Detecting Equipment Activities by Using Machine Learning Algorithms

A. Fischer, M. Liang, V. Orschlet, H. Bi, S. Kessler, J. Fottner

Chair of Materials Handling, Material Flow, Logistics, Department of Mechanical Engineering, Technical University of Munich, Germany (e-mail: anne.fischer@tum.de).

Abstract: Discrete-event simulation can serve as a tool for using equipment data to control processes and calculate alternative scenarios. For this purpose, the simulation requires knowledge of the process states on the construction site. One way using these process states automatically in the simulation is to interpret sensor data using machine learning. This work shows the procedure and the results for the application of machine learning to a practical example in the field of special civil engineering. Sensor data (features) and activity data (target values) are used for data acquisition. Data preprocessing is performed using the moving average method to suppress data noise; data outliers are filtered using the box method. Feature selection is based on statistical considerations and the SHAP values. A total of five supervised machine learning algorithms are used to classify the existing data: (1) Decision Tree, (2) Logistic Regression, (3) Support Vector Machine, (4) Naive Bayes, (5) Artificial Neural Networks. Confusion matrices, cross-validation, and learning curves are used to evaluate the algorithms. Overall, the paper shows that machine learning is very well suited to supporting the integration of current data into the simulation.

Keywords: Discrete event modeling and simulation, Production planning and control, Logistics in manufacturing, Process states detection, Activity detection, Classification in supervised machine learning.

1. INTRODUCTION

The number of digital technologies in the construction industry is steadily increasing, while the amount of productivity increase is significantly lower than in other sectors (McKinsey Global Institute, 2017). One component for improving this situation is the usage of discrete-event simulation (DES) as a proven tool for testing complex systems in advance and for validating variants with minimal effort (AbouRizk et al., 2011). DES thus offers the potential to discover unproductivity and initiate corrective actions (Tommelein, 1998).

Owing to the specifics of the construction industry, such as the unique character of construction projects and the segmentation within the individual construction processes, increased demands are placed on a simulation model (Fischer et al., 2020). Despite the advantages, especially in the prediction of uncertainties, DES has still not arrived in practice (AbouRizk, 2010; Leite et al., 2016; Abdelmegid et al., 2020). One reason is the lack of dynamic update of the simulation models with the real conditions on the construction site.

To counteract this, Fischer et al. (2020) introduce a dynamic feedback control system for construction progress data. The core of this system is the integration of equipment data into a simulation model via middleware. If one takes a look at the equipment data, one can see that it needs to be prepared for use in the simulation, e.g. by correctly matching live equipment data to the current process step. The equipment data refers to the data of a rotary drilling rig, which is generated during the production of large-diameter bored piles according to the most used drilling method, named Kelly drilling method. Until now, there exist no sensors attached either on the rotary drilling rig or on its accessories. Sensor problems occur, e.g., due to heavy loads or wide ranges while drilling. Thus, this use case requires a data-analytic method based on the equipment’s telematics data.

This study starts here and shows the application of machine learning to activity detection in equipment data. Machine learning is chosen because pile production is a highly complex field, which requires a lot of experience and expert knowledge. Simultaneously, it is characterized by a high degree of uncertainties and variations, involving a variety in process variability. It presents and evaluates five classification models for activity detection on the rotary drilling rig: Decision Tree (DT), Logistic Regression (LG), Support Vector Machine (SVM), Naive Bayes (NB), and Artificial Neural Networks (ANN).

2. RELATED WORK AND RESEARCH OBJECTIVE

The application of machine learning to activity detection in construction equipment is divided into sensor-based and image-based activity detection (Rashid and Louis, 2020). Ahn et al. (2015) and Akhavian and Behzadan (2015) install simple measurement sensors inside the driver’s cab.
Ahn et al. (2015) use low-cost accelerometers to measure acceleration in three directions. Akhavian and Behzadan (2015) use cell phones to record acceleration and position during earthworks. By identifying activities such as “Engine-off”, “Working” and “Idling”, conclusions can be drawn about the duration and productivity of the operation. When trying to detect three or more activities using machine learning, the accuracy deteriorated to below 90% (Akhavian and Behzadan, 2015). Ahn et al. (2015) use the following classification models: DT, NB, k-Nearest-Neighbor (KNN), and ANN. Instead of NB, Akhavian and Behzadan (2015) use LG and SVM.

