Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8849734, 12 pages
https://doi.org/10.1155/2020/8849734

Research Article

Train Type Identification at S&C

Martina Kratochvílová,[1] **Jan Podroužek**,[1] **Jiří Apeltauer**,[2] **Ivan Vukušič**,[3,4] **and Otto Plášek**[3]

[1] *Institute of Computer Aided Engineering and Computer Science, Faculty of Civil Engineering, Brno University of Technology, Veveří 331/95, 602 00 Brno, Czech Republic*
[2] *Institute of Road Structures, Faculty of Civil Engineering, Brno University of Technology, Brno, Czech Republic*
[3] *Institute of Railway Structures and Constructions, Faculty of Civil Engineering, Brno University of Technology, Brno, Czech Republic*
[4] *Výzkumný Ústav Železniční, a.s. (VUZ), Novodvorská 1698, 142 01 Prague, Czech Republic*

Correspondence should be addressed to Martina Kratochvílová; kratochvilova.m@fce.vutbr.cz

Received 28 August 2020; Revised 19 October 2020; Accepted 6 November 2020; Published 24 November 2020

Academic Editor: Petr Dolezel

The presented paper concerns the development of condition monitoring system for railroad switches and crossings that utilizes vibration data. Successful utilization of such system requires a robust and efficient train type identification. Given the complex and unique dynamical response of any vehicle track interaction, the machine learning was chosen as a suitable tool. For design and validation of the system, real on-site acceleration data were used. The resulting theoretical and practical challenges are discussed.

1. Introduction

A key and irreplaceable part of every railway track is its switches and crossings (S&C). In terms of dynamic effects, these are some of the most loaded track sections. They not only interrupt runway continuity but also see a change in track stiffness. S&C represent only a small part of a railway network in terms of the length of track; however, their maintenance (which includes special rail structures such as road crossings), relative to conventional tracks, can involve high maintenance costs [1–3]. The primary reason for this is the complex force effect that enables the train to pass through the S&C section; another factor is the requirement to maintain the upkeep of the many components that make up the S&C. As well as the significant direct costs, the maintenance of these sections generates indirect costs (due to delayed trains because of maintenance or slower travel, alternative train routes, and even alternative forms of transportation). Therefore, it is essential that any maintenance of such sections should be planned carefully. However, currently, there is no reliable device which can tell us when it is time for maintenance [4]. Moreover, it is a well-accepted fact that structures respond in a very uncertain manner to probabilistically different motion events while there is very limited a priori knowledge on the structural behaviour [5].

For the above reasons, condition monitoring of railways (not only S&C) is a very current topic. In recent years, various sensors and methodologies for measuring and evaluating results have been developed [6–13].

According to [14], machine learning (ML) methods in S&C are often used for condition monitoring and evaluation in data-based fault detection and diagnosis systems. In these cases, they help in large data to search for features that match different failure mechanisms.

This paper is focused on the first part of the self-diagnostic system for railway switches and crossings (S&C)—Train Identification System (TIS). A similar system for predictive maintenance for rail switches is, for example, Konux [15] or ESAH-M [16].

TIS is based on real on-site data from the acceleration sensor and in the future is assumed use of embed vibration acceleration sensors. Accelerometers provide various benefits including the following temperature stability (over a wide range of temperatures), wide frequency response, linearity, adaptability, and ruggedness. As such, they are suitable for fully operational online measurement in the long term. Different dynamic effects can be observed for each train type [17, 18]. To obtain an accurate comparison of these effects, it is vital that the same train types are compared at the same passing speeds. A precise comparison can help in the detection of faults and/or deterioration at the very early stages. The main benefits of this approach are use of predictive maintenance which can reduce costs [19] and better planning of the regular maintenance and decision support for infrastructure manager about maintenance activity (such as tamping, component replacement, and surface build-up welding).

This research was part of an initiative whose aim is to investigate, develop, validate, and initially integrate radically new concepts for switches and crossings that have the potential to lead to increases in capacity, reliability, and safety while reducing investment and operating costs.

The first part of the article is dedicated to the description of measurement of the data, selection of datasets, their analysis, and building of a vector for machine learning. The second part deals with the application of support vector machine and validation of the results.

2. Dataset

2.1. Measured Data. The used data were collected during several measurement campaigns which took place in the years 2013 and 2014. The measurements were made primarily on two locations: Choceň and Ústí nad Orlicí and on two S&C per each location. The accelerometers were mostly placed around the crossing because of the maximal dynamical effects on rails and bearers during the train passage. The placement of the sensors is shown in Figures 1 and 2.

All data were acquired with measuring system Dewetron DEWE 2502 and acceleration sensors triaxial piezoelectric Brüel &Kjær 4524 B001 (for rail) and piezoelectric Brüel & Kjær 4507 B004 (for bearer). Sampling frequency was set on 10 kHz, high-pass filter frequency 3 Hz, and low-pass filter frequency on 1000 Hz [20].

The train speed was measured by radar speed gun Bushnell.

Acceleration is measured at several points along the crossing. The observed magnitude is chosen as the vertical acceleration of the bearer under the crossing nose, as this is the point at which the greatest dynamic effects on bearers occur. Undoubtedly, any damage to the trackbed or the crossing would influence the frequency response. Figure 3 shows an example of an acceleration plot, which was observed during the passing of a train.

2.2. Measurement Selection. The full dataset consists of over 100 complex measurements (in addition to the acceleration, which was measured at several S&C locations, train speed and rail displacements were also measured), taken from trains passing through crossings at a number of stations. However, for building successful classifier, it is required to have data that were obtained under the same or very similar conditions. In Figure 4 are shown differences in vectors obtained from Ústí nad Orlicí and Choceň. The individual columns (i.e., the corresponding scalars of the individual vectors) were normalized and these normed scalars were assigned a colour shade on a scale between yellow and orange based on the value. The locations have different types of bearers, and therefore, the acceleration signals are incomparable. It is easy to see that first two and second two rows are from distinct locations. Due to the higher number of measurements, the data from Choceň were chosen. Though there are measured signals from two S&C from this location, it was not possible to use them for training of one classifier as each one has different dynamical behaviour due to the distinct conditions of stiffness of its support. Because of all these restrictions, there left very little data suitable for training and testing artificial intelligence (AI). Another complication with data comparability was the renovation of the common crossing that was done between the measurement campaigns and so the latter passages were measured under other conditions. Because of the lack of the training data, it was decided to keep these passages. At the same time, this allowed to verify the robustness of the classifier for this kind of S&C reparation.

2.3. Train Details. The available dataset was able to meet the requirements mentioned for only four trains. However, the number of measurements was still sufficient to build the minimum number of data subsets for training and testing. The mechanical properties of the trains are given in Table 1. The trains are shown in Figure 5.

