2013 IEEE International Conference on Distributed Computing in Sensor Systems

Sensor Networks for Railway Monitoring: Detecting Trains from their Distributed Vibration Footprints

Eugen Berlin and Kristof Van Laerhoven
Department of Computer Science
Technische Universität Darmstadt
{berlin,laerhoven}@ess.tu-darmstadt.de

***Abstract*—We report in this paper on a wireless sensor network deployment at railway tracks to monitor and analyze the vibration patterns caused by trains passing by. We investigate in particular a system that relies on having a distributed network of sensor nodes that individually contain efficient feature extraction algorithms and classifiers that fit the restricted hardware resources, rather than using few complex and specialized sensors. A feasibility study is described on the raw data obtained from a real-world deployment on one of Europe's busiest railroad sections, which was annotated with the help of video footage and contains vibration patterns of 186 trains. These trains were classified in 6 types by various methods, the best performing at an accuracy of 97%. The trains' length in wagons was estimated with a mean-squared error of 3.98. Visual inspection of the data shows further opportunities in the estimation of train speed and detection of worn-out cargo wheels.**

Keywords*—*feature extraction, sensor data abstraction, event classification, railway monitoring, wireless sensor networks

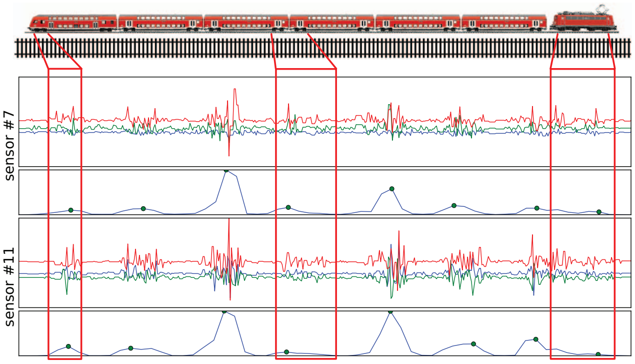


Fig. 1: Miniature sensor nodes attached to the railway tracks capture the vibrations caused by passing trains. From the raw 3D acceleration data of these events (upper plots), features can be extracted that are characteristic enough to be be used for on-sensor train classification (bottom plots). Using a network of such nodes makes the detection more robust and allows additional analysis, such as estimation of the train's speed by using time delays between sensors.

I. Introduction

Sensor networks have become a popular tool for various applications, due to being able to cover and monitor large areas and drastically reduce the intrusion into existing environments as well as disturbance of its inhabitants. The ability of wireless sensors to span a sensing and communication network with minimal resources by using small, robust, power-efficient and inexpensive hardware, highly benefits large-scale monitoring application. Such applications traditionally aim at periodic sampling of sensor values for long time periods, in order to obtain a detailed overview on physical phenomena in the environment. Many sensor network applications focus hereby on collective observation of slowly changing physical values, including temperature, humidity, gas concentrations in the air or particle concentration in the water. Hereby, the sensor nodes have to periodically wake up from low-power state in order to sample their sensors and disseminate the information through the network.

Other popular sensor network applications aim at detecting sporadic events, such as abrupt rising or falling of temperature and humidity, extremely high or hazardous concentrations of gas or pollutants in the air. The ability of the sensor nodes to detect such events directly at the source is of great advantage to the whole network, allowing to significantly reduce the amount of wireless communications within the network, thus preserving the limited power supplies.

Advances in software and hardware technology during the past decades made wireless sensor networks (WSN) more scalable, allowing the sensor nodes to be deployed for much longer time spans, in part due to the introduction of power-saving idle and sleep modes of the hardware components. With these advances, sensor networks have been increasingly deployed in scenarios with the aim to detect, monitor and report on more complex critical phenomena, such as seismic activity [1], disaster detection [2] or emergency scenarios [3], [4]. These applications require high-fidelity sensor data that preferably should be analyzed in or close to real-time, which conflicts with the fact that wireless sensor nodes within a network tend to be heavily constrained by their hardware capabilities and resources. Wireless communication is known to be one of the most energy-expensive operations, so that transmitting raw sensor data to a remote base station through the network will deplete the limited power supply in a short amount of time, resulting in sensor nodes and thus the network running the risk to become in-operational.

Our work focuses specifically on sensor data abstraction in an application where the sensors have been sampled at relatively high frequencies. Hereby, sampling rates range from hundreds of Hertz, for acceleration sensors and gyroscopes, up to thousands of Hertz for microphones. Using efficient and easy to compute features such as mean, variance, signal amplitude, and similar, abstraction of such sensor data is possible directly on the sensor nodes, even with their limited hardware

978-0-7695-5041-1/13 $26.00 © 2013 IEEE
DOI 10.1109/DCOSS.2013.38

resources. For applications where events require large amounts (i.e. multiple hundreds or thousands) of sensor readings to be adequately captured, computing such abstractions significantly reduces the amount of data in comparison to the original signal.

In this work, this approach of evaluating local features on a distributed set of wireless nodes is applied on a railway monitoring scenario. Figure 1 depicts one event from a data set recorded for the case study: The data set was obtained by deploying a network of sensor nodes that are equipped with sensitive inertial sensors at the railway tracks, capturing the vibrations caused by passing trains. We show that from the raw sensor data captured by the sensors in such a network, we are able to classify the type of train as well as the train's length. To achieve this in a realistic setting, we limit our approach to a set of sufficiently efficient features that can be implemented on the sensor node in an on-line fashion, thus allowing on-sensor event detection, train type classification and length estimation.