Rashid and Louis (2020) indicate the high degree of influence of the cabin vibrations on the measurement results and hence on the data evaluation. By attaching acceleration and gyroscope sensors to the individual articulated structural parts of an excavator boom, they improve the activity detection of construction equipment with detailed recording of the movements. The classification models considered are DT, SVM, KNN, and ANN.

Golparvar-Fard et al. (2013) also track the movement of the excavator boom, but with the help of image recognition. They compare the supervised classification model SVM together with two unsupervised classification models (NB and pLSA (probabilistic Latent Semantic Analysis)). It turned out that the supervised classification model performed best.

All the investigations’ sensor data is based on accelerometers. Furthermore, the main subject is either a front-end loader (Akhavian and Behzadan, 2015) or a hydraulic excavator (Ahn et al., 2015; Rashid and Louis, 2020). Thus, there is a lack of analysis of the equipment’s own sensor data and transmission to other equipment types. Here is where the research objective arises: in special civil engineering with its rotary drilling rigs, it is hard to attach additional sensors to structural parts of the drilling rigs due to the enormous loads. Therefore, the approach presented here focuses on sensor data sent by the drilling rigs’ telematics. Specifically, rotary drilling rigs for use with the Kelly drilling method are considered. This method is the most commonly used method for the production of large-diameter bored piles. The data analysis using machine learning considers the drilling process more closely to use it in a dynamically updated simulation model. To identify the activities of the current process step during drilling, the following five supervised machine learning classification models are compared: DT, LG, SVM, NB, and ANN. Since SVM is selected, this paper treats not KNN, which also considers the spatial proximity of data points in space.

3. METHODOLOGY

The presented data analysis using machine learning follows Kim and Choi (2018); see Fig. 1.

The data to be analyzed is first recorded using the equipment’s sensors. Noise from this data is eliminated using suitable methods, also known as signal processing. Next, various features are extracted, and the appropriate features for the machine learning model are selected. Then supervised machine learning algorithms are applied and compared with each other to find the best one.

### 3.1 Data acquisition

In total, data is used from 16 large-diameter bored piles generated during the construction of a bridge project called “Westtangente Rosenheim”.

Table 1 shows the recorded sensor data of the equipment. In total, the data includes 20 features.

For the application of supervised classification models, the sensor data requires appropriate target values. Hence, the activity, as well as the corresponding start time, are determined manually, in parallel with the recording of the sensor data.

![Fig. 1. Procedure for data analysis using machine learning (Kim and Choi, 2018)](image)

| No. | Feature                   | Unit | Variance |
|-----|---------------------------|------|----------|
| 1   | Depth                     | m    | 105.602  |
| 2   | Torque of rotary drive    | kNm  | 12199.223|
| 3   | Speed of rotary drive     | rpm  | 112.154  |
| 4   | Main winch rope force     | t    | 18.609   |
| 5   | Crowd-force               | t    | 60.633   |
| 6   | Main winch rope speed     | cm/min | 249.353 |
| 7   | Pressure pump 1           | bar  | 14095.235|
| 8   | Pressure pump 2           | bar  | 13713.176|
| 9   | Pressure pump 3           | bar  | 4078.962 |
| 10  | Pressure pump 4           | bar  | 2262.839 |
| 11  | Torque of Kelly bar       | %    | 854.352  |
| 12  | Aux. winch rope force     | t    | 0.065    |
| 13  | Crowd depth               | m    | 0.001    |
| 14  | Casing length             | m    | 0.000    |
| 15  | Status rig                | %    | 0.000    |
| 16  | Main winch gear mode      | °    | 0.698    |
| 17  | Inclination X             | °    | 0.210    |
| 18  | Inclination Y             | °    | 0.432    |
| 19  | Boring threshold          | m    | 0.000    |
| 20  | Torque steps              |      | 0.342    |
The following five main activities are selected to break down the process of borehole drilling in a suitable way for the simulation later on: “Emptying” (0), “Lowering” (1), “Drilling” (2), “Pulling” (3), “Drilling in casing” (4). The numbers in brackets indicate the predicted labels of the target values in the following.