2.3.1. Locomotive Classes 151, 362, and 380. These locomotives are very similar, in terms of both geometry and design. They were made by Czech industrial conglomerate Škoda Works. All the locomotives are electrical; however, class 151 can be powered only by direct current (3 kV) while both 362 and 380 are adapted for other standardised voltages and current (362 is equipped with double system 3 kV DC/25 kV 50 Hz and 380 is equipped with even triple system 3 kV DC/25 kV 50 Hz/15 kV 16,7 Hz). The maximal speed is 160 km/h for type 151, 140 km/h for 362, and 200 km/h for 380. Locomotives 151 and 380 have the same fixed wheelbase and pivot spacing.

2.3.2. Leo Express. Leo Express train is Stadler Flirt IC five-car electric multiple unit. That means the train signal should always have 12 peaks. The major difference between LE and previously mentioned trains is system of chassis. The LE has two powered bogies (at both ends of the train) and 4 Jacobs bogies [21] between the carriages. These characteristics allow well distinguishing the Leo Express signal from other train types. The maximal travel speed is 160 km/h.

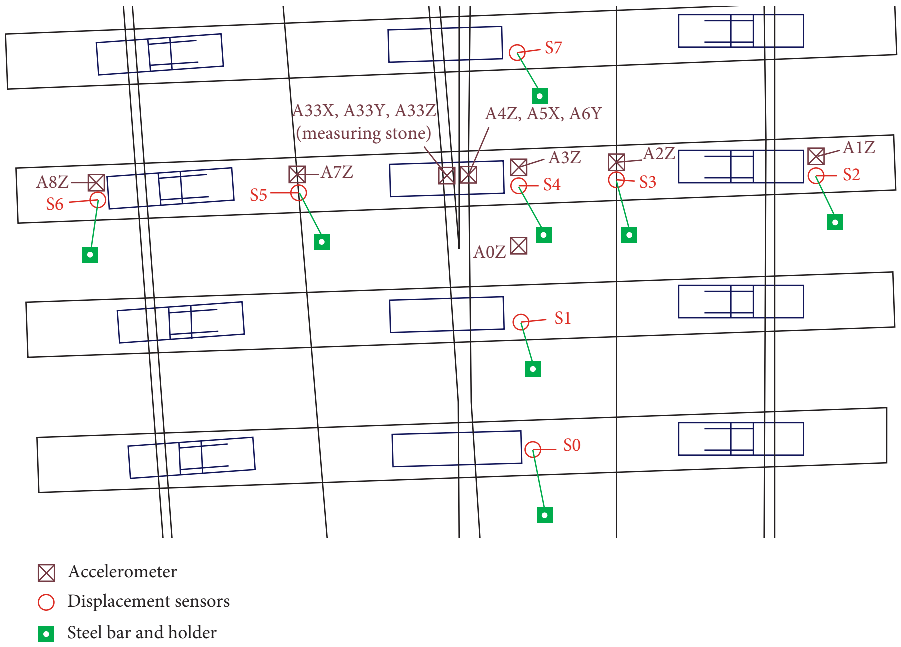

Figure 1: Measurement methodology used for data acquisition from the crossing part of the S&C.

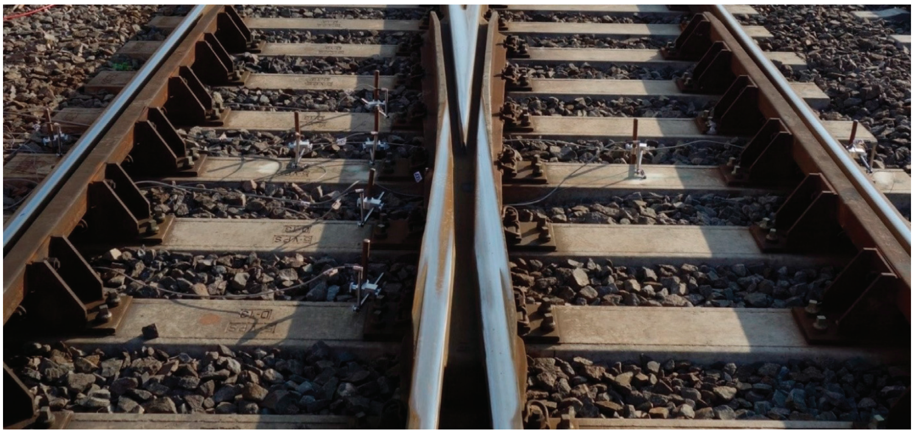

Figure 2: On-site installation of the sensors according to measurement methodology above.

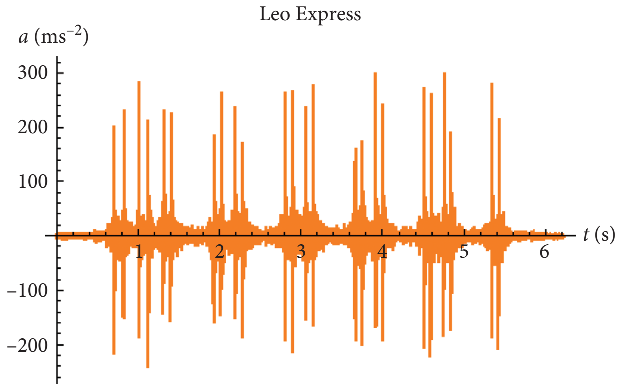

Figure 3: Acceleration plot of the bearer under the crossing nose.

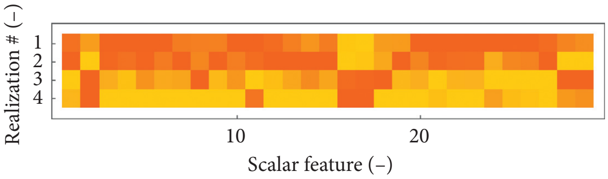

Figure 4: Comparison of data vectors from Ústí nad Orlicí (rows 1 and 2) and Choceň (rows 3 and 4).

3. Data Analysis

There are 3 considered groups of methods: (i) complex time-frequency methods, (ii) methods based on statistical processing, and (iii) combination of the two previously

Table 1: Mechanical properties of the trains.

Locomotive class	151	362	380	Leo
Distance between pivots (m)	8.3	8.3	8.7	16.0
Axle spacing (m)	3.2	3.2	2.5	2.7
Max. axle loading (t)	20.5	21.75	21.5	—
Weight (t)	82.0	87.0	86.0	150.0[1]

[1] Total weight of the whole five-car unit.

