To check for the latter, the performance of the mapping $Y'$ is validated by calculating the RMSE between the prediction based on $c_1$ of a measurement by a car $c_2$ at the same location and an actual measurement by $c_2$:

$$\text{RMSE}(c_1, c_2) = 1/N \sum x \text{RMSE}(Y' (X_1, c_1 \rightarrow c_2, CC), X_2)$$

(7)

Note that an accurate mapping is an indication of accuracy but not a necessary condition for $Y_a$ to be accurate, as calibration still includes an additional step.

3 DATA DESCRIPTION

The opportunistic measurements included in this paper have been collected between April 2020 and August 2021. Forty-one cars of various makes and models are included in the dataset. As seen in Figure 3, trips cluster around the hometowns of the owners (private or not) of the measurement vehicles. Hence, it is not guaranteed that each pair of cars have occasionally measured the same locations. To illustrate this, connectivity of a connected graph ($C, E$) is formed (see Figure 4) with $C$ the set of cars and $E$ the set of edges $(c_1, c_2)$. Each edge gets a weight $n$ that reflects the number of possible paired measurements between $c_1$ and $c_2$. It is the sum over all road segments of the product of the number of passages of $c_1$ and the number of passages of $c_2$ during the time interval $\Delta t$.

Seventeen cargo vans, 12 sedans, five light SUVs, and seven light vans (passenger cars) are included in the dataset. The full dataset contains 5979 h of continuous measurements at $> 20$ km/h. Between December 2020 and August 2021, 87% of the data has been collected as a result
of the start of the systematic rollout. The top-five cars have collected 57\% of the data. Four of these top cars are taxi vans and measure in the Port of Antwerp. Twenty-three

313 km of unique roads have been covered (multiple lanes are counted separately). The top 10\% of the covered segments are sampled with 28 passages or more, the top 20\% with 12 passages or more, 50\% of the segments are only sampled twice or less, and 35\% of the segments are only sampled once.

To train the SCCR models, a machine learning dataset is constructed. As mentioned in Section 2.3, due to the unbalanced connectivity between measurement devices shown in Figure 4, the training and test datasets need to be balanced. The assignment of data to the train and test set needs to guarantee maximal independence and selecting 20-m road segments randomly does not match this requirement. The OpenStreetMap way identifier mostly identifies longer road sections with possibly multiple lanes. Hence, 20\% of the covered ways are assigned to the test set and 80\% to the training set. This way identifier assures stronger independence between the train and test set. The training pairs are selected according to Algorithm 1 in the Appendix. The following design objectives are taken into account. (1) Each device pair \((c_1, c_2)\) needs to have approximately the same amount of samples in the training dataset. (2) Device pairs with fewer than 5000 samples are excluded. (3) The pairs are selected at random. (4) A scaling function assures that locations with high \(n_x\) do not dominate the training dataset. Note that the amount of possible pairs scales with \(O(n_x^2)\) with \(n_x\) number of passages per segment. This procedure results in a balanced dataset consisting of 327 device pairs with each around
50,000 examples in the train set and 10,000 examples in the test set.

4 RESULTS AND DISCUSSION

This results section focuses preliminary on the performance of different aspects of the self-supervised part of the model. To this end, the proposed models have been implemented in Keras with Tensorflow 2.6 GPU backend. All SCCR models have been trained until convergence on the test set with the Adam optimizer (Kingma & Ba, 2015) at a fixed learning rate of $5 \times 10^{-5}$.

Section 4.6 investigates in more detail the overall results for evenness.

4.1 Normalization by device and sampling procedure

In this section, a comparison between model SCCR1 and SCCR2a is explored. Both models have the same neural network backbone. However, the performance of the normalization by device (SCCR2a) is compared to the global normalization (SCCR1). To assess the performance between SCCR1 and SCCR2a, the RMSE is computed for each device pair and feature on the test set. The distribution of the difference in RMSE over the different device pairs is then shown in Figure 5; in addition, RMSE by device pair of SCCR2a is shown. The best reductions in RMSE (improvement) can be found in the $L_{eq}(f)$ feature. The major improvements affect 15% of the device pairs. $L_{eq}(1000 \text{ Hz})$ has a reduction of up to 2 dB. For $L_{eq}(20 \text{ Hz})$ a reduction of at least 0.5 dB can be noticed for 50% of the devices and a reduction up to a maximum of 4.7 dB. A reduction of up to 3 dB can be seen for $a_z(f)$ at high frequency. $a_z(f)$ is mostly unaffected by SCCR2a. However, an outlier device pair can be noticed, which increases RMSE with 0.7 dB at low and high frequency for $a_z(f)$. For $T(\lambda)$ a slight decrease is observed for 15% of the device pairs. The difference in the performance of the mapping function between SCCR1 and SCCR2a is minimal, which indicates that the SCCR1 model already handles the offset per car that is introduced via normalization very well for most of the device pairs.

