2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops) | 978-1-6654-0424-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/PERCOMWORKSHOPS51409.2021.9430996

Towards Predictive Safety Maintenance for IoT Equipped Bikes

Viktor Matkovic
University of Duisburg-Essen
Duisburg, Germany
viktor.matkovic@uni-due.de

Marian Waltereit
University of Duisburg-Essen
Duisburg, Germany
marian.waltereit@uni-due.de

Torben Weis
University of Duisburg-Essen
Duisburg, Germany
torben.weis@uni-due.de

***Abstract*—We present a novel approach for predictive maintenance using acceleration data. Modern bikes can be equipped with additional smart features that enable early detection of deteriorating brake performance. This allows individual user feedback based on the condition of their bikes and therefore improve safety. We evaluate the suitability of various machine learning approaches for predictive maintenance using acceleration data of bike rides with good and bad brake performance. Here we compare two methods of measuring acceleration, namely we use hall sensors and inertial sensors. Overall, we achieve a F1-score of up to 0.76 using the time series specialized k-nearest neighbor in a preliminary evaluation. Furthermore, our results show that inertial sensors are better suited for measuring acceleration data than hall sensors.**

***Index Terms*—Predictive Maintenance, Bikes, Smart Devices, Datasets, IoT**

I. Introduction

People are becoming more eco-conscious and are moving to classic means of transportation such as the bike. For example, in Germany more than 44% of all people use their bikes regularly [1].

Many efforts are being made to improve urban mobility. Digitalization is contributing to the widespread use of emerging mobility services such as bike or e-scooter rentals. In urban areas, these rentals are ideal for short-distance commuting and a convenient alternative to traditional public transportation.

Another popular trend is the e-bike, which supplements a bike with an electric motor that supports cyclists during their ride by reducing paddling effort. Bikes-sharing vehicles or e-bikes are usually equipped with system on a chips (SOC) and also fitted with various sensors, e.g. to track the bike in case of bike-sharing or to drive the electric motor in case of e-bikes. However, digitalization efforts neglect the potential for improving safety for new mobility solutions with the help of IoT hardware.

Predictive maintenance techniques for instance are widely used for heavy machinery and tools to prevent catastrophic failure. In the case of bike-sharing services, maintenance is usually initiated by direct feedback from users or by predefined maintenance time intervals. Relying on user feedback is not practical as it is uncertain whether feedback will be given. In addition, exceptional circumstances may reduce the brake performance of a particular bike acutely, so that predefined maintenance intervals will no longer be sufficient.

Modern e-bikes for private use can be equipped with additional smart features that utilize early detection of deteriorating brake performance to provide users with individual feedback based on the condition of their bikes. With brake behavior/performance in the following discussions, we refer to the brake behavior/performance of bikes.

This paper is organized as follows. First, we discuss related work in Section II. In Section III, we present the method of data acquisition and pre-processing and the different machine learning approaches we tested. In Section V, we evaluate the feasibility of our approach for future work. Finally, we conclude the paper in Section VI.

II. Background and Related Work

The role of maintenance is to prevent losses due to equipment failures before they occur. The term "predictive maintenance" is often used in connection with resource maximization in large industrial plants with the justification of cost reduction. There are also examples of cars where abrasive parts such as brake discs or brake blocks close a circuit when enough material has been worn off and warn the driver of potential brake loss in the near future [2]. Such systems do not exist for bikes, to the best of our knowledge.

There are efforts to investigate damage to certain components of mountain bikes using smartphones [3]. This involves recording vibrations and distinguishing between broken and normal components through feature engineering and machine learning (SVM). The components that can be examined are the rotor, chain, wheel bearing, steering head, derailleur cog. However, these components are not directly involved in the optimal brake characteristics. Furthermore, the mountain bike must be mounted on a maintenance rig and the smartphone has to be placed in close proximity to the relevant parts for evaluation, which makes everyday use more demanding.