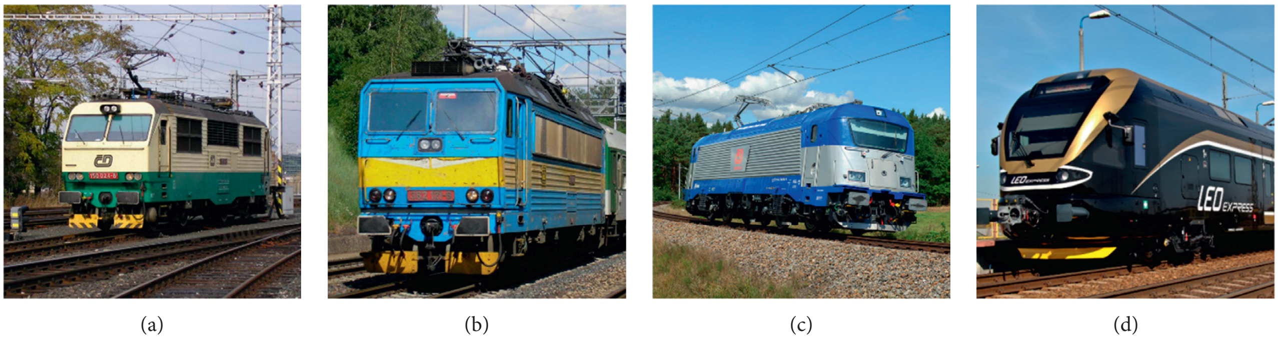
(a) (b) (c) (d)

Figure 5: Train types: (a) 151, (b) 362, (c) 380, and (d) Leo Express.

mentioned. The first group analyses signal simultaneously in both time and frequency domains. There are several time-frequency distribution functions, such as wavelet transform (WT), Wigner–Ville transform (WVT), and short-time Fourier transform (STFT). With these methods, it is possible to conduct a sufficiently detailed analysis of the structure's frequency response to reveal minor differences in the individual signals that can suggest that there are vehicle and track faults. However, the major disadvantage of these methods is their significant requirement for data performance and, hence, for computing resources. This is problematic when attempting to ensure long-term in situ measurements for multiple S&C. The use of expensive sensors is also necessary to ensure the high quality of the signals; however, this may not align with other deployment objectives.

The second group of analysis methods can be used as an alternative, and these are based on statistical processing. For example, it is possible, with these methods, to evaluate the maximum amplitudes, as well as their count, standard deviation, and long- and short-term variance. This group of methods is, in essence, the opposite of the time-frequency methods because they have little sensitivity to imperfect input signals, their computational difficulty is negligible (in comparison with the first group of methods), and the device built as a result can be inexpensive. However, the main disadvantage of the second group is that there is limited information in the frequency domain, meaning that the detection of any defects might be too late to be of use. Nonetheless, the time domain of the signal provides very accurate information.

The methods in the third group are a combination of the two approaches mentioned previously, enabling the time domain of the signal to be analysed using statistical methods. In identified areas of interest (for example, maximum amplitude axles), a simple frequency analysis can be conducted using the selected signal subsection's frequency spectrum and its statistical properties. This method is advantageous for our research because it is economical on computer performance while being able to adequately describe the signal.

3.1. Signal Evaluation in the Time Domain Using Statistical Methods. The use of statistical methods was inspired by previous research [22] that focused on train detection and classification. This innovative method evaluates the accelerometer record as a windowed variance of acceleration, based on 12–20 records at a sampling rate of 100 Hz, a sensitivity of ±4 g, and a resolution of 10 bits. Despite the minimalistic resolution (as well as the minimal power and hardware requirements), the system can achieve very accurate results as well as detect and classify trains with over 95% precision. It has a battery capacity of 180 mAh (units of percent of conventional smartphone battery capacity), enabling the device to take measurements for approximately two weeks. An SD card is used to store the results.

As a truly economical system, this can easily be scaled and expanded to other variables (as demonstrated in Figure 6), including maximum number, standard deviation, and absolute and local maximum, requiring minimal power and hardware. To identify short signal sequences, time-based input analysis can be used when a more detailed analysis is conducted in the frequency domain (as shown in Figure 7).

3.2. Signal Evaluation in the Frequency Domain Using Statistical Methods. Evaluation is performed in the frequency domain with the Seewave package [23], using R language to process the model example. Practical deployment would require a lower level of programming language, probably at

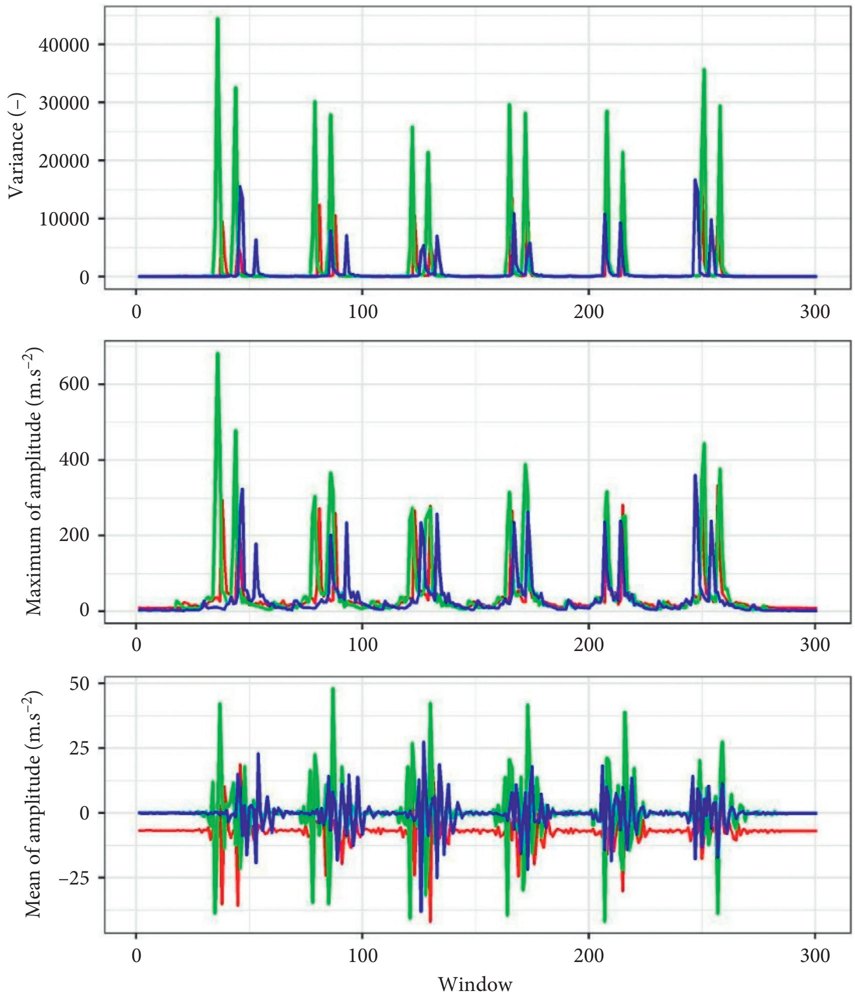

Figure 6: LEO Express train signal windowed variance (three passages). Window size: number of samples/300. Top: windowed variance value. Middle: windowed maximum. Bottom: windowed mean.