The remainder of this paper is structured as follows: Section II presents relevant related work. Section III is dedicated to our sensor hardware, the deployment location and the obtained data set. In section IV we propose and evaluate a set of features, both for train type classification and train length estimation. We discuss our results and highlight interesting findings as motivation for future work in section V. Finally, we list our conclusions made in this paper in section VI.

II. Related Work

Many research scenarios motivate the deployment of wireless sensor networks through the suitability of small sensors to densely monitor infrastructure, such as buildings, bridges, roads, rails. The huge diversity in sensors hardware, types of applications, deployment procedures, methodical and implementation approaches is astounding. One of the driving forces for monitoring structures and detecting relevant events is the goal for improving safety and organizing maintenance tasks. Various scenarios with alarm or control systems also motivate the detection and monitoring of critical events. This section will therefore present several application scenarios and frame our train monitoring study amid these related work.

A multitude of research, including [5], [6], [7] or [8] describe different application scenarios for wireless sensor networks, where detection and classification of rare events is of particular interest. The sensor networks, equipped with vision-based, acoustic, seismic, magnetic and infrared sensors, facilitate distributed observation of an area, aiming first and foremost at spotting and classifying ground vehicles or humans. While the scale of these deployments varies a lot, the need for energy-efficient sensors accounts for the features to be relatively simple to compute. For the car toll system application [8], the authors follow a similar approach to our work by choosing simple features (vehicle length, the average observed energy and peak-patterns in the signal) to detect and classify various ground vehicles, such as cars, pickup trucks, vans, buses and motorcycles.

Vibration sensors are often used for monitoring and ensuring infrastructure safety, such as in [9] or [10], where particular vibration frequencies in the raw data were considered as well as various complex features were utilized.

Railway safety and train detection plays an important role both from practical as well as research point of view, spawning several application scenarios utilizing different types of sensors: In [11], the authors present a system for short-term deployments that uses accelerometers to detect arriving trains in order to warn maintenance personnel working on tracks. Detecting and classifying train events by means of an acceleration sensor was suggested by [12]. To enhance railway safety, [13] deployed a vibration sensor on running trains, aiming at detecting rail deformations during motion. Electromagnetic sensor array can be used to detect and count wheels, as shown in [14]. Railroad operation monitoring through a wireless sensor network is presented in [15], aiming at more safety and improved efficiency of railway maintenance.

Reducing the energy consumption in wireless sensor network deployments, is often very crucial for the lifetime of the deployment. Data compression approaches, such as in [16], aim at reducing wireless communication payload. In the wearable and mobile sensor research domains, limited power supplies require more data processing directly at the device, avoiding energy-consuming transmission or storing to local memory. Reducing the communication payload by detecting activities directly on a mobile device, as presented in [17], will also extend the lifetime of the sensors.

Our work focuses on a wireless sensor network application scenario, where observed events can not be detected with simple threshold approaches and require on-line data processing. The aim is to classify train types and estimate train lengths by means of their vibration footprint directly on the sensor nodes. The events in this scenario tend to occur sporadically and last for only a short time period. The ability to detect, classify and monitor such events with sensor nodes deployed along railway tracks would allow to deploy such a network for various long-term railway applications.

III. Deployment

This section deals with the deployment of our sensor network that gathered the data set, the placement on the railway tracks as well as the underlying hardware choices.

A. Hardware

Since we aim at a long-term deployment of the sensor network at a given railway track of interest (Figure 2), the environment can be expected to be rough. Even though the sensor nodes used for the deployment are still research prototypes, their hardware and plastic enclosures especially manufactured for this purpose have been designed to withstand harsh outdoor environments. Further practical considerations that have been taken into account include protection of the sensor nodes against rain, snow damage, dust accumulation, and exposure to the sun, as well as attachment methods to the metal rails. As in many typical wireless sensor network deployments, we cannot assume power to be readily available next to the railroad and our deployment therefore has to rely on local batteries.

The sensor nodes for this paper's deployment and evaluation were designed to be small, robust and inexpensive enough to be left at the railway tracks surviving varying weather conditions. The sensor's main board features a Microchip PIC18F46J50 microcontroller as the main processing unit, an

Fig. 2: Our system's concept: A sensor network deployed along railway tracks captures vibrations caused by passing trains. Immediately computing efficient features from streaming sensor data allows train type classification and counting wagons on the sensor nodes. In future deployments, these can be used to estimate train speed and detect worn-down cargo wheels. As such, this system is envisioned as a flexible instrument to assist in railway monitoring and maintenance tasks.

Fig. 3: One of the sensor nodes attached to the rails during the experimental deployment. The sensor node is wrapped up in a plastic bag to protect the sensor from dust and humidity.

Analog Devices 3D MEMS accelerometer for capturing the vibrations and a micro-SD card connected via SPI-bus for local storage. Furthermore, the main board contains interfaces for reprogramming, wireless extension, and additional sensors such as light or temperature. A mini-USB port is used for connecting the sensor node to a computer, which allows configuring the sensor, initiating the logging process and accessing the recorded sensor data afterwards. A plastic case (overall dimensions: 37x33x15mm) manufactured through 3D printing holds the components and provides basic protection.