In Figure 5, the distribution of the absolute RMSE by device pair and feature of SCCR2a is also shown. The measurement car, feature, and frequency are significant influencers on the RMSE. For $L_{eq}(250 \text{ Hz})$ and $L_{eq}(1000 \text{ Hz})$, 95% of the device pairs have an RMSE lower than 3.6 dB. The spread of RMSE peaks at high frequency. For $a_z(f)$, the outlier (maximum and minimum) is less prominent. At 1 Hz, 95% of the pairs are 7 dB or less. In contrast to $a_z(25 \text{ Hz})$, 95% of the pairs are 3.5 dB or less. Spread between RMSE of different device pairs is small for $a_z(f)$. For $T(\lambda)$, the maximum RMSE is up to 8 dB. However, 95% of the device pairs have an RMSE smaller than 4 dB for $\lambda \leq 1 \text{ m}$. At $\lambda = 5 \text{ m}$, 95% of the pairs have an RMSE smaller than 6 dB.

FIGURE 5 At the top row, the difference in root-mean-squared error (RMSE) by device pair and feature column between SCCR2a and SCCR1 is shown. At the bottom row, RMSE by device pair for SCCR2a. The lines show from bottom to top: minimum (dotted), 5%, 15%, 35%, 50% (thick black line), 65%, 85%, 95% percentiles and maximum (dotted).
Variations in network architecture for the encoder and decoder (Section 2.4) were analyzed for SCCR2a. In Figure 6, the training curve is shown. In comparison to SCCR2a-Dense, SCCR2a-Gated and SCCR2a-ReLU converge smoother and much faster. However, the final MSE on the test set is not significantly lower than for SCCR2a-Dense. Thus, these alternative architectures mainly introduce an opportunity for over-fitting the training data. Based on the training curve, it could also be expected that the improvement by introducing ReLU is more important than the actual introduction of gating in SCCR2a-Gated. Even though other neural network architectures could still be envisioned, the SCCR2a-Dense approach is therefore still simple and acceptable.

4.3 | Generalization to unseen device pairs

In Figure 4, it is seen that a lot of devices are not connected as they were not driving at the same location. To test the generalization capability of the SCCR2a-Dense model—that is the ability of the mapping $Y$ to translate measurements between devices that have never driven at the same place—the neural network is retrained with a reduced amount of training pairs. This results in a model SCCR2a-Dense-takeout. In the training set, for each node in the connectivity graph (a car), one randomly chosen connection is removed by deleting all examples connecting $(c_1, c_2)$. A new training dataset is obtained that is sampled from the pairs $E' \subset E$. The test dataset is left untouched.

To analyze the result, the relative change in MSE between model SCCR2a-Dense-takeout (Model 2) and SCCR2a-Dense (Model 1) is introduced in Equation 8:

$$\Delta MSE (A) = 1/|A| \sum_{(c_1, c_2) \in A} \frac{1}{MSE_1 (c_1, c_2)} \times (MSE_2 (c_1, c_2) - MSE_1 (c_1, c_2)) \tag{8}$$

$A$ is either $E'$ (seen device pairs) or $E \setminus E'$ (unseen device pairs). Figure 7 shows that the mean $\Delta MSE$ between both models is well below 1% for the device pairs that remain in the training set, while it is at most 5% for $L_{eq}$ and $T$ and at most 1% for $a_z(f)$ for device pairs that were removed from the training set.

4.4 | Generalization to unseen devices

By construction, SCCR2a can only process data from devices included in the training set. Thus, in this section, SCCR2b is explored—a model that is introduced in Section 2.3, which uses a latent representation of cars identified by their typical measurements.