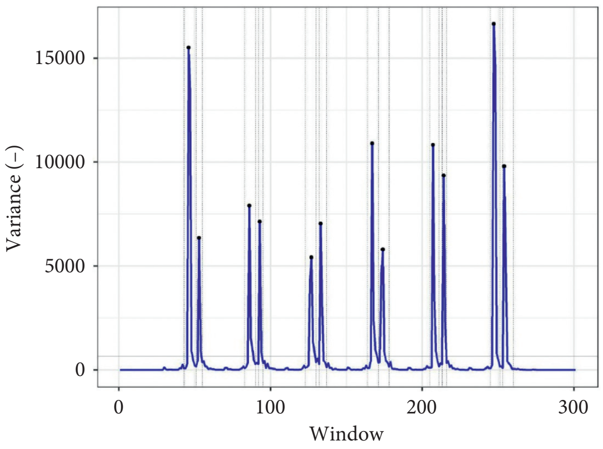

Figure 7: Peak detection of windowed variance. Window size: number of samples/300. Defines the area in the time domain; the analysis is then conducted in the frequency domain.

the firmware level. However, the frequency analysis is very complex, and therefore, only a limited sample of data is performed. Statistical methods are used for sample selection in the time domain. As a result, the processing is computationally efficient, particularly regarding the amount of memory used. Described by a relatively short vector of statistical properties, the spectrum transforms into a discrete probability density (as illustrated in Figure 8).

The analysis is supplemented further by the maximum and minimum frequencies at three density intervals (0.0001–0.00015, 0.00015–0.0002, and 0.0002–0.0004). Combining the scalar features of the statistical properties in the frequency and time domains enables a vector to be obtained that represents the signal in the time-frequency domain but with minimum resources in comparison with traditional methods such as WT or STFT. However, it should be noted that not all vector values are relevant.

3.3. Machine Learning Methods: Building a Vector. A wide variety of data and formats can be used as inputs for ML. A high-resolution accelerometer signal (such as 10 kHz) as an input is likely to be the simplest option. However, this would require a particularly powerful computing subsystem with substantial memory, which would render the method unsuitable for use in situ or on larger scales. In addition, it is

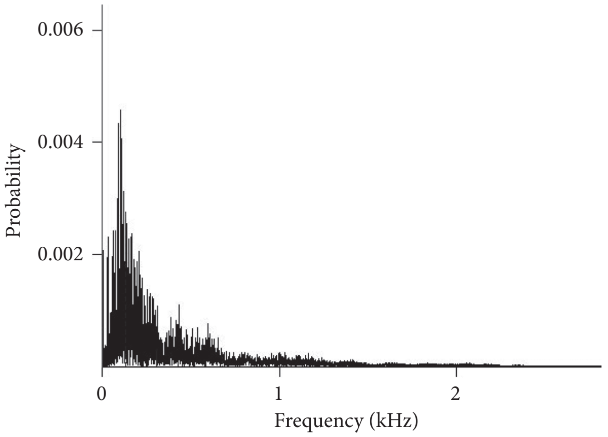

Figure 8: Discrete probability density plots of the Leo Express.

not guaranteed that such a procedure will lead to the best results. Therefore, the selection of the descriptive features is an important step in the ML-based identification process, along with the creation of a sequence of *n* scalar features (or representations) by reducing the recorded acceleration time history. The representations will include event duration, total amount of vibration caused by the train, number of peaks extracted from the windowed variance, average distance between peaks, maximum peak value, average peak amplitude, average peak area under the curve, total area under the curve, and variance of peak distances. The computational power requirements are reduced by several orders of magnitude by using the combined time-frequency characteristics vector defined in the previous section. However, it is highly likely that some of the features will be random or similar for each individual train, and including such features in the calculation could easily confuse the machine (e.g., SVM or neural network), leading to incorrect results.

The initial set of 27 scalar features contains number of peaks (number of axles), their minimum and maximum, standard deviation, and total sum. Furthermore, the mean of the signal, standard deviation, median, standard error of the mean, 25% and 75% quantile, interquartile range, centroid, skewness, kurtosis, spectral flatness measure, and minimum and maximum frequencies for a given interval of discrete probability density. This vector was reduced to 5 using an iterative optimisation process, whereby accuracy was maximised by the minimisation of training time, evaluation time, and classifier memory and loss. The initial set of 27 scalar features contains number of peaks (number of axles), their minimum and maximum, standard deviation, and total sum. Furthermore, the mean of the signal, standard deviation, median, standard error of the mean, 25% and 75% quantile, interquartile range, centroid, skewness, kurtosis, spectral flatness measure, and minimum and maximum frequencies for a given interval of discrete probability density. The use of the whole vector was considered; however, due to the low number of data and the large number of possible parameters, this is an overdetermined problem, and therefore, according to the authors, it did not make sense to do a detailed sensitivity analysis. Figure 9 shows visualization of velocity and scalar features which were selected for description of the individual train passages. The data are sorted by train type. It can be seen that values of some scalar features of some classes are correlated with the train type, and hence, they are clustering whereas other classes have values widely scattered. For this reason, there is a need to have more than one scalar features to correctly classify the signal. However, as was said earlier, it is not advantageous to use all 27 scalar features not only because of high computational demands but also because of the well-known phenomenon of curse of dimensionality [24]. Velocity was not selected into the vector because it is secondarily included in the other characteristics and for some S&C may be strongly influenced by the position in the track and not by the train type.

The following scalar features were chosen to describe the train passage:

(i) n_{peaks}: number of peaks detected during windowed variance. The R language findpeaks function was used for the detection. The number represents the number of axles on the train.

(ii) $\text{peaks}_{\text{sum}}$: sum of maximum values of npeaks detected. To a certain extent, this expresses the absolute amount of dynamic energy that is transmitted to the sleeper

(iii) sem: the random sampling process is described using the standard error of the mean. The variation in measurements is described using the standard deviation of the sample data. The sem is a probabilistic statement that describes how the sample size, considering the central limit theorem, will provide a better boundary on estimates of the population mean.

(iv) IQR: the interquartile range, which is also known as the midspread or the middle 50% (or, technically, H-spread), is a measure of statistical dispersion, which is equal to the difference between the upper and lower quartiles, or between the 75th and 25th percentiles. The IQR value represents the bandwidth of energy transferred to the sleeper.

(v) prec: the spectrum's frequency precision.

4. Machine Learning-Based Analysis

The aim of this study is to confirm the hypothesis regarding the possibility of using recorded acceleration data to identify specific train types at rail S&C. Utilisation of ML methods [25] seems appropriate due to the unique and complex dynamic interactions involved in the process, including those involving the vehicle itself and the wheel, as well as railway S&C components and ballast, and also the recorded signal's stochastic components. A further consideration is that ML might be able to identify not only a specific train type but also any possible damage to the wheel surface and parts of the S&C [26].