The PIC18F46J50 microcontroller features 65528 bytes of program memory and 3776 bytes or random access memory and is equipped with a real-time clock which is driven by a precise $32.768kHz$ Abracon crystal with a frequency tolerance of $\pm 20ppm$. The real-time clock drift amounts to few milliseconds per day, allowing exact time-stamping of the occurring events. Time synchronization in the network therefore needs to happen infrequently, allowing synchronized monitoring of passing trains along the mounted sensors and speed estimation based upon event delay. This particular microcontroller also supports USB communication and is able to swiftly change into low-power modes depending on the tasks at hand.

The 3D ADXL345 accelerometer is configured to sample its data at $100Hz$ and transfer the readings in bursts of 32 samples to the microcontroller. The time span between the bursts is long enough allowing the microcontroller to process the previous burst of sensor data and to switch into a low-power idle mode to preserve limited battery power. All sensor nodes were configured to sense vibrations in the $\pm 4g$ range and at 10-bit resolution. A tiny lithium polymer rechargeable battery with a capacity of $180mAh$ was chosen as the power source, allowing a single sensor node to run for approximately two weeks while logging all data to the micro-SD card.

The choice for a sensor node design with a low-power microcontroller makes the entire module small and cheap to produce, but also results in another significant challenge: The limited amount of program and operating memory as well as the lack of a floating-point unit poses a harsh limit on the used algorithms and their implementation. Thus, the proposed feature extraction routines have to work under extreme memory constraints and should avoid the use of larger functions (as they are for example used in Fourier analysis).

Choosing inexpensive off-the-shelf MEMS accelerometer sensors to detect and characterize trains by their vibration footprint results first and foremost in nodes that can easily be built in large quantities, allowing deployment of large-scale sensor networks. A second benefit lies in the accelerometer packages occupying very little space and generating sensor data at a bandwidth and resolution that can be processed directly at the sensor node with available processing capabilities. On the other hand, one can expect the sensor data quality to be less accurate than that of specifically-designed and more expensive vibration sensors, making the extraction of distinctive and characteristic features more important.

The sensors' raw acceleration logging routine requires roughly a fifth of program and random access memory. The proposed feature extraction algorithms would therefore easily fit on these sensor nodes, allowing on-line computation and forwarding of features or classification results through the sensor network instead of the much larger amount of raw sensor data. In order to evaluate these features, the following subsections present an experimental deployment for obtaining real-world vibration patterns.

B. Location

Before being permitted to deploy the sensor nodes on highly busy railway tracks, railway company officials were involved in the planning of this paper's experiments. The location for the deployment of the sensor nodes was suggested by the railway company officials themselves, specifically due to the variety of train types and their maximum possible speeds. The particular spot featured four tracks running completely straight for multiple kilometers, thus allowing train speeds of up to 250 kilometers per hour.

Two out of these four railway tracks are general purpose high-speed approved tracks for different train types, including regional and high-speed passenger trains from two European countries, and cargo trains. The other two tracks, due to their technical characteristics, were used by low-speed passenger trains connecting nearby cities.

During the deployment, each sensor was additionally wrapped up in a plastic vacuum bag in order to protect it from dust, humidity and rain, and attached to the rails using double-sided adhesive tape, as shown in Figure 3. Attaching the sensors to the rails was carried out under the supervision

Fig. 4: Snapshots from the video recordings showing in this case a single locomotive passing by on the 2nd high-speed track. These videos were used as ground truth for the evaluation of type of train and train composition (wagon count).

ID	Class	Description	Count
1	Regio	passenger trains connecting cities in a region	63
2	CityRail	trains service city center, suburbs and nearby cities	15
3	Cargo	various cargo trains	39
4	Loc	single locomotives being transferred	10
5	Thalys	French inter-city high-speed passenger train	5
6	ICE	German InterCity Express high-speed passenger train	9

TABLE I: From all the train events in raw sensor data, 141 could be annotated and used for the evaluation. Here, different train types and their count in the data set are shown.

of a railroad maintenance crew, whereby the tracks had to be officially closed for train traffic for a short period of time. For the deployment, six sensor nodes have been deployed along both high-speed and low-speed tracks, with a distance of 10 meters between them. At configuration time, the nodes' realtime clock units were synchronized with a camera setup registering both audio and video from passing trains, so that the data could easily be synchronized afterwards.

C. The Data Set: A Visual Inspection

This subsection describes the variety of trains present in the data set, consisting of train events captured by the sensors, and the video footage recorded for annotation purposes.

In total, the sensors have captured 186 train events, of which 141 could be annotated based on video footage recorded during the deployment. Figure 4 illustrates the video data with a series of frames from video footage capturing a single locomotive passing by. Table I provides an overview on the six different train types: four different passenger train classes – two types of high-speed trains, regional passenger trains and city rail trains – along with a cargo and locomotive classes.