To this end, a device classification model (Figure 2c) is first constructed. Its input, 100 random observations made by the same device, is fed to the set encoder, a fully connected layer with ReLU activation and 30 nodes, a latent layer with dimension $L_C$ without an activation, a fully connected layer with ReLU activation and 30 nodes, and finally a softmax car classification head. The following parameters have been decided in the set encoder: the number of processing steps (10), memory dimension (210), query vector dimension (210). Given the number of devices, $L_C$ is set to 2. The model with 568k parameters is trained by the Adam optimizer with a batch size equal to 10. The learning rate is $10^{-3}$ for the first five epochs and then fine-tuned to $5 \times 10^{-5}$ for the remaining epochs. The model takes around 1 h to train per epoch.

Five thousand sets of 100 observations are selected for each car from the training dataset, and 1000 sets of 100 observations for each car are selected from the test dataset. To test generalization to cars not present in the training set, three cars are picked at random and are removed from the training dataset. The classification model converges after 15 epochs and reaches 100% accuracy on the training data and an accuracy of 95% on the test dataset. In Figure 8, the latent space is visualized. Each dot in the graph corresponds to an identification of the car based on a distinct set of 100 measurements. As expected from the high accuracy of the classifier, dots cluster in clearly identifiable regions. Also, the 3 cars not included in the training set map to condensed clusters of dots, which shows that the model generalizes well for new unseen devices.
Next, the SCCR2b model (Figure 2b) is constructed. The SCCR2b neural network is equal to SCCR2a-Gated, with the exception that the one-hot device identifier is removed and replaced by the 2D latent representation. The car identifier is mapped to the mean value of the representations of the 1000 sets of 100 measurements (test set) in the latent space and entered in the main model. The same procedure is followed for unseen devices. SCCR2b is again trained to convergence (with 38 devices). Once more, mean $\Delta MSE$ is used to compare the two models SCCR2b and SCCR2a-Gated (Figure 9). For cars that were included in the training, both models perform equally well, which could be expected based on the uniqueness of the representation in the latent space. For cars not included in the construction of the model, mapping to and from measurements made by these cars seems rather difficult and $\Delta MSE$ reaches values of 20% to 40% depending on the feature.
4.5 Effect of calibration on texture features

In this section, the analysis of the fully calibrated data is presented. \( Y'' \), \( Y_x \), and \( \Delta Y_x \) are calculated with SCCR2a-Dense, using the metrics described in Section 2.8. For scalability reasons, 10 leader cars \( C_L \) are appointed as discussed in Section 2.5. The quality of features relevant for road mega-texture and waviness, either \( a_z \), \( a_z'' \), \( T \), and \( T'' \), is compared. Results are indicated in Figure 10.

Index \( w(T) \) is slightly smaller than \( w(a_z) \). This means that part of the variation in measurements at the same location is probably caused by driving speed. Calibration lowers \( w(M) \) significantly. Similar results are achieved for \( w(a_z'') \) and \( w(T'') \), which means that the advantage of using the texture feature instead of \( a_z \) disappears after the DAE transformation. However, \( a_z' \) is still dependent on \( T \) either way because \( T \) is an input feature of the DAE. For the same reason, index \( o(T) \) is lower than \( o(a_z) \). It can be also seen that the measurements vary between locations more...
for lower $\lambda$ or higher $f$. Calibration lowers $\alpha$ as well (most notable for low frequencies). For ratio $r$, the calibration algorithm has a significant increase from 2 for low to 10 for high frequencies. Moreover, ratio $r(T)$ corresponds to $r(a)_{\alpha}$; hence, no significant improvement is found here. To conclude $a''$ and $T''$ perform similarly, and the biggest performance gain on the ratio is due to calibration. Finally, the ratio is slightly lower than 1 for non-calibrated measurements, which means that the measurement uncertainty is higher than the sensitivity and the raw measurement is rather useless.

### 4.6 Validation with EC

The reference EC measured with the ARAN vehicle has been obtained for 2019, which unfortunately introduces some uncertainty concerning possible additional damage or aging by the time the opportunistic measurement campaign started. The following roads overlap with the opportunistic measurements: E40(Brussel-Oostende), E17(Kortrijk–Antwerpen), A17(Brugge–Kortrijk), B401(Gent), R4 (ring road around Gent), N9 (Maldegem–Gent), N43(some sections between Gent–Kortrijk), N60(Gent–Oudenaarde), N50(Brugge–Kortrijk). These are included in both directions on the right-most lane; hence, the total length is 784 km. Long, consecutive sections of the roads are appointed to either the test set or the train set to ensure generalization of the model to new data.

