The Eleventh International Conference on
Engineering Computational Technology
23–25 August, 2022 | Montpellier, France

Proceedings of the Eleventh International Conference on
Engineering Computational Technology
Edited by B.H.V. Topping and P. Iványi
Civil-Comp Conferences, Volume 2, Paper 7.3
Civil-Comp Press, Edinburgh, United Kingdom, 2022, doi: 10.4203/ccc.2.7.3

Creation of a Data Pipeline to Determine the Filling Level of Storage Boxes

J. Kneifel, S. Chmielewski, R. Roj, R. Theiß and P. Dültgen

Forschungsgemeinschaft Werkzeuge und Werkstoffe e.V., Remscheid, Germany

Abstract

For the storage of small parts and bulk materials, companies usually use storage boxes, which, in addition to the required load capacity, must also be stackable and easily accessible. While a large number of variants are commercially available in the lower price segment, only few approaches exist that enable modern warehouse management. This work describes the equipment of storage boxes with a vibration-based measurement system to detect the filling level. This enables the functionality for intelligent filling level detection as well as for automated reordering of the respective goods.

From results of simulations as well as from initial findings of the investigations on a test rig, it was concluded that vibration excitation together with analysis of the natural frequencies is superior to the concept for investigating the decay behaviour. Thus, it was determined that FFT, PSD, and RMS approaches should be pursued.

It was found that different crate types, fill levels and filling materials led to a large variance in measurement results and thus to the differentiability of contents and fill level. The respective differences were recognizable in the measurement data, but due to the amount of variation, a manual evaluation was not performed. It was decided that an algorithm based on artificial intelligence should be applied. This work describes the details about the design of a data pipeline in order to process the data. Beside information about the utilized software tools, also details about the artificial intelligence methods as well as the constructive design of the boxes are provided.

Keywords: artificial intelligence, data acquisition, data pipeline, machine learning, oscillations, storage box.

1 Introduction

Storing a variety of items within a warehouse in an industrial context is usually done using plastic or metal boxes stacked on a shelf. Those kind of storage facilities rely mostly on manual inspection to find out the filling levels of boxes containing various bulk goods. A few examples for those fillings are screws, nuts, and bolts made of metal, plastic, or wood. Approaches to automate such tasks in inventory management and logistics utilize cameras and some type of image classification. The detection is therefore limited to boxes that sit on a shelf and those systems tend to be expensive. Usually, it requires extensive modification of the shelves and the storeroom, hence call for a high investment. It offers great inducement to develop a low-cost, easy to install and more versatile alternative.

To expand on the idea to be able to overview the status of inventory, a hub is set up to communicate between the boxes and the user and acts as a digital twin for the warehouse. The intention is to develop a smart box that works on any surface as long as it is within the reach of a local wireless area network and does not require any further modification of the facility. Another requirement is to only incorporate low-budget parts and make use of open-source software to further lower the estimated price and therefor the barrier of entry.

The detection is achieved by analysing the resonance response of the boxes from a stimulus given by a vibration motor that produces an oscillating force across a single axis (cf. Figure 1).

The pipelines in the development process are split between the training and the evaluation phase. After the initial model creation, the training parts will be deactivated and the system will be running in pure detection mode. It will do scheduled filling level detections in set time intervals and automatically present the updated values on a website giving the user the status of the boxes without the need to manually do inventory.

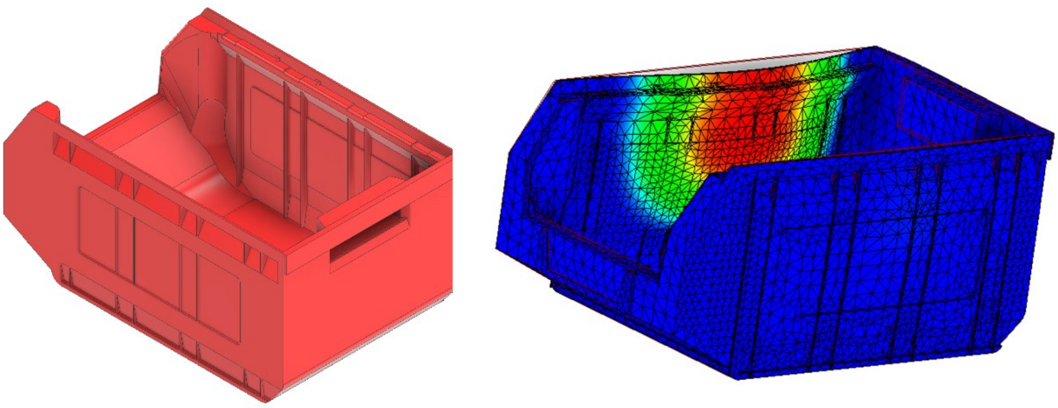

Figure 1: CAD-model of a typical storage box and FEM-model indicating oscillations

2 Methods

The entire project is realized in python and several library-add-ons, except for the code running on the microcontroller nodes, which requires C++. The vibration is picked up by a MEMS (microelectromechanical system) accelerometer and the sensor signal is transported to the hub with an Arduino compatible microcontroller. The raw data gets saved in the hub, which is a SoC (system on a chip) with a file storage, a database, and a webserver running all the data engineering, data science, machine learning, and data distribution pipelines as well as a framework to manage the inventory.