Another project tries to infer maintenance probabilities from maintenance protocols based on bikes from the city of Oslo (Finland) [4]. The project uses a random forest machine learning approach with several features, including: ride duration, year of birth of the cyclist, gender and date, divided into month, hour and day. This approach does not work for individual bikes and is more interested in the general maintenance behavior of cyclists. For example, the month of April is of great importance for the accuracy of the model, which can be explained by seasonal preventive bicycle safety checks to be prepared for warmer, more ideal weather conditions for cycling.

Influences on the brake performance on bikes are manifold, for example: (1) damaged or kinked brake cables, (2) insufficient lubricant in the brake cable housing, (3) air or water in hydraulically operated brakes, (4) insufficient tire traction on the road, (5) long hand brake lever travel, to name a few situations in which bikes suffer a loss of brake power. Apart from direct physical influences, the cyclist's riding behavior also affects the brake power (either positively or negatively). Therefore, we try not to identify the individual influences on the brake performance, but the overall brake characteristic. Ultimately, we compare the deceleration during brake events. If we compare past and present brake performance and observe that it deteriorates, we may give the driver an early recommendation for maintenance.

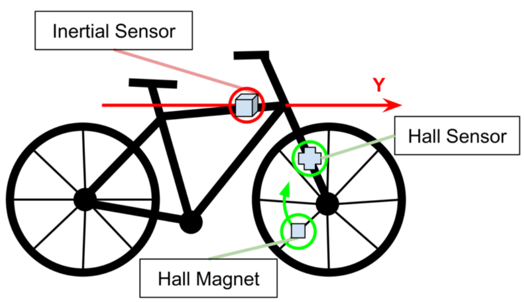


Fig. 1: Overview of two methods for measuring acceleration. The first method is the use of inertial sensors (red). Such sensors record acceleration in three spatial dimensions (x,y and z). A single axis must be oriented in the direction of the frontal motion of the bike. In our particular scenario, we chose the y-axis. The second method uses hall sensors and magnets attached to the spoke of the front wheel (green). For more information, see Sections IV-A and IV-B.

III. System Model

We assume that a bike is equipped with embedded hardware for processing acceleration data. In this work, we consider two methods for acquiring acceleration: hall sensors and inertial sensors. Hall sensors can be used to monitor the speed/acceleration of a given bike. The sensor detects and measures the magnetic field passing it. This type of sensor is widely used for speed measurement and other high-speed switching applications. Inertial sensors use the piezoelectric effect to measure acceleration. When stress, e.g. acceleration, is applied the sensor creates electrical charges. Furthermore, we require a ride as a whole and to be recorded continuously. The resulting data is then pre-processed and evaluated by our approach.

Our approach is based on continuous acceleration/motion data. We make no restrictions on how acceleration/motion is measured. In this paper we focus on two methods for measuring acceleration/motion: (1) inertial and (2) hall sensor.

We consider in this paper: (1) the hall sensor is placed on the front fork, (2) a magnet is fitted to a front wheel spoke, (3) inertial sensors attached at a fixed position and (4) all sensors connected to a central computing unit. See Fig. 1 for an overview of the placements of the individual components on the bike.

As a baseline for training and testing we use an ideal cyclist with exemplary brake and traffic behavior on the road. We define cyclists as exemplary when, they follow laws, established guidelines and advice on how to behave in public road traffic with the bike. To stop a wheel optimally the cyclists must pull both brakes at the same time. Furthermore, the cyclists should pull the front brake so firmly that the rear wheel does not lift off. If the rear wheel is lifted, the rear brake cannot work properly since the rear wheel lacks traction.

IV. Approach

We try not to identify the individual influences on the brake performance, but the overall brake characteristic by focusing on the deceleration during bike trips. If we compare past and present brake performance and observe that it deteriorates over time and therefore derive that safety might be compromised.

This Section is structured as follows: First, in Section IV-A and IV-B, we describe two methods of collecting acceleration data. In Section IV-C, we describe our method for filtering data that degrades classification performance. Finally, in Section IV-D, we outline several machine learning models for later evaluation.