A different GAM model is fitted for every target EC with a selection of features calibrated with SCCR2a as inputs. These features have a high linear correlation with EC:

| $\Delta T$ (25 cm), $\Delta T$ (155 cm), $d_{\text{max}}''$ (32 Hz), $d_{\text{max}}''$ (2000 Hz), $\Delta a''$ (1 Hz), $\Delta a''$ (5 Hz), $\Delta a''$ (25 Hz), $\Delta L_{eq}$ (25 Hz), $\Delta L_{eq}$ (800 Hz), $\Delta L_{eq}$ (1000 Hz), $\Delta L_{eq}$ (1200 Hz) |

The supervised GAM regression results in a reconstructed EC, which is plotted for the test set against the ground truth in Figure 11.

Then, the thresholds given in Section 2.7 are applied to find roads that are in bad shape. The results of this binary classification as well as correlation coefficients on the continuous variable are displayed in Table 3. To identify roads that do not match expectations, $EC_{\lambda} = 40$ m has the best tradeoff between recall and precision performance; recall is nearly perfect. Next is $EC_{\lambda} = 10$ m, which still has decent precision and recall (respectively, 82% and 81%). $EC_{\lambda} = 2$ m lacks recall (a lot of false negatives); however, the precision is still high.

### 5 CONCLUSION AND FUTURE WORK

This paper showed that the opportunistic sensing using sound and vibration obtained from cars that are on the roads for purposes other than measuring the road quality can be used to assess the mega texture and, in particular, the evenness of the roads. However, all this cannot be achieved without a suitable calibration and denoising model. Hence, a model is introduced that scales robustly when the number of vehicles equipped with sound and vibration sensors grows. Compared to the previous work, this model self-calibrates even if all measurement vehicles do not have the access to the same road segments. Indeed, in the presented examples, only 327 out of 1681 possible car pairs were found in the connectivity graph. Moreover, it was shown that the removal of an additional set of 82 random pairs only slightly reduced the quality of mapping $Y'$ between the cars.

Several variants of the neural network architecture of the self-supervised calibration and confounder removal (SCCR) model have been tested while three of the models were discussed in detail. The differences between the models were small; however, it was shown that the network that includes the gating based on the car identifier converges slightly faster and results in a slightly lower MSE on the test set, compared to other models. On the other hand, the lower MSE on the train set reflects that the model overfits on the training data if more freedom is added. All this indicates that the two hypotheses—(a) two measurements under the same conditions at the same location within a reasonable time interval should give the same result, and (b) known dependencies such as the driving speed and temperature are independent of the local differences in road quality—have been extensively explored. The remaining error could indeed come from the fact that these hypotheses cannot be completely fulfilled. Possible reasons

### Table 3 Results of Evenness Coefficient (EC) Model Using Calibrated Features

| Train | Test |
|-------|------|
| $EC_{\lambda=2}$ m,l | 0.46 7.9 0.61 87% 54% 6.22 |
| $EC_{\lambda=2}$ m,r | 0.39 8.6 0.57 100% 65% 6.18 |
| $EC_{\lambda=10}$ m,l | 0.62 23.0 0.64 72% 82% 21.65 |
| $EC_{\lambda=10}$ m,r | 0.52 27.2 0.62 82% 81% 22.48 |
| $EC_{\lambda=40}$ m,l | 0.46 135.9 0.56 94% 99% 98.39 |
| $EC_{\lambda=40}$ m,r | 0.40 169.50 0.54 97% 99% 116.41 |

Notes: The subscripts l and r indicate EC ground truth measured at the left and right wheels, respectively. For precision and recall, the task enlists identifying the sections, which are in bad shape = (true).
could be that not all cars drive exactly on the same lateral location, for example, while avoiding potholes; the driving speed taken from the GPS signal fails near the crossings and in the city centers where vehicles frequently accelerate and decelerate; the temperature used for modeling differs from the actual pavement and tire temperature; and so forth.

In a scalable approach, the continuous introduction of new measurement vehicles poses a challenge. A possible solution was explored in the model SCCR2b where the additional sub-model was introduced to identify cars from their typical measurements using the latent representation as an input to the main model. Such identification of cars worked particularly well but the generalization of the model to unseen devices still introduced a significant uncertainty in the mapping $Y^*$ between cars. At the moment this publication was written, 41 devices had sufficient mileage to include them in the analysis of the model behavior. However, obtaining data from even more vehicles may additionally improve the generalizability of the presented model.