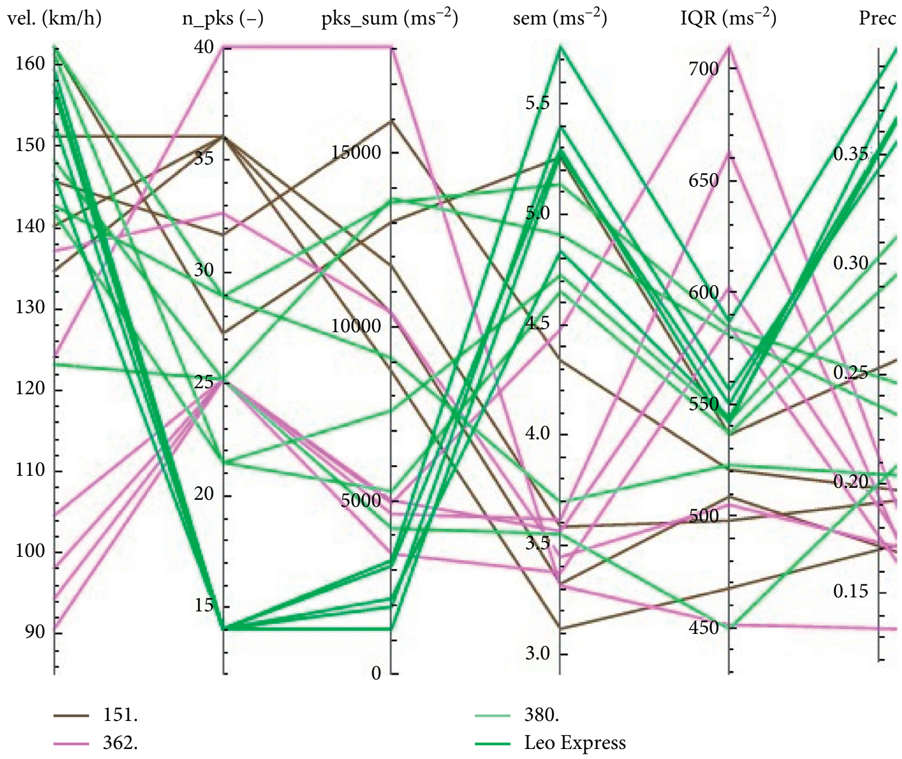

Figure 9: Visualization of clustering of scalar features for individual train passages sorted by train type.

An in-depth literature review showed that the use of measured acoustic or acceleration signals with ML to identify train type was performed successfully on a segment of plain-line railway [22]. However, no record was found of the successful application of ML, genetic algorithms, or pattern recognition [27] for train type identification at S&C.

4.1. Comparison of ML Methods. Currently, there are many machine learning methods that differ in the structure and complexity of the algorithm and the suitability for use with different types and sizes of input data. Based on recommendations derived from the literature review, as well as initial investigations using the available ML methods at Mathematica [28], the support vector machine (SVM) was identified as an optimal classifier. The following methods were considered:

(i) A decision tree [29] is a structure designed as a flowchart. Internal nodes represent “tests” for particular features; branches represent the outcomes of the tests; and the leaves represent classes or value distributions.

(ii) Gradient boosting [30] is an ML technique used for regression and classification problems. It produces an ensemble of trees that represent a prediction model. The trees are trained in sequence with the aim of compensating for the weaknesses of previous trees.

(iii) Logistic regression [31] uses a linear combination of numerical features to model the log probabilities of each class. However, its biggest disadvantage for our task is strong sensitivity for outliners.

(iv) In a Markov model [32], each class has a computed *n*-gram language model during training. During testing, each class’s probability is computed according to Bayes’ theorem.

(v) A naive Bayes [33] uses an assumed probabilistic independence of features. This method is convenient for large datasets with high dimensionality because it can identify the most significant features.

(vi) Nearest neighbours [34] use instance-based learning. It is easy to implement and works well for multiclass problem, but as the datasets grow, speed and efficiency of the algorithm decline fast. Another disadvantage is sensitivity for outliners and problems associated with curse of dimensionality.

(vii) The random forest [35] uses ensemble learning for classification and regression. It operates by constructing a number of decision trees. The prediction offered by the forest is obtained using the most common class or the mean value of the tree predictions. The training set is divided such that each decision tree is trained on a random subset of features. This algorithm is easy to train because there are not many options for tuning. When there are large input datasets, random forest gives robust model.

(viii) A neural network (NN) [36] is made up of stacked layers. Each layer performs a simple computation.

Starting from the input layer to the output layer, information is processed one layer at a time. The neural network is trained to minimise the training set's loss function using gradient descent and naturally learns nonlinear decision boundaries; however, it often converges to local minimums and can start to consider noise as a part of pattern and therefore overfit the classifier. The NN is parametric; this means that its size is constant with growing input datasets. There are many setting possibilities and it requires experience to set up the algorithm correctly. For this reason, the NN is not advantageous for TIC, which should be operated by engineers not by scientists.

(ix) Unlike neural network, the support vector machine [37] can produce reliable results even with small input datasets. Moreover, it is not sensitive to outliners. The principle is to find an optimal hyperplane dividing areas of different classes. The word "plane" can be somewhat misleading because it does not always have to be a flat plane (or line in 2D). The SVM is linear in its natural form, but it is possible to use other kernel functions that allow to operate in multidimensional space without calculating data coordinates. This can greatly save computing time. In this classifier was used radial basis function kernel. Another distinction is that SVM is nonparametric, and therefore, its complexity increases with the number of training samples. This means that SVM may be beneficial for this research, where is only small input datasets, but in actual implementation with multiple train type classes with higher number of passages, the calculation may take too long.

In this research, machine learning and its postprocess were performed in Wolfram Mathematica 11.1 [28]. The same analysis with the same inputs was also run in version 11.2 but with worse results. Even the choice of SVM as a best method was not validated in the newer version and gave better results for neural network. This may be caused by distinct setup of the embedded algorithm in both versions.

Comparison of ML methods shows the accuracy, training times, and required computation memory for some of the previously mentioned methods (Table 2). It can be seen that SVM gives the highest accuracy, but training takes twice as long as the second slowest method and even nearly 50 times longer than the fastest. However, it should be noted that the nearest neighbour method is the fastest because it does not need any training time—samples are sorted according to the class of their nearest neighbour (or k-neighbours).

4.2. Support Vector Machines.

In terms of implementation, SVMs are regarded as binary classifiers [25]. Features are extracted from the examples using a kernel function. During training, the classifier locates the maximum-margin hyperplane that separates the classes. Then, the problem of multiclass classification is reduced to a set of problems of binary classification (using a strategy of one-versus-one or one-versus-all). The LibSVM framework in C/C++ is used in the implementation.