Thalys, a French high-speed passenger train, typically consists of head and tail locomotives and 8 passenger wagons which are connected to a single continuous unit, resulting in 10 wagons in total. The last-generation InterCity Express (ICE) is a German high-speed passenger train which in our experiment typically contained 8 railmotor wagons (i.e., no locomotive). Figure 5 depicts models and truck constellations for these two train types, while Figure 6 shows examples of the captured vibration footprints and window variance used for feature extraction, revealing tiny differences: Thalys' windowed variance peaks in the middle of the train correspond to single trucks (often referred to as "Jacobs bogies") between the wagons; ICE peaks correspond to two adjacent trucks, thus resulting in slightly wider variance peaks.

The Regio class contains the regional passenger trains that connect nearby cities within a region, but do not stop at stations

Fig. 5: The four passenger train types in the data set, from above: a) Thalys, b) ICE, c) Regio, d) CityRail. Note that the trucks' locations differ among the train types, with two trucks for a wagon (e.g. ICE) or one between them (e.g. CityRail), resulting in characteristic vibration footprints.

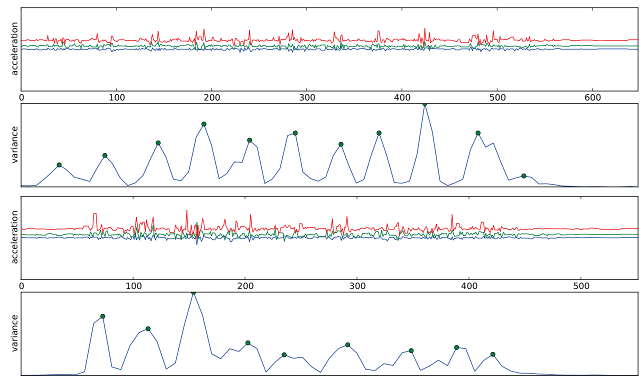


Fig. 6: Example plots showing Thalys (above) and ICE (below) high-speed passenger trains raw sensor data and windowed variance with extracted peaks. These plots show how different constellation of trucks produce distinctive vibration footprints (cf. train models in Figure 5).

in between. Regional trains consist of a locomotive pulling or pushing a number of bi-level wagons (as shown in Figure 2 and the corresponding model in Figure 5c). In our experiment, these trains' lengths were 3, 5, 6 and 7 wagons in total.

The Cargo class has proven to be rather versatile, with one characteristic feature that all cargo trains have in common: at least one locomotive is pulling a highly varying number of wagons. Both the locomotives as well as the wagons themselves can be of different types (e.g., tanks, containers, car- or freight wagons), as well as different lengths and truck constellations. In our experiment, cargo trains had mostly one, sometimes two, locomotives with a total number of wagons ranging from 13 up to 43.

The locomotives class was added due to single locomotives being transferred to another station. In the experiment, 10 such events have been captured, whereby both single as well as two connected locomotives have been observed.

The city railway trains (CityRail) connect larger cities with its suburbs and other smaller towns nearby. The train typically consists of two electrical units with 4 wagons each (Figure 5d). The CityRail trains in our experiment were running exclusively on the separate low-speed tracks.

With this data set, we can now perform an extensive evaluation, in which particular focus is given to finding a set of efficient-to-calculate features that can be implemented directly on the sensor nodes for on-line train type classification. A second objective that has been identified as valuable information to automatically detect by the sensor network is the estimation of train length. The following section will present the proposed features, the classification performance and train length estimation results.

IV. Evaluation

This section is divided into three parts. First we present features that can be efficiently extracted on the sensor nodes from raw 3D accelerometer data. The second part deals with the train type classification performance of the proposed features, specifically aiming at finding the optimal parameters as well as the best performing combinations of features. The third part evaluates how well the train length can be estimated.

A. Features

Aiming at a railway monitoring application which relies on detecting and classifying passing trains directly at the sensor nodes, a set of characteristic features that are able to describe and distinguish various train events is necessary. In this scenario, train events occur sparsely over the course of time, so that most sensor data acquired by the sensor nodes can be discarded as not relevant, when no trains are passing by. These flat signal sections between train events can be accurately segmented by utilizing the standard deviation in a sliding window over signal.

When a train passes by, the sensor node will be able to detect this event by the changing acceleration values, and due to the real-time clock also the exact time when the event begins and ends. From this, the duration of an event can already be derived as the first feature. When considering the whole event, the total amount of vibration caused by the train can easily be computed in an on-line fashion and thus used as a feature.

Visual inspection of the vibration footprints has led to the assumption that single trucks (containing one or multiple axles) can be found in the signal. To achieve this, a small window buffer is used to compute windowed variance from the raw sensor data (cf. Figure 1) or 6), which can then be used to derive further features. The number of peaks in the windowed variance plot is such a feature, which captures trucks and can be used to count the number of wagons. Since the peaks' detection highly depends on the width of this sliding window, it is being considered a parameter in the following evaluation.

After computing the peaks, more information can be extracted: the amount of vibrations of the trucks through maximum and average of the amplitudes, truck distances through the average distance between peaks, variety of wagon lengths or trucks constellations via variance of peak distances. Additionally, the overall area under the variance curve, as well as the average area per peak will be considered. For the offline evaluation we compute this feature using the Python `scipy.integrate` library. Table II summarizes the proposed features that were used in this study.

With this set of features at hand, our interest lies in finding the appropriate parametrization for the sliding window and from that the best performing feature combination.