To achieve a scalable solution that will be robust and reliable in operation, but also easy to expand, the implemented pipelines to orchestrate data flow with Kedro were optimized [1]. Many operations are performed each time new inputs are fed into these data pipelines with the ability to remote control certain parameters via a CLI (command line input) and REST-API (representational state transfer application program interface). All data processing after model deployment is done in the hub, so if there will be updates to the algorithm, it can be uploaded from afar without the need to reprogram.

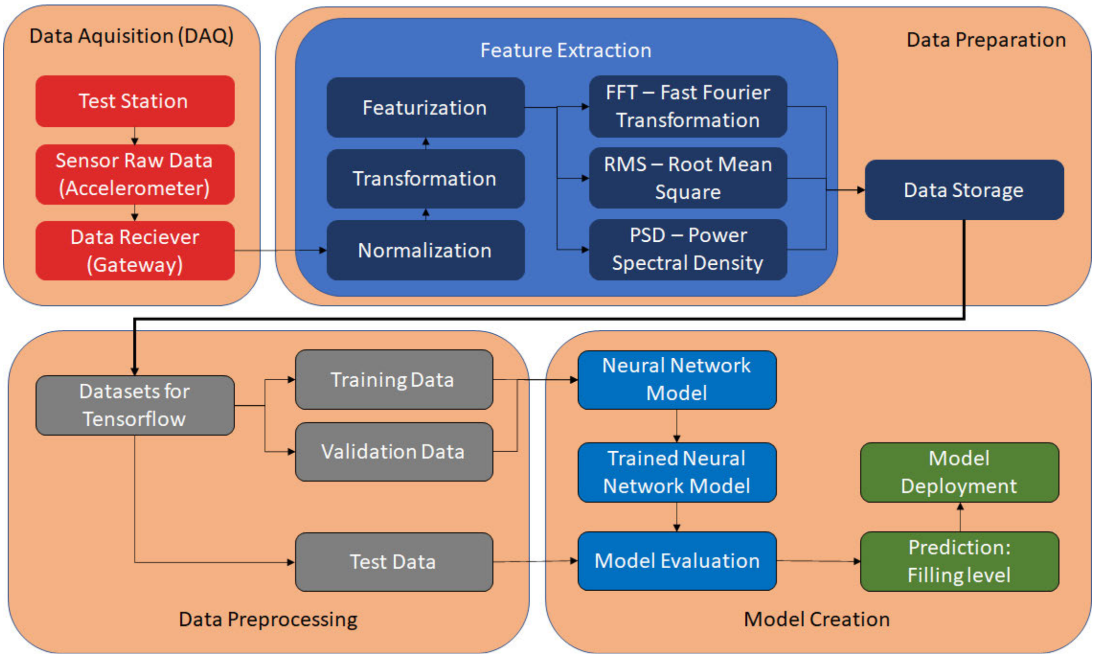

Figure 2: Pipeline for transforming and processing data for training machine learning model

The pipelines evolve around the data storage that also acts as a database for the inventory management system (cf. Figure 2). There is a training phase in which a dataset with fingerprints of different filling levels is collected to train a machine learning model and an evaluation phase that executes it after deployment. Both phases share the same data engineering code as it maintains the architecture, which is used for analytics and transforms the data into what is expected from the model. The dataset contains entries with a variety of filling materials and corresponding levels which was generated on a test bench over a long-term trial.

3 Results

At first, new sensor data is transmitted from one of the box nodes to the hub on which it enters the data engineering and data science pipeline. The raw data contains a table with timestamps, acceleration data, and, while in training phase, a label of the filling level saved to a CSV file. It gets parsed into a pandas data frame before any doubles and inconsistencies get filtered out [2,3], so that every datapoint consistently has the same length and sample rate. The featurization is handled by Scipy [4], giving a fingerprint in form of PSD (power spectral density), FFT (fast Fourier transform), and RMS (root mean square) data tables. While in training phase, the results get further preprocessed to be split into train, test, and validation data before entering the TensorFlow framework to create the neural network model [5]. The training is complete once it reports a sufficient accuracy. After deployment, the data gets imported directly into the model to estimate the inquired filling level, which is send to a database or an API from which the InvenTree inventory management system pulls the current values [6].

Initial measurement data suggested that the determination of the levels of any product can be tracked with sufficient accuracy using artificial intelligence. It is not necessary to mount actuators and sensors on opposite walls, and from a design point of view, integration on the underside of the storage box is associated with the least effort as well as sufficiently accurate results. The entire electronics can thus be integrated in a small area and do not have to be connected with cables over larger distances.