Although classification using SVM can be controlled in a number of ways [28], such as gamma scaling parameter, kernel type, polynomial degree, and multiclass strategy, the training dataset is characterised reasonably well by the automatic settings. However, the training dataset is somewhat limited in terms of repeated identical observations (i.e., the same train on the same switch at a similar speed), which means that a detailed analysis of the effects of any particular setting is difficult.

Full validation of the classifier is impossible due to the limited number of comparable train passages. In the smallest classes, it is only possible to use one train passage for validation, whereas it is possible to use the remaining four comparable train passages for training. This is the case for all combinations. In total, 19 train passages are used, with the recorded acceleration time history being reduced to 5 scalar features.

4.3. Building of Train and Test Sets.

Due the low number of comparable train passage in the classes, the reliability of the classifier highly depends not only on the selection of the scalar features, but also on the choice of the vectors (passages) for the training set. To avoid cherry-picking and decrease possibility of incorrect results due to the inappropriate selection of data for training and testing, the bootstrap analysis was performed. Bootstrapping is a compute-intensive method for statistical data analysis [38]. The train passages for the training subset were chosen randomly for each class and the spare ones were used for testing. That means, as the smallest class has only 5 comparable train passages, the training set has 4 vectors per class and one vector for testing. According to [39], the imbalance in the size of the classes can significantly influence the results. Therefore, all classes for training have the same size of 4 passages. Figure 10 shows visualization of all train passages used for ML. The vectors (each containing 5 scalar features) were projected into two-dimensional space with Mathematica built-in function "DimensionReduce." The class 362 has two outliners which can easily confuse the classifier if selected into training subset or be falsely classified during validation. Furthermore, it can be seen that there is no clear boundary between classes 151 and 380. However, it is possible that, with a larger number of samples, the separation of groups would be more obvious.

4.3.1. Implementation of SVMs.

As soon as the sets were ready, the ML was performed and classifier was built. The result of the consecutive testing was confusion matrix. This process of building subsets, training and testing, was repeated 1000 times. As the outcome of this repetition process, 1000 confusion matrices were obtained (i.e., one matrix per one subsets selection).

Table 2: Comparison of ML methods.

Method	SVM	Neural net.	Log. reg.	Nearest neigh.	Rnd. forest
Accuracy (%)	75	58	67	58	50
Train. time (s)	2.0715	0.9397	0.293	0.0406	0.0529
Memory (kB)	323.384	219.512	189.240	126.824	199.520

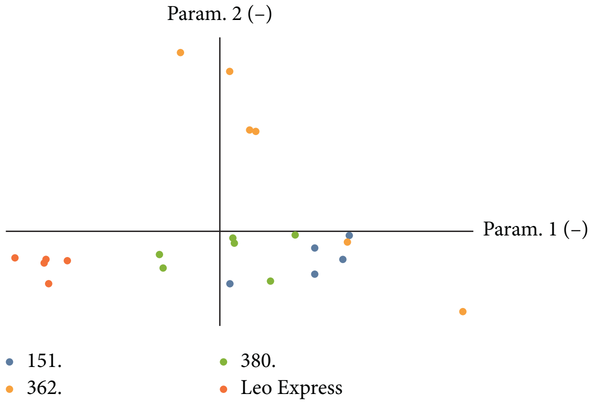

Figure 10: Visualization of clustering of scalar features for individual train passages sorted by train type.

In ML, a confusion matrix (also known as an error matrix [40]) is a very specific table layout that allows the performance of the supervised learning to be visualised (it is most frequently known as a matching matrix in unsupervised learning). Each row of the matrix indicates instances in a predicted class; each column indicates instances in an actual class (or vice versa). The name of the matrix is taken from the fact that it enables the user to check whether the system is confusing (i.e., mislabelling) two classes. It is a particular type of contingency table that has two dimensions (an "actual" and a "predicted" dimension), as well as identical sets of "class" in each dimension (each combination of dimension and class is identified as a variable in the contingency table). There are three random examples of confusion matrices from the analysis in Figure 11.

From all 1,000 matrices, one average matrix was evaluated (i.e., total sum of all results on the same location in matrix was divided by number matrices). For easier understanding, values in each row of the matrix were rescaled to give total sum of 1 so it is possible to seen probability of (miss)classification for this class. Because the test sets were not the same size, the colour of each field tells the information about significance of the testing—the darker, the higher number of test samples. That means that if there are in test subset, for example, 6 train passages from the same train type for testing and it gives the probability of correct classification 0.9, it is more reliable than if there is only 1 testing passage.

The confusion matrix shown in Figure 12 shows a perfect match for the train type Leo Express. This result was expected due to the big differences in train construction (Jacobs bogie). For the locomotive classes 151, 362 and 380, the prediction is worse due to the fact that the trains are very similar (weight, number, and distance of axes) and there was too little data for capturing such subtle differences. The locomotive class 151 is correctly classified in 70% of cases and in 25% of cases is falsely classified as a 380. In the opposite case, class 380 is classified correctly in 61% of cases and confused with 151 in 39% of cases. The classification of class 362 is reliable in 70%.

Although the data contain passages from before and after the common crossing was renovated, the identification method is sufficiently robust, based on the probabilities, to allow for railway crossing component modification, provided that measurements are obtained at the same locations, and as long as the primary objective is TIS only, not condition assessment.

5. Summary and Concluding Remarks

In this paper, the authors have conceptually approached the AI-assisted Train Identification System (TIS), a component of the self-diagnostic system for S&C, utilizing real on-site acceleration data from TEN-T railway lines in Czech Republic. This research is part of the S-CODE project; the overall aim is to investigate, develop, validate, and perform initial integration of radically new concepts for S&C with the potential to increase their capacity, reliability, and safety, while reducing investment and operational costs. Presented approach is unique in attempting the TIS based on measured acceleration time histories in S&C rather than in straight track.

The presented accuracies of the various 5 ML classifiers are clearly limited due to the number of uncontrollable variables and uncertainties, as well as due to limited number of comparable train passages, considering the dimensionality of both the physical problem and the abstract models. As the classification procedure can be sensitive to unequal class sizes, all training classes (train types) have the equal size of 4.

Although a bootstrapping analysis has been performed (1,000 training and testing subsets) in order to fully utilize the experimental evidence and to more objectively select the data for training and testing, the resulting average confusion matrices show prohibitive probabilities, which can be attributed to similarities of the 151 and 380 locomotives, low number of observations, and complex dynamic interactions at S&C in general.