A. Hall Acceleration Calculation

The hall sensor is placed on the front fork and a magnet is fitted to a front wheel spoke and aligned with the hall sensor. While cycling the magnet rotates with the front wheel. With each rotation of the wheel the magnet moves past the hall sensor. Each time the hall sensor measures a magnetic field, it triggers (see Fig. 1). For each trigger we calculate the time difference between two successive rotations where $\Delta t_i = |t_i - t_{i+1}|$ for all rotations $i \in N-1$. With the circumference (c) of the front wheel we calculate the average delta velocity (v) and acceleration (a) for each Δt where:

$$v_i = \frac{c}{\Delta t_i} \quad a_i = \frac{v_i - v_{i-1}}{\Delta t_i}, i > 0$$

B. Inertial Sensor Acceleration

To use the acceleration data of the inertial sensor effectively, some aspects must be considered. First, we need a linear acceleration without the acceleration due to gravity, since this introduces a bias into the data. Second, we need to remove the inherent rotation bias from the data as much as practically possible. Most modern inertial sensors provide the necessary capabilities to handle those requirements.

We define acceleration as a 3D-vector with the components x, y and z. The linear acceleration vector $\vec{a}(t)$ provides for time t the acceleration on all components x, y, and z minus the earth's gravitational pull. The components are fixed relative to

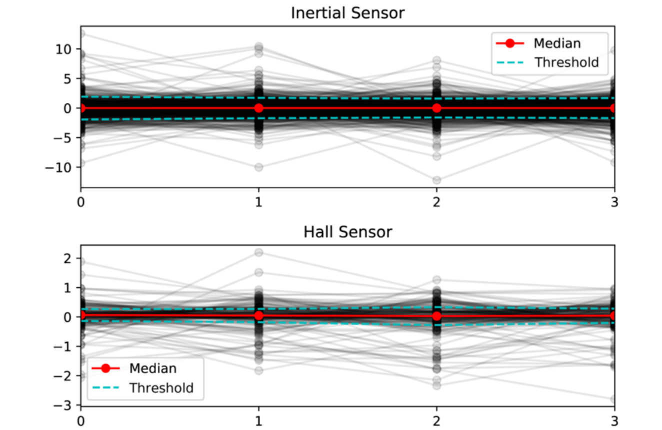
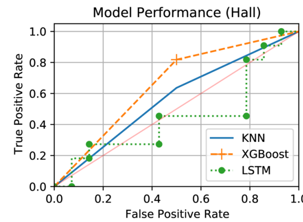
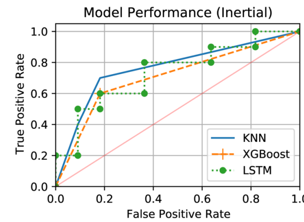


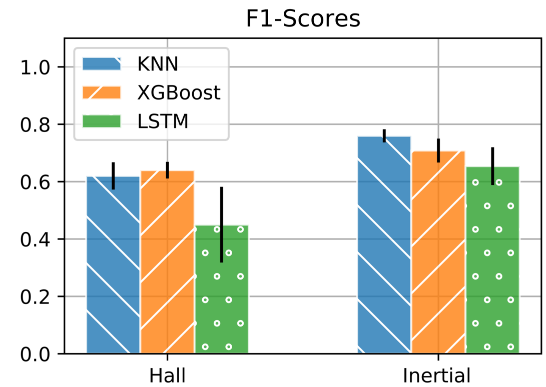
Fig. 2: Overlapping illustration of all data and filtering of time series of $W = 4$ window points. All points in a window must be outside the threshold, otherwise the window will be filtered out. Most of the points are around the zero line.



(a)



(b)



(c)

Fig. 3: The ROC curves (a & b) and F1-score (c) results for each classifier for both datasets hall ($\bullet_{\mathrm{H}}$) and inertial ($\bullet_{\mathrm{I}}$). KNN performed best overall on both datasets, achieving an AUC of 0.57_{H} and 0.76_{I} in tandem with F1-scores of 0.62_{H} and 0.76_{I}. XGBoost performed best with the hall sensor dataset. Overall resulting in AUC of 0.66_{H} and 0.71_{I} as well as F1-scores of 0.64_{H} and 0.71_{I}. LSTM performed the weakest with an AUC of 0.45_{H} and 0.65_{I} along with F1-scores of 0.45_{H} and 0.65_{I}.

the sensor position, i.e. the measured forces acting on the axes depend on the orientation of sensor.