Finally, the result of the self-calibration and noise removal has been validated by adding a final layer to predict the ground truth introduced by the laser measurements of the EC. This final layer is constructed as a GAM with a selection of features as inputs. It was shown that the outcome of such layer correlates well with the ground truth on an independent test set: The results showed a good recall and precision on the task of classifying the roads according to the minimal requirements suggested by the road authority.

For future work, several different methods could be explored: to include the notion of time of the car dynamics (Jeong et al., 2020), to exploit the spectral nature of features with CNN’s, to construct models with fewer data (Pareira et al., 2020) or to unleash more powerful supervised learning algorithms (Ahmadlou & Adeli, 2017; Alam et al., 2020; Rafiei et al., 2017). As an alternative to the latent car identification (SCCR2b), local-to-global learning could handle a growing car fleet (Cheng et al., 2019).

ACKNOWLEDGMENTS
This research received funding from the Flemish Government under the “Onderzoeksprogramma Artificiele Intelligente (AI) Vlaanderen” programme and the Research Foundation Flanders (FWO-Vlaanderen) [G0D5215N]. The Flemish road administration (Agentschap Wegen en Verkeer) are thanked for the laser reference measurements. The authors would like to thank the reviewers for their insightful, in-depth survey of the paper, which has greatly improved the quality of the paper. With sadness, the authors report that Professor De Coensel has recently passed away in September 2021. Professor De Coensel is thanked for his important contribution to the research.

CONFLICT OF INTEREST
The authors declare employment by and share ownership of ASAsense BV, which commercializes sound and vibration-based road sensing.

ORCID
Wout Van Hauwermeiren https://orcid.org/0000-0001-9107-546X
Karol Filipan https://orcid.org/0000-0002-2966-5984
Dick Botteldooren https://orcid.org/0000-0002-7756-7238
Bert De Coensel https://orcid.org/0000-0003-1815-6436

REFERENCES
Ahmadlou, M., & Adeli, H. (2017). Enhanced probabilistic neural network with local decision circles: A robust classifier. *Integrated Computer-Aided Engineering, 17*(3), 197–210.
Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing with Applications, 32*, 8675–8690.
Alhassan, A., White, D. J., & De Brabanter, K. (2016). Wavelet filter design for pavement roughness analysis. *Computer-Aided Civil and Infrastructure Engineering, 31*, 907–920.
ASTM D6433–20. (2020). *Standard practice for roads and parking lots pavement condition index surveys*. ASTM International.
ASTM EI926 - 08 (2015). *Standard practice for computing international roughness index of roads from longitudinal profile measurements*. ASTM International.
Baldini, G., Giuliani, R., & Geib, F. (2020). On the application of time frequency convolutional neural networks to road anomalies’ identification with accelerometers and gyroscopes. *Sensors, 20*(22), 6425.
Balzano, L., & Nowak, R. (2007). Blind calibration of sensor networks. *Proceedings of the 6th International Conference on Information Processing in Sensor Networks, IPSN ’07*, 79–88, Cambridge, MA.
Bang, S., Park, S., Kim, H., & Kim, H. (2019). Encoder–decoder network for pixel-level road crack detection in black-box images. *Computer-Aided Civil and Infrastructure Engineering, 34*, 713–727.
Bello-Salau, H., Aibinu, A. M., Onumanyi, A. J., Onwuka, E. N., Dukiya, J. J., & Ohize, H. (2018). New road anomaly detection and characterization algorithm for autonomous vehicles. *Applied Computing and Informatics, 16*, 223–239. https://doi.org/10.1016/j.aci.2018.05.002
BRRC-b. (2020). *Instrumenten voor wegbeheerders, APL: meting van de langsvalsklacht van wegen; nl, Instruments for road administrators, APL: measurement of longitudinal unevenness of roads; en*. Technical report, Opzoekingscentrum voor de Wegenbouw (OCW). https://brrc.be/nl/expertise/expertiseoverzicht/apl-meting/langsvalsklacht-wegen
BRRC-a. (2020). *Handleiding voor slemlagen* (Technical manual). Belgian Road Research Institute.
Cheng, H., Lian, D., Deng, B., Gao, S., Tan, T., & Geng, Y. (2019). Local to global learning: Gradually adding classes for training deep neu-
rational networks. *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, Long Beach, CA.