Nevertheless, based on the presented theoretical and practical arguments, it can be concluded that the support vector machines (SVM) can be recommended as most suitable ML method. This conclusion is in line with the published evidence (TIS based on straight track measurements) and is supported by the presented comparison of alternative ML methods. The obvious trade-off for highest accuracy, the increased training time, and memory,

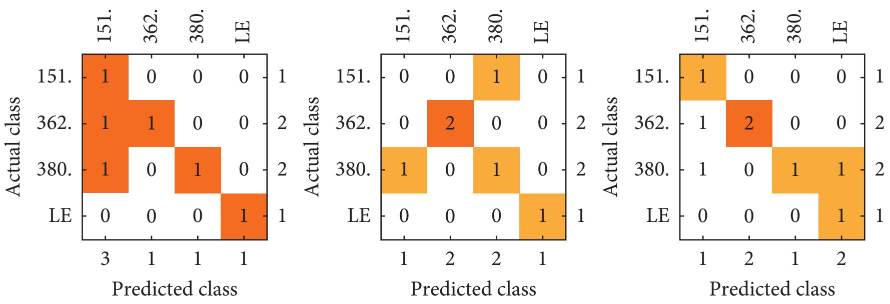

Figure 11: The 3 random confusion matrices obtained by random selection of train and test sets.

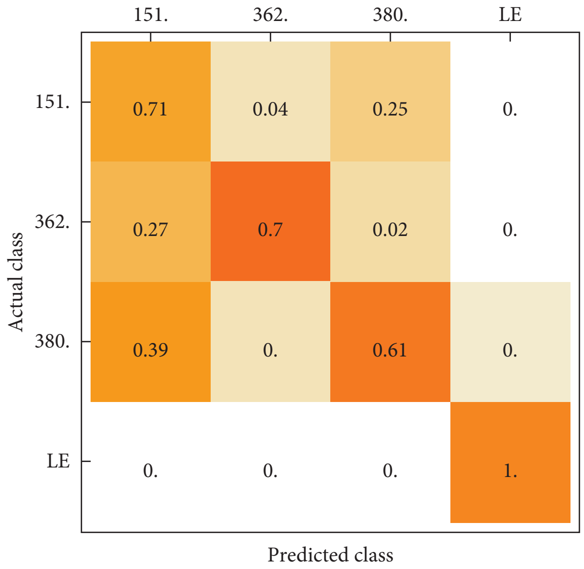

Figure 12: An average confusion matrix calculated during cross-validation using the SVM classifier. The accuracy is relatively poor at 61%, but this can be attributed to the lack of training data and train similarities.

however, is relatively cheap considering the efficiency and availability of current low (energy harvested battery powered) powered computer modules and relative to hardware resources required for statistical preprocessing of the recorded vibration time histories.

In fact, the average accuracy of 75% for SVM-based TIS at S&C cannot be considered entirely off if the published results from straight track TIS using SVM yields accuracy of 96%, and considering the inherently more complex and uncertain response of S&C compared to straight track and the clear similarities of the 151 and 380 locomotives.

For future applications within the system of early warning, it would be advisable to implement the SVM method, use it within the experimental envelope, avoid excessive extrapolation (as can be generally recommended for all ML methods), and combine the diagnostic from S&C with straight track measurement, where it is possible to identify defects on carriage, such as a flat wheel. This would dramatically improve the sensitivity and specificity of the TIS by, e.g., avoiding false positives from boogie defects. Although current optical systems can be used to identify trains by detecting and evaluating the mark placed on each locomotive, these systems are relatively expensive and sensitive to maintenance and weather conditions, compared to the vibration data-driven ML models.

One of the contributing factors to the overall uncertainty is the variable number of passengers in each wagon, significantly affecting the dynamic characteristics of the train formation. This particular aspect can be approached by trimming the signal so that only the locomotive remains, resulting in easier-to-classify data while simultaneously reducing hardware requirements. However, it would be necessary to define objective and universal applicable method of trimming, due to the complex interference of the vibrations caused by the locomotive and the following car, the nonuniform number of locomotive axles, or the presence of Jacobs bogie. In addition, by shortening the signal, some data that can provide valuable information about the condition are lost, and, most importantly, for evaluating the locomotive-only signal, analytical approaches are typically sufficient (classification based on, e.g., distance and number of axles), i.e., ML methods are not required at all.

From a pure TIS perspective, best input would clearly be represented by repeated passages of (specially scheduled) separate locomotives; however, such system could hardly be considered as an early warning system, but a preventive monitoring, as is routinely done, e.g., in the field of structural health monitoring of bridges with scheduled passages of specialized instrumented vehicles.

Although the cross-validation options available clearly limit the statistical significance, the results are unique in demonstrating that

(i) ML- (SVM-) based TIS at S&C is feasible if, within the S&C, the monitoring location is consistent. In cases in which the monitoring location is not consistent, identification is not successful.

(ii) Specifically the approach using SVM is insensitive to common crossing renovation, i.e., data from before and after the renovation can be combined, if only TIS without S&C condition assessment is considered.

(iii) The input vector that reduces full recorded time histories to a set of scalar characteristics must always be chosen subjectively so that it characterises all important features sufficiently while maintaining realistic hardware requirements stemming from the intended in-situ implementation on energy harvested battery powered modules.

(iv) During an iterative optimisation process in which accuracy is maximised and training time, evaluation time, and classifier memory and loss are minimised, the initial vector of 27 scalar features is reduced to 5.

Abbreviations

AI:	Artificial intelligence
S&C:	Switches and crossings
ML:	Machine learning
TIS:	Train Identification System
NN:	Neuron network
SVM:	Support vector machine
LE0:	Leo Express train
WT:	Wavelet transform
WVT:	Wigner–Ville transform
STFT:	Short-time Fourier transform.

Data Availability

All data are available on request through corresponding authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is part of the S-CODE project which received funding from the Shift2Rail Joint Undertaking under the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no. 730849. The support of the FAST-J-19-6062 project is also acknowledged. Furthermore, the research was supported by the project TAČR CK01000091Výhybka 4.0.

References

[1] A. M. Zarembski, "Factors involved in turnout maintenance," *Railway Track & Structures*, vol. 3, pp. 12-13, 1995.

[2] E. Kassa and J. C. O. Nielsen, "Dynamic interaction between train and railway turnout: full-scale field test and validation of simulation models," *Vehicle System Dynamics*, vol. 46, no. 1, pp. 521–534, 2008.

[3] W. J. Zwanenburg, "Modelling degradation processes of switches & crossings for maintenance & renewal planning on the Swiss railway network," Technical report, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2009.

[4] C. Ngamkhanong, S. Kaewunruen, and B. Costa, "State-of-the-art review of railway track resilience monitoring," *Infrastructures*, vol. 3, no. 1, p. 3, 2018.

[5] J. Podroužek, C. Bucher, and G. Deodatis, "Identification of critical samples of stochastic processes towards feasible structural reliability applications," *Structural Safety*, vol. 47, no. 2, pp. 39–47, 2015.

[6] M. Sysyn, O. Nabochenko, U. Gerber, V. Kovalchuk, and O. Petrenko, "Common crossing condition monitoring with on board inertial measurements," *Acta Polytechnica*, vol. 59, no. 4, pp. 422–433, 2019.

[7] R. Skrypnyk, U. Ossberger, B. A. Pålsson, M. Ekh, and J. C. O. Nielsen, "Long-term rail profile damage in a railway crossing: field measurements and numerical simulations," *Wear*, vol. 2020, Article ID 203331, 2020.

[8] A. Kowalska-Koczwara, F. Pachla, P. Stecz et al., "Vibration-based damage identification and condition monitoring of metro trains: warsaw Metro case study," *Shock and Vibration*, vol. 2018, Article ID 8475684, 14 pages, 2018.

[9] M. Hamadache, S. Dutta, R. Ambur, O. Olaby, E. Stewart, and R. Dixon, "Residual-based fault detection method: application to railway switch & crossing (S&C) system," in *Proceedings of the 2019 19th International Conference on Control, Automation and Systems (ICCAS)*, pp. 1228–1233, IEEE, Jeju, Republic of Korea, October 2019.

[10] G. Jing, M. Siahkouhi, K. Qian, and S. Wang, "Development of a field condition monitoring system in high speed railway turnout," *Measurement*, vol. 169, p. 108358, 2020.

[11] K. Gundampati, *Wireless sensor network (WSN) platform for railway condition monitoring*, PhD thesis, University of Huddersfield, Huddersfield, UK, 2019.

[12] A. Paixão, E. Fortunato, and R. Calçada, "Smartphone's sensing capabilities for on-board railway track monitoring: structural performance and geometrical degradation assessment," *Advances in Civil Engineering*, vol. 2019, Article ID 1729153, 13 pages, 2019.

[13] J. Li and H. Shi, "Rail corrugation detection of high-speed railway using wheel dynamic responses," *Shock and Vibration*, vol. 2019, Article ID 2695647, 12 pages, 2019.

[14] M. Hamadache, S. Dutta, O. Olaby, R. Ambur, E. Stewart, and R. Dixon, "On the fault detection and diagnosis of railway switch and crossing systems: an overview," *Applied Sciences*, vol. 9, no. 23, p. 5129, 2019.

[15] T. Böhm and N. Weiß, "Weichenanalytik—smarte sensoren und künstliche Intelligenz für die rundum gesunde Weiche," *Eisenbahntechnische Rundschau ETR*, vol. 5, pp. 42–45, 2017.

[16] A. Zoll, U. Gerber, and W. Fengler, "Das Messsystem ESAH-M (the measuring system ESAH-M)," *EI-Eisenbahningenieur Kalender*, vol. 1, pp. 49–62, 2016.

[17] S. Iwnicki, *Handbook of Railway Vehicle Dynamics*, CRC Press, Boca Raton, FL, USA, 1st edition, 2006.

[18] I. Vukušič, D. Sadleková, J. Smutný, L. Pazdera, V. Tomandl, and J. Hajniš, "Measurement and analysis of the dynamic effects on the crossings," in *Proceedings of the 3rd International Conference on Road and Rail Infrastructure*, Split, Croatia, April 2014.

[19] V. Reddy, G. Chattopadhyay, P.-O. Larsson-Kraik, and T. Allahmanli, "Technical vs. economical decisions: a case study on preventive rail grinding," in *Proceedings of the 5th Asia-Pacific Industrial Engineering and Management Systems Conference & The Seventh Asia-Pacific Division Meeting of the International Foundation of Production Research*, Gold Coast, Australia, December 2004.

[20] I. Vukušič, *Analysis of the Dynamic Effects in the Turnout*, Brno University of Technology, Faculty of Civil Engeneering, Institute of Railway Structures and Constructions, Brno, Czech Republic, 2015.

[21] Bogies, 2019, http://www.railway-technical.com.

[22] E. Berlin and K. van Laerhoven, "Sensor networks for railway monitoring: detecting trains from their distributed vibration footprints," in *Proceedings of the 2013 IEEE International Conference on Distributed Computing in Sensor Systems*, pp. 80–87, IEEE, Cambridge, MA, USA, May 2013.

[23] Sueur, J. Package Seewave, 2018.

[24] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.

[25] T. M. Mitchell, J. G. Carbonell, and R. S. Michalski, *Machine Learning: A Guide to Current Research*, Vol. 12, Springer Science & Business Media, Berlin, Germany, 1986.

[26] H. Tsunashima, "Condition monitoring of railway tracks from car-body vibration using a machine learning technique," *Applied Sciences*, vol. 9, no. 13, p. 2734, 2019.

[27] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2/3, pp. 95–99, 1988.

[28] Wolfram Research, I. Mathematica.

[29] J. R. Quinlan, "Learning decision tree classifiers," *ACM Computing Surveys*, vol. 28, no. 1, pp. 71-72, 1996.

[30] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[31] D. T. Bui, T. A. Tuan, H. Klempe, B. Pradhan, and I. Revhaug, "Spatial prediction models for shallow landslide hazards: a comparative assessment of the efficacy of support vector machines, artificial neural networks, kernel logistic regression, and logistic model tree," *Landslides*, vol. 13, no. 2, pp. 361–378, 2016.

[32] K. Seymore, A. Mccallum, and R. Rosenfeld, "Learning hidden Markov model structure for information extraction," in *Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction*, pp. 37–42, Orlando, FL, USA, July 1999.

[33] I. Rish, "An empirical study of the naive Bayes classifier," in *Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, pp. 41–46, Seattle, WA, USA, August 2001.

[34] Y. Liao and V. R. Vemuri, "Use of *K*-nearest neighbor classifier for intrusion detection," *Computers & Security*, vol. 21, no. 5, pp. 439–448, 2002.

[35] Powerful Guide to Learn Random Forest in R and Python, 2015, http://www.analyticsvidhya.com/blog/.

[36] The scuffle between two algorithms -neural network vs. Support vector machine, 2018, http://www.analyticsvidhya.com/blog/.

[37] Understanding Support Vector Machines Algorithm (Along with Code), 2017, http://www.analyticsvidhya.com/blog/.

[38] A. R. Henderson, "The bootstrap: a technique for data-driven statistics. Using computer-intensive analyses to explore experimental data," *Clinica Chimica Acta*, vol. 359, no. 1-2, pp. 1–26, 2005.

[39] F. Provost, "Machine learning from imbalanced data sets 101,"vol. 68, pp. 1–3, in *Proceedings of the AAAI'2000 Workshop on Imbalanced Data Sets*, vol. 68, pp. 1–3, AAAI Press, Austin, TX, USA, August 2000.

[40] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote Sensing of Environment*, vol. 62, no. 1, pp. 77–89, 1997.