3.2 Signal processing

The sensor data acquired is first formed into input data for the supervised classification models, then filtered to exclude noise and outliers. Finally, it is scaled to exclude dependencies on units.

The input data for the classification models contain both the features and the corresponding target values. Therefore, each data point of the sensor data gets a target value, which represents the process steps for this data point. The timestamp links each data point of the sensor data and the corresponding process step from the activity data. After assigning the process steps to the sensor data, the data points correspond to the input data for the classification models. The five activities considered comprise a total of approximately 142,000 data points.

The sensor data, which serves as a set of input variables for the classification models, is influenced by several factors. According to Wang et al. (1995), one main influencing factor is the presence of noise. Noise can affect data collection and data preparation processes in data mining applications. Errors often occur as a result. To suppress noise, the moving-average method is applied to the sensor data. According to Köhler and Lorenz (2005), this method stands out compared to other noise suppression methods. This noise suppression is performed externally in the software Matlab using the Smoothdata() function.

Next, outliers for each feature of the considered data points are detected graphically and filtered for each process using box methods. Finally, the input data is scaled by standardization using the Scikit-Learn class StandardScaler().

3.3 Feature extraction

The sensor data used already contains 20 features; see Table 1. Thus, this study did not further investigate the extraction of new features from sensor data.

3.4 Feature selection

Feature selection is used to select the features that influence the specified target values. The relevance is selected using the variance filter method (Das and Cakmak, 2018) and the SHAP (SHapley Additive exPlanations) value method (Lundberg and Lee, 2017).

The variance method eliminates the features that are almost zero; see Table 1. Except for the values of “Inclination X” and “Inclination Y”, the ranking of the SHAP values confirms the results of the analysis of variance; see Fig. 2. By comparing the variance filter method as well as the SHAP value of different features, the optimal number of features to be selected is ten; see Table 1 Nos. 1–10.

3.5 Supervised classification models

After the feature selection, the input data consists of ten sensor data (features) aiming to detect five activities (target values). Different algorithms for supervised machine learning are used to generate classification models. However, many existing algorithms have been developed for specific applications and provide good classification models only for these applications. Under consideration of the amount of data, the number of features, the robustness against features’ collinearity, and the robustness against data noise, the following algorithms of supervised learning are used for the input data: (1) DT, (2) LR, (3) SVM, (4) NB, (5) ANN.

The prediction accuracy of a model can be very high for training data, but for new data, this value is low. This phenomenon is called the overfitting of a model according to Géron et al. (2020). To avoid overfitting, the training data is divided into a training data set and a validation data set before training a model to avoid overfitting. The validation data needs to be selected sufficiently well to ensure a high prediction quality, which can be determined with the test data set. So-called cross-validation helps to divide the training data set randomly into a certain number of subsets of equal size. These subsets are called folds. The folds are combined to form K different splits, whereas one subset of the fold is the validation data set, and the others are the training data set.

This work uses the Scikit-Learn class GridsearchCV() from Pedregosa et al. (2011) to select the best hyperparameter combination for training a final model with all the available data. The K-value is ten by default.

Except for NB, Table 2 shows the possible and best hyperparameter combination for the implemented algorithms.

4. EVALUATION

Cross-validation, learning curves, and confusion matrices are introduced to evaluate the used algorithms, which assess the resulting classification models.
### Table 2. Overview of the hyperparameter tuning according to Pedregosa et al. (2011)