El-Wakeel, A. S., Li, J., Noureldin, A., Hassaneln, H. S., & Zorba, N. (2018). Towards a practical crowdsensing system for road surface conditions monitoring. *IEEE Internet of Things Journal*, 5, 4672–4685.

EN 13036-5:2019. (2019). Road and airfield surface characteristics—Test methods—Part 5: Determination of longitudinal unevenness indices. European Committee for Standardization. https://standards.cenelec.eu/

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. http://www.deeplearningbook.org

Haul, H. D. (2005). Road defect identification and condition assessment using measured truck response. [*Master's thesis*]. Department of Mechanical and Aeronautical Engineering, University of Pretoria.

Heydinger, G. J., Bixel, R. A., Garrott, W. R., Pyne, M., Howe, J. G., & Guenther, D. A. (1999). Measured vehicle inertial parameters-NHTSA’s data through November 1998. *SAE Transactions*, 108, 2462–2485.

Hsieh, Y.-A., Yang, Z., & Tsai, Y.-C. J. (2020). Convolutional neural network for automated classification of jointed plain concrete pavement conditions. *Computer-Aided Civil and Infrastructure Engineering*, 36(11), 1382–1397. https://doi.org/10.1111/mice.12640

ISO 13473-1:2019. (2019). Pavement surface texture—Part 1: Determination of mean profile depth. International Organization for Standardization.

ISO 8608:2016. (2016). Mechanical vibration—Road surface profiles—Reporting of measured data. International Organization for Standardization.

Jeong, J., Jo, H., & Ditzler, G. (2020). Convolutional neural networks for pavement roughness assessment using calibration-free vehicle dynamics. *Computer-Aided Civil and Infrastructure Engineering*, 35, 1209–1229.

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Kim, G., Lee, S. Y., Oh, J.-S., & Lee, S. (2021). Deep learning based estimation of the unknown road profile and state variables for the vehicle suspension system. *IEEE Access*, 9, 13878–13890.

Menegazzo, J., & von Wangenheim, A. (2020). Vehicular perception based on inertial sensing: A structured mapping of approaches and methods. *SN Computer Science*, 1, 255. https://doi.org/10.1007/s42979-020-00275-z

Mirtabar, Z., Golroo, A., Mahmoudzadeh, A., & Barazandeh, F. (2020). Development of a crowdsourcing-based system for computing the international roughness index. *International Journal of Pavement Engineering*, 23(2), 489–498.

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

Pallas, M.-A., Tufano, A.-R., Chiello, O. (2020). Separation of rail and wheel roughness from on-board vibroacoustic measurements. *Proceedings of Forum Acusticum*, 2020, Lyon, France.

Park, Y. S., Jeon, J. H., & Kang, Y. J. (2020). ISO 8608-based pavement roughness classification with artificial neural networks using suspension vibration measurements. *Proceedings of Inter.Noise 2020*, Seoul, South Korea (pp. 5651–5661).

Pasindu, H. R., Sandamal, R. M. K., & Perera, M. Y. I. (2020). A framework for network level pavement maintenance planning for low volume roads. In C. Raab (Ed.) *Proceedings of the 9th International Conference on Maintenance and Rehabilitation of Pavements* (pp. 103–113). Springer International Publishing.

Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J., & Adeli, H. (2020). *FEMA: A finite element machine for fast learning*. *Neural Computing and Applications*, 32(10), 6393–6404.

Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 28(12), 3074–3083.

Sawyers, M. W. & Karamihases, S. M. (1998). *The little book of profiling: basic information about measuring and interpreting road profiles*. Technical report. https://deepblue.lib.umich.edu/handle/2027.42/21605

Servén, D., Brummitt, C., & Abedi, H. (2018). *dswha/pyGAM*: v0.8.0. https://doi.org/10.5281/zenodo.1476122

Sun, Y., Hu, M., Zhou, W., & Xu, W. (2020). Multiobjective optimization for pavement network maintenance and rehabilitation programming: A case study in Shanghai, China. *Mathematical Problems in Engineering*, 2020, 3109156.