As a result, it was possible to determine that no major technical restrictions had to be observed in order to design a vibration-optimized storage box. In addition to the integration of the electrical components in a housing, the same properties as for conventional open fronted storage boxes, such as walls reinforced by ribbing or the application of a high load, had to be taken into account.

More detailed concepts for modular mounting of the measuring electronics were developed. Both modular retrofit-ability and large-scale production were considered. Screw-in, clip-in, and slide-in solutions were evaluated, with emphasis on contact pressure. This was important to ensure that there is no gap between the storage box and the measuring box and that the connection for vibration transmission is guaranteed.

4 Conclusions and Contributions

The solution solves a common task that requires constant attentiveness like making inventory to an automatic process running in the background. To realize this in a scalable manner while staying on a small budget, it requires the development of methods, like using accelerometers and vibration analysis for filling level detection as well as incorporation of modern technologies in data orchestration and machine learning.

The goal was to make inventory management smarter and easier to handle while keeping the costs low. The benefit of neural networks was used to simplify the detection process of filling levels making use of its black box characteristic, which proved to be outstanding in handling noisy signals and inconsistencies in datasets to a certain degree. Implementing data pipelines further streamlines the data engineering and data science part to a degree that it is manageable on a big scale, making this process reproduceable, expandable, and seamless. It shows that creating pipelines and a robust data infrastructure can be beneficial for both, large and small endeavours. Due to the nature of open-source code and the implementation of standardized protocols, it is possible to connect various endpoints with another, making it possible to utilize the always up-to-date inventory data and plan ahead in areas like stock and order management or logistics.

The project has hit three milestones: 1) the substitution of cameras to low-budget accelerometers (also advantageous for privacy compliance), 2) the detection of filling levels by interpreting the resonance response using machine learning, and 3) the orchestration of data flow through pipelines on a scalable platform (cf. Figure 3).
It shows that it is possible to automate inventory management for pourable bulk goods stored in boxes. To keep the costs as low as possible, it is advisable to handle the training of the machine learning model in the cloud and let the hub on the edge execute the predictions after deployment. Therefore, also lowering the technical requirements for the system is on premise. Pretrained models will be included in the hub so that it can be used for common goods immediately and only needs retraining in unforeseen cases that produce very inconsistent filling level estimates.

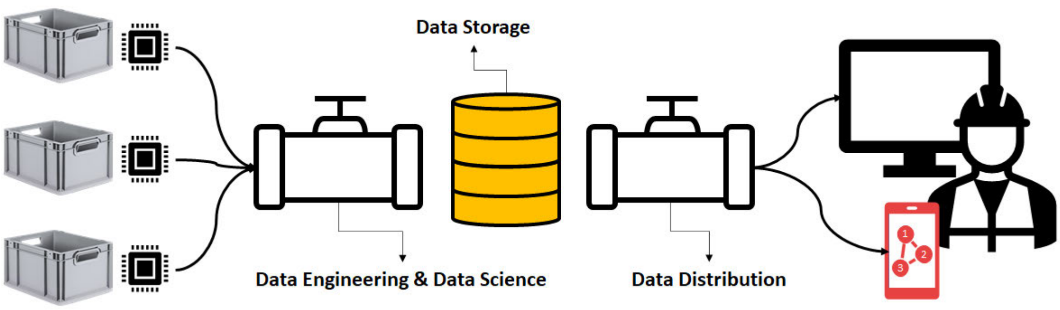

Figure 3: Graph of data pipeline for storage box deployment

Acknowledgements

The research is supported by the Federal Ministry of Economic Affairs and Climate Action the German Federation of Industrial Research Associations (AiF) with the funding numbers ZF4215311AB9 and ZF4744402AB9.

References

[1] S. Alam, L. Bàlan, G. Comym, Y. Dada, I. Danov, L. Hoang, R. Kanchwala, J. Klein, A. Milne, J. Schwarzmann, M. Theisen, S. Wong, “Kedro, Version 0.17.7”, https://github.com/kedro-org/kedro, Accessed 02 February 2022.

[2] W. McKinney, S. Van der Walt, J. Millmann, “Data Structures for Statistical Computing in Python”, 2020, 51-56. https://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf, Accessed 02 February 2022.

[3] The pandas development team: “Pandas, Version 1.2.4”, https://github.com/pandas-dev/pandas, Accessed 06 January 2021.

[4] V. Pauli, G. Ralf, E. Travis, Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. Van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.N. Archibald, A.H. Ribeiro, F. Pedregosa, P. Van Mulbregt, “SciPy 1.0 Contributors: Scipy, Version 1.6.3”, https://github.com/scipy/scipy, Accessed 20 January 2021.

[5] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems”, 2015, https://github.com/tensorflow/tensorflow, Accessed 01 February 2022.

[6] The InvenTree development team: “InvenTree, Version 0.6.0”, https://github.com/inventree/InvenTree, Accessed 06 January 2022.