ID	Feature	Description
0	duration	event duration (vibration exceeding a threshold)
1	variance	total amount of vibration caused by the train
2	peaks	number of peaks extracted from windowed variance
3	max. amplitude	maximum peak value
4	avg. peaks	average distance between peaks
5	avg. amplitude	average peak amplitude
6	area	total area under curve
7	avg. area	average peak area under curve
8	var. peaks	variance of peak distances

TABLE II: Overview of all features considered for train type classification. During the 5-fold cross validation on the data set using an SVM classifier, all possible feature combinations (with a minimum number of three) have been tested, whereby the features were also computed with a varying window size.

B. Train Type Classification

This section discusses the feature extraction parametrization, mainly depending on the window size over which the windowed variance of the raw signal is being computed. Since the sensor nodes were set to a sampling frequency of $100Hz$, the following range of window sizes was found to be of interest for evaluation: 12, 14, 16, 18, 20 data points.

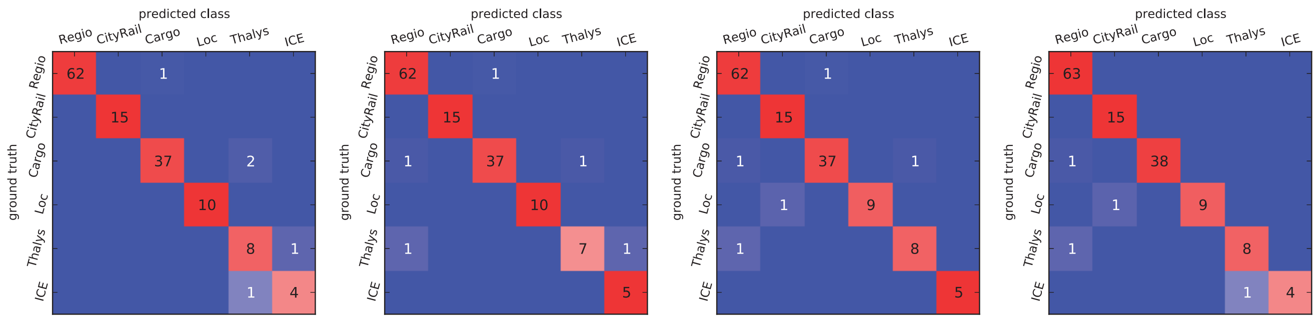
For the train type prediction, the versatile support vector machine (SVM) classifier has been chosen. The feature space was normalized before being used for the evaluation.

The performance evaluation was carried out through a stratified 5-fold cross-validation, whereby the size proportion of the six classes was preserved. The classifier was trained on 4 parts, while one part was left out for testing. Hereby, all possible feature combinations have been tested (with three as a minimum features set size), resulting in 466 combinations. Multiplying this with the range of window sizes, we end up with 2330 cross-validation runs. Here, we present the cheapest (regarding the number of features required) best performing feature combinations.

The classification results obtained during the 5-fold cross validation were accumulated, and confusion matrices were computed averaged over the number of folds. Figure 7 shows the four most illustrative confusion matrices, along with their parameters, the window size and the set of features.

For the first evaluations with the first three features only (duration, total variance and number of peaks), the SVM classifier was able to reach an overall accuracy of 90.78% for the window size of 18 data points. Adding the maximum amplitude to the feature set led to an increase of total accuracy 93.62% for the window size of 16. Adding more features to the set or interchanging them would improve the accuracy in very little steps, such that many combinations would reach a classification performance with 136 out of 141 train types correctly identified (96.45% accuracy). Figures 7a, 7b and 7c show three confusion matrices with corresponding feature sets that were able to achieve this high classification performance.

The feature set consisting of feature IDs 1, 4, 5, 6, 7 and 8 (computed from the windowed variance with a window size of 16 data points) has reached the maximum possible accuracy of 97.16%, with 137 of 141 train events being correctly identified. Figure 7d depicts the confusion matrix for this best performing feature set and window size.



(a) ww=14; ftrs=[2, 3, 4, 5] (b) ww=16; ftrs=[2, 4, 5, 6, 7] (c) ww=16; ftrs=[1, 2, 4, 6, 7, 8] (d) ww=16; ftrs=[1, 4, 5, 6, 7, 8]

Fig. 7: Exemplary selection of confusion matrices obtained during the 5-fold cross-validation with the SVM classifier. The first three matrices show an accuracy of 96.45%, for different sets of features computed on different window sizes. The right-most confusion matrix shows the feature set performing best, reaching an accuracy of 97.16%.

These classification performance on our data set suggests that training an SVM classifier offline and implementing it on the sensor nodes would allow to detect train events, compute features and predict train types directly at the signal source with high accuracy.

Predicting the types of passing trains with reaching accuracies up to 96% and 97%, would be sufficiently promising for several applications. Following our scenario, deploying a sensor network with such a SVM classifier implemented on each sensor node, it is still possible to improve on the classification performance. This can be achieved by utilizing the sensor network's communication capabilities and let neighboring sensor nodes decide upon the train type by a voting mechanism amongst classifiers.

After evaluating the train type classification performance with features, the following section will give insight on how well the train length estimation worked.