For the sake of simplicity, we limit the rotation to a fixed position on the bike. We assume that our system is used in a SOC where the SOC is housed in a bike. This allows us to deal effectively with rotational bias.

In general, any axis (x,y or z) is suitable for our intended use. It only must be aligned in the direction of the frontal movement of the bike. In our particular scenario, we choose the y-axis, see Fig. 1. This setup allows us to use the y-acceleration for further calculations. We sample the sensor at a rate of $100\mathrm{Hz}$ and calculate the average acceleration in windows of $200\mathrm{ms}$.

C. Windows and Filtering

Normally, the brakes are not used most of the time while riding a bike. The variance in acceleration is also very small, as there is no abnormal acceleration while riding normally. Therefore, we need to filter out most of the data points. The data is aggregated in windows of $W = 4$ points for filtering and machine learning. For example for the inertial sensor data a window would contain 4 data points of which each is averaged in intervalls of 200ms. The time delta between the individual points in the window may be arbitrary. We calculate for each index in the window the vertical sample median and variance. We keep a window if all W points in the window are outside of the median $\pm$ variance. See Fig. 2 for an illustration of the filtering.

D. Machine Learning

We implemented three machine learning methods to assess the feasibility of our proposed idea, this includes:
(1) Time series implementation of k-nearest neighbor (KNN), which utilizes dynamic time warping (DTW) to measure the similarity between two arbitrarily time sequences [5], [6]. For inertial acceleration data we use soft-DTW which is an extension of the DTW proposed by Cuturi et al [7] to enable differentiability of DTW. Soft-DTW showed that it is robust to time shift/dilation/reduction and enabled gradient descent as a cost function. (2) XGBoost which is optimized for gradient tree boosting with decision tree ensembles [8]. Hereby we use standard deviation, variance, mean, median, min and max values for each window as features. (3) Long short-term memory (LSTM) classifier using mean square error as our cost function implemented with Keras [9].

V. Evaluation

We recorded two trips with a bike in ideal condition for optimal braking (about $1\mathrm{km}$ in length). We also made an effort to achieve a similar speed for each trip to avoid bias in the evaluation of the data. Afterward we modified the bike in such a way that the brake performance deteriorated considerably. This includes the lengthening of the brake cable and the widening of the brake clasp. With the worse brake performance we have carried out two trips as well. We labeled all data that remained after filtering (see Section IV-C) based on the brake performance for each trip.

Thus, we have two classes (good/bad) that we use for binary classification. We evaluated the predictive performance of the classifiers by utilizing stratified k-fold cross-validation with $k = 2$. Stratified refers to each test and train fold/set containing the same class proportion to avoid bias.

For ease of use we employed a smartphone inertial sensor to measure acceleration while cycling. The smartphone is fixed to the bike frame in a custom-made housing, which also houses an ESP-WROOM-32 module [10]. The module is a low-cost and low-power system on a chip (SOC) with Wi-Fi capability. The remaining sensors (i.e. hall) are connected to the module. The module itself is directly connected to the smartphone via Wi-Fi. All sensor data (including those of the module) are time stamped by the smartphone. We measured the delay between the smartphone and the module. The results show a negligible start time delay of less than 2 ms.

We implemented a native Android application to record the readings of the inertial sensors and receive the data from the SOC module via Wi-Fi. Furthermore, we have implemented a custom serialization protocol to avoid computational and memory bottlenecks in both the SOC and the smartphone.

We use the receiver operating characteristics (ROC) with the area under the ROC curve (AUC) to visualize the performance of the classifiers tested [11]. The straight line in the figures shows the classification performance when all predictions of a classifier are random. This means that anything above this line is better than chance. The AUC is a scalar value used for comparison purposes. If a classifier has perfect predictive performance, the AUC of the ROC is 1. Thus, the higher the AUC, the better the predictive performance of the classifier. Furthermore, we use the F1-score to avoid imbalances in favor of other classification classes.