Belgian Government - FOD Mobiliteit (2010). *Evolutie van het wegenet*. https://mobilit.belgium.be/nl/mobiliteit/mobiliteit_cijfers/evolutie_wegenet

Trog, J., Botteldooren, D., De Coensel, B., Martens, L., Joseph, W., & Piets, D. (2020). Map matching and lane detection based on Markovian behavior, GIS, and IMU data. *IEEE Transactions on Intelligent Transportation Systems*. Advance online publication. https://doi.org/10.1109/TITS.2020.3031080

Tyan, F., Hong, Y.-F., Shun, R., Tu, H., & Jeng, W. (2009). Generation of random road profiles. *Journal of Advanced Engineering*, 4(2), 1373–1378.

Valikhani, A., Jaberi Jahromi, A., Pouyanfar, S., Mantawy, I. M., & Azizinamini, A. (2021). Machine learning and image processing approaches for estimating concrete surface roughness using basic cameras. *Computer-Aided Civil and Infrastructure Engineering*, 36, 213–226.

Van Hauwermeiren, W., David, J., Dekoninck, L., De Pessemier, T., Joseph, W., Botteldooren, D., Martens, L., Filipan, K., & De Coensel, B. (2019). Assessing road pavement quality based on opportunistic in-car sound and vibration monitoring. *Proceedings of the 26th International Congress on Sound and Vibration (ICSV 2019)*, Montreal, Canada.

Van Hauwermeiren, W., Filipan, K., Botteldooren, D., & De Coensel, B. (2021). Opportunistic monitoring of pavements for noise labeling and mitigation with machine learning. *Transportation Research Part D: Transport and Environment*, 90, 102636.

Varona, B., Monteserin, A., & Teysyere, A. (2020). A deep learning approach to automatic road surface monitoring and pothole detection. *Personal and Ubiquitous Computing*, 24, 519–534. https://doi.org/10.1007/s00779-019-01234-z

Vaswani, A., Shazeer, N., Permar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you Need. *Proceedings of Advances in Neural Information Processing Systems* 30, 2017, Long Beach, CA.
APPENDIX
Algorithm 1. Training data selection. Run this procedure once for training segments and once for test segments

Input:
S: selected locations to be included in the dataset
D: amount of examples per device pair
\( D_{min} \): threshold, minimum available samples per devices
\( b = 20 \)
C:= cars
T:= time periods (3 months sliding window with stride of 1 month)
\( n(c,s,t) \): number of passages of a car \( c \) on a certain segment \( s \) and during a time period \( t \) (eg 3 months)
T x S x C x C: Carthesian product

Helper functions:

\( \text{scaling}(x) := \begin{cases} \sqrt{b(x/b-1)}, & x \leq b \\ \frac{x}{x+b}, & x > b \end{cases} \)

\( \text{max_pairs}(c1, c2, s, t) := \begin{cases} \text{scaling}(\text{max}(n(c1,s,t)-1,0)) & c1 = c2 \\ \text{scaling}(n(c1,s,t))(n(c2,s,t)) & c1 \neq c2 \end{cases} \)

\( \text{MP}(c1, c2) := \sum_{s} \sum_{t} \text{max_pairs}(c1, c2, s, t) \)

\( p(c1, c2) := \begin{cases} \frac{\text{MP}(c1, c2)}{c1(c1, c2)} & \text{if } C(c1, c2) \geq D_{min} \\ 0 & \text{if } C(c1, c2) < D_{min} \end{cases} \)

random_sample(s, c, t): pick a random sample with equal probability out of segment \( s \), for device \( c \) and within time period \( t \)

Algorithm:

\( X = [] \)
\( Y = [] \)

for \( t, s, c1, c2 \in T \times S \times C \times C \):

\# desired amount of pairs to pick at a certain segment
\( nd = p(c1, c2) \times \text{max_pairs}(c1, c2, s, t) \)

for \( i \in [1 \ldots \text{floor}(nd)] \):

\( x = \text{random_sample}(s, c1, t) \)
\( y = \text{random_sample}(s, c2, t) \)

if \( x \neq y \):

\( X[j] = x; Y[j] = y; j := 1; \)

if \( \text{random()} \leq \text{decimal}(nd) \):

\( x = \text{random_sample}(s, c1, t) \)
\( y = \text{random_sample}(s, c2, t) \)

if \( x \neq y \):

\( X[j] = x; Y[j] = y; j := 1; \)

Output:

\( X, Y: \text{the balanced dataset} \)