C. Train Length Estimation

To estimate the train length, we primarily rely on counting the number of wagons in the trains. This can be achieved by using the already introduced feature "number of peaks" as a basis. Additionally, using the train type obtained from the previous estimation step is used as a prior. The wagon count can furthermore be improved by a comparison and voting procedure amongst neighboring sensor nodes on the same railroad track.

Besides using the raw signal and computed features, it is useful to include inherent model knowledge about the train type constellations: The ICE and Thalys high-speed passenger trains as well as the CityRail trains consist of specific wagons only (locomotives are built-in or the wagons are motorized themselves). Regio trains consist of varying amount of wagons with a separate locomotive. While with these trains the axles constellations are fixed due to defined sets of wagons, the cargo train class poses a much higher variety: wagons with single, double and triple axles per truck, wagons of different lengths, and varying load are possible.

Using the annotations from the video footage as ground truth (number of wagons) and the number of peaks extracted

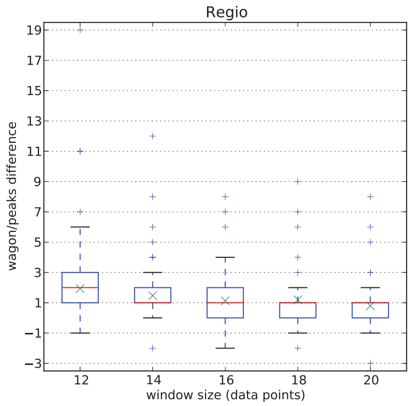
Window	Overall	Regio	CityRail	Cargo	Loc	Thalys	ICE
12	28.14	11.78	1.87	78.95	7.30	4.00	2.00
14	10.33	4.65	1.00	27.95	2.60	3.56	0.20
16	**4.02**	2.84	0.27	**8.74**	2.20	2.33	**0.00**
18	**3.98**	2.89	0.33	8.87	1.20	**1.44**	0.60
20	5.62	**2.43**	**0.20**	14.28	**1.00**	6.56	2.00

TABLE III: Mean-squared error of the estimated train lengths for the whole data set and per class. The window size has a huge impact on the quality of the estimation. Overall, window size of 18 performs slightly better than a window size of 16.

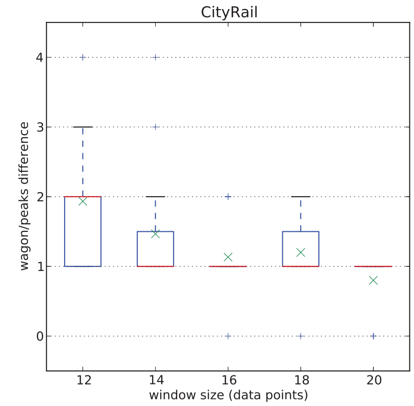
from the windowed variance, we use their difference for performance analysis. Since the peaks correspond to the trucks, the number of peaks usually is by 1 more than the amount of wagons in the train. This deviation of 1 can be visually recognized in the box plots shown in Figure 8. In addition, we compute the mean-squared error for the whole data set as well as for each individual class (see Table III), whereby the deviation has been accordingly taken into account.

For a more concrete example, let us consider a regional passenger train with 7 wagons (including the locomotive). For this train, the peak detection algorithm extracts 8 distinctive peaks (cf. Figure 1). Hereby we observe that the first peak belongs to the first trucks (double axles) of the first wagon, the following 5 peaks belong to adjacent wagon trucks (two times double axles for passenger wagons), and the last two peaks represent the last wagon and the locomotive (which has triple axle trucks that can not be distinguished in the signal with the fixed window size). Due to Regio trains' variety in length (3, 5, 6 and 7 wagons including one or even two locomotives) and their varying speed when passing by the sensors, the relation of wagons to the number of peaks tends to highly deviate as well as show lots of outliers (Figure 8a).

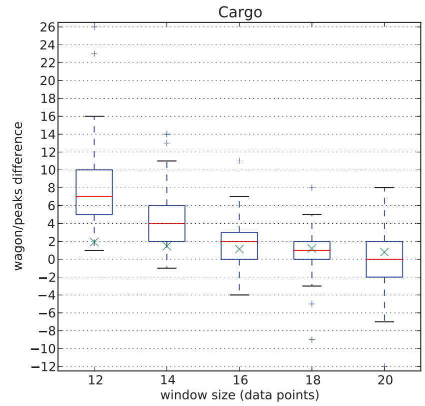
The window size to compute the windowed variance from raw sensor data has a significant impact on the peak detection. On the other hand, leaving the window size fixed at the best performing size of 16 data points (for classification and length estimation) would deteriorate the system's performance, as a fixed window size for a fixed sampling rate of the sensor node leads to the issue of not being speed independent.



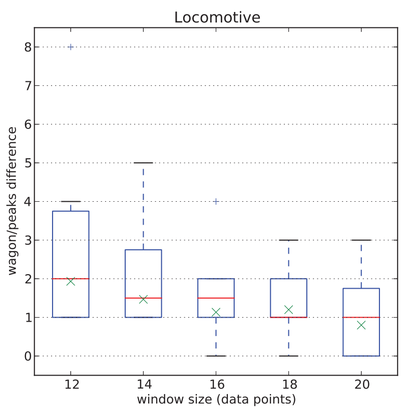
(a) Regio trains with varying length and speeds result in high deviations and outliers.