Looking at Fig. 3b, 3a and 3c, we see the ROC, AUC and F1-scores for all the classifiers trained with the hall and inertial acceleration data. The result shows a high discrepancy between the hall and inertial data. Here, the classifier trained with the data from the inertial sensor performed best. Overall, all three classifiers could distinguish relatively well between good and bad brake performance. The KNN classifier performed best with an F1-score and AUC of 0.76 and 0.76 respectively. With hall sensor data, only the XGBoost classifier performed relatively well with an F1-score and AUC of 0.64 and 0.66 respectively.

We suspect that the inertial sensors are much more sensitive to acceleration change which leads to better classification performance for our particular problem. In Fig. 2 we see data peaks at a maximum of $\sim 2\mathrm{m/s^2}$ for hall and $\sim 13\mathrm{m/s^2}$ for inertial sensors (almost 7 times higher). We assume that with worse brakes, the recorded acceleration is not as pronounced compared to good brakes and is therefore easier to separate which is not the case with hall sensor data.

VI. Conclusion

In this paper we presented a novel idea for the predictive maintenance of bikes. We showed that we are able to distinguish between acute bad and good brake performance trips while riding a bike using acceleration data. We tested k-nearest neighbor (KNN), long short-term memory (LSTM) and XGBoost classifiers with two different means of measuring acceleration, i.e. inertial and hall sensors. Here, the classifiers trained with the inertial sensor data performed the highest, with KNN performing the strongest overall. In this test, we were able distinguish between the different classes (good/bad) after $1\mathrm{km}$ trips.

For future work, we would like to develop a method that uses unsupervised learning/clustering approaches. We are confident that this is the optimal way to detect progressive degradation in brake performance for individual cyclists. Currently, we consider KNN with dynamic time warping as our preferred method, which is also suitable for unsupervised learning.

We also need to consider extreme cases such as driving uphill or downhill. In such situations, the acceleration of the bike deviates too much from normal riding behavior. We consider normal riding behavior to be riding on roads that are not sloping. We plan to filter data where the orientation is significantly deviating from the normal flat orientation. In addition, we intend to evaluate our approach and future adjustments in a long-term study with several cyclists to verify our preliminary results.

VII. Acknowledgment

I would like to thank my supervisor Torben Weis for his advice and support, as well as my colleague Marian Waltereit for his help with these projects. This work has been partly funded by a BMBF grant for the project KI-LiveS.

References

[1] Sinus. (2019) Fahrrad-Monitor Deutschland 2019 - Ergebnisse einer repräsentativen Online-Befragung. [Online]. Available: https://www.bmvi.de/SharedDocs/DE/Anlage/K/fahrradmonitor-2019-ergebnisse.pdf

[2] F. Wolfgang and S. Martin, "Brake pad wear indicator, disc brake with such a display and brake shoes for such a disc brake," Patent DE102 010 048 988A1.

[3] Rumble: Mountain bike predictive analytics from your smartphone. [Online]. Available: https://www.ischool.berkeley.edu/projects/2019/rumble-mountain-bike-predictive-analytics-your-smartphone

[4] Predictive maintenance of bicycles. [Online]. Available: https://gregoirejan.github.io/project/maintenancebike/

[5] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, ser. AAAIWS'94. AAAI Press, 1994, p. 359–370.

[6] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, and E. Woods, "Tslearn, a machine learning toolkit for time series data," *Journal of Machine Learning Research*, vol. 21, no. 118, pp. 1–6, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-091.html

[7] M. Cuturi and M. Blondel, "Soft-dtw: a differentiable loss function for time-series," 2018.

[8] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *CoRR*, vol. abs/1603.02754, 2016. [Online]. Available: http://arxiv.org/abs/1603.02754

[9] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[10] ESP-IDF Programming Guide. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/

[11] W. Krzanowski and D. Hand, *ROC Curves for Continuous Data*, ser. Chapman & hall/CRC Monographs on Statistics & Applied Probability. CRC Press, 2009.