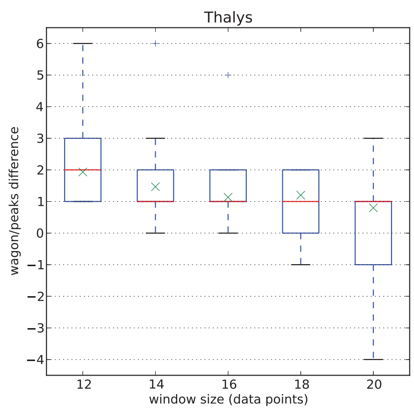
(b) CityRail trains' trucks can be quite accurately detected with window size of 16 & 20.



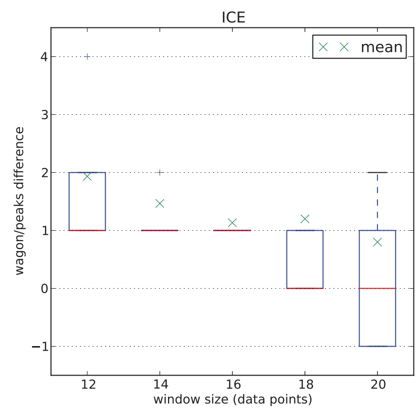
(c) Cargo trains' variety in number and types of wagons results in high peak count variance.



(d) High variance for single locomotives due to locomotive type, axles configuration, speed.



(e) Thalys trains trucks between wagons can be best detected with a window size of 18.



(f) ICE trains double-axles trucks per wagon are optimally detected with a window of 16.

Fig. 8: Differences between the real number of wagons and number of peaks computed from the windowed variance for each of the six train classes. The number of peaks and therefore the accuracy of the wagons count depends on the size of the sliding window and the train speed. From these results, window sizes of 16 and 18 data points are performing best for counting wagons. This is verified by the overall minimum squared-mean error of approximately 4.0 shown in Table III.

V. Discussion and Outlook

This section discusses the evaluation results and will point out particularly interesting findings for the underlying scenario.

First, good train type classification results (up to 97% accuracy) could be achieved on our data set with proposed features. During the evaluation, suitable window sizes (16 data points) both for type prediction and train length estimation could be found. Better performance in this regard can be achieved through implementing distributed voting among neighboring sensor nodes inside the sensor network. This would allow the sensor nodes to compare decisions and remove outliers.

Estimating train length with a fixed window size bears the problem of not being speed independent. In case of a very slowly moving train, which is very likely to happen and has also been observed in our data set, the fixed window will lead to detecting separate axles instead of trucks, resulting in a completely misleading peak count. One possible approach to tackle this issue would be the introduction of a variable window size. This would, on the other hand, lead to a more complex feature extraction routine and result in more computation on the sensor node and therefore in a higher power consumption.

Besides the problems addressed in this work, the recorded data set allows to extract much more information useful for the railway monitoring scenario. Estimating train speeds belongs to this category of very useful details and can be achieved with multiple sensors placed at a predefined distance which, in our experiment, was 10 meters. Detecting a passing train on two sensors and then computing the time delay between event arrival seems to be an easy and reasonable approach. For this, time synchronization inside the sensor network is important, but can be nowadays considered as a solved problem. An example for the feasibility is shown in Figure 9: vibrations caused by a passing regional passenger train are captured by two sensors in 10 meters distance from each other. By aligning these raw sensor data in time, the delay became visible.

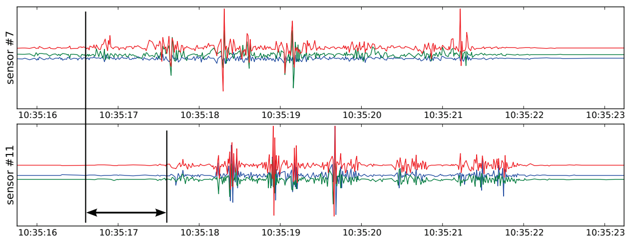


Fig. 9: Two sensor nodes showing the vibrations footprint caused by a passing regional passenger train. Knowing the distance between the sensor nodes ($10m$ in our deployment) and the time delay of the event between two sensors (markers in the plots) will allow estimating the train speed.

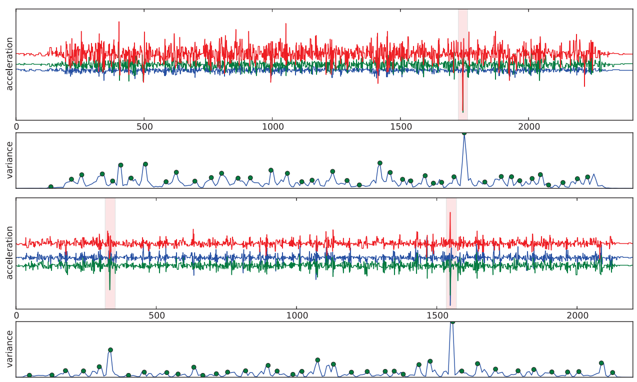


Fig. 10: The sensor nodes have picked up extreme impacts (large peaks in raw data and variance plots) from a passing cargo train, caused by a defect wheel that is not completely balanced due to wear during braking. Since these wheels could cause damage to rails, rail bed, and the wagons, detecting such events automatically would be of significant interest as well.

Another very promising application for such a sensor network would be the detection of worn-out or defect wheels. Figure 10 shows data from two sensors which have picked up extreme accelerations caused by a passing cargo train. These extreme amplitude peaks in the raw sensor data are most likely caused by worn wheels (having lost their roundness due to abrasion caused by blocking when the train breaks). These worn-down wheels could cause damage to rails or rail bed, as well as the wagons themselves, thus making the detection of such events particularly interesting.

VI. Conclusion

We evaluated the suitability of a sensor network consisting of tiny, inexpensive sensor nodes for a train monitoring application. It relies on sensor data from 3D MEMS accelerometers that are able to capture vibrations caused by running trains. To enable in-network event detection and train type classification, we have proposed a set of features that can be computed efficiently and in an on-line fashion directly on the sensor nodes. After deploying the sensors at railway tracks, recording a real-wold data set and video footage for annotation purposes, we conducted an evaluation of the proposed features. Using an SVM classifier, the feature set (1, 4, 5, 6, 7, 8) and the window size of 16 data points were found to produce optimal results for this data set. The SVM classification performance reached 97% accuracy. The length estimation performance accounted to 3.98 mean-squared error for the whole data set.

Acknowledgment

We would like to thank Hans Gabler, Hubert Schmühl and Harald Behr from the Deutsche Bahn Netz AG who made this real-world deployment possible. This work was sponsored by the Project "Long-Term Activity Recognition with Wearable Sensors" (LA 2758/1-1) from the German Research Foundation (DFG).

References

[1] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, "Deploying a wireless sensor network on an active volcano," *Internet Computing, IEEE*, vol. 10, no. 2, pp. 18–25, 2006.

[2] M. Ramesh, S. Kumar, P. Rangan, and A. Vidyapeetham, "Wireless sensor network for landslide detection," in *International Conference on Wireless Networks (ICWN 2009)*, 2009, pp. 89–95.

[3] T. Gao *et al.*, "The advanced health and disaster aid network: A light-weight wireless medical system for triage," *IEEE Trans. on Biomedical Circuits and Systems*, vol. 1, no. 3, pp. 203–216, September 2007.

[4] T. Gao, C. Pesto *et al.*, "Wireless medical sensor networks in emergency response: Implementation and pilot results," in *Conference on Technologies for Homeland Security (HST 2008)*, 2008, pp. 187–192.

[5] G. P. Mazarakis and J. N. Avaritsiotis, "Vehicle classification in sensor networks using time-domain signal processing and neural networks," *Microprocess. Microsyst.*, vol. 31, pp. 381–392, 2007.

[6] S. Gupte, O. Masoud, R. Martin, and N. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 37–47, March 2002.

[7] P. Dutta *et al.*, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," in *Int. Symp. on Information Processing in Sensor Networks (IPSN)*, 2005, pp. 497–502.

[8] S. Keawkamnerd, J. Chinrungrueng, and C. Jaruchart, "Vehicle classification with low computation magnetic sensor," in *Int. Conf. on ITS Telecommunications (ITST 2008)*, 2008, pp. 164–169.

[9] H. Jingqiu *et al.*, "On vibration signal analysis in bridge health monitoring system by using independent component analysis," in *Society of Instrument and Control Engineers (SICE)*, 2010, pp. 2122–2125.

[10] K. Sukun *et al.*, "Health monitoring of civil infrastructures using wireless sensor networks," in *Int. Conf. on Information Processing in Sensor Networks (IPSN 2007)*. ACM, 2007, pp. 254–263.

[11] L. Angrisani, D. Grillo, R. Moriello, and G. Filo, "Automatic detection of train arrival through an accelerometer," in *Instrumentation and Measurement Technology Conference (I2MTC)*, 2010, pp. 898–902.

[12] E. Berlin and K. Van Laerhoven, "Trainspotting: Combining fast features to enable detection on resource-constrained sensing devices," in *Int. Conf. on Networked Sensing Systems (INSS)*, 2012, pp. 1–8.

[13] C. Wang *et al.*, "On-line vibration source detection of running trains based on acceleration measurement," in *Int. Conf. on Intelligent Robots and Systems (RSJ 2006)*, 2006, pp. 4411–4416.

[14] P. Donato, J. Urena, M. Mazo, J. Garcia, and F. Alvarez, "Electromagnetic sensor array for train wheel detection," in *Sensor Array and Multichannel Signal Processing Workshop*, 2004, pp. 206–210.

[15] E. Aboelela, W. Edberg, C. Papakonstantinou, and V. Vokkarane, "Wireless sensor network based model for secure railway operations," in *IPCCC 2006*, 2006, pp. 624–628.

[16] A. Reinhardt, D. Christin, M. Hollick, and R. Steinmetz, "On the energy efficiency of lossless data compression in wireless sensor networks," in *Conf. on Local Computer Networks (LCN 2009)*, 2009, pp. 873–880.

[17] F.-T. Sun, C. Kuo, and M. Griss, "Pear: Power efficiency through activity recognition (for ecg-based sensing)," in *Int. Conf. on Perv. Comp. Techn. for Healthcare (PervasiveHealth)*, 2011, pp. 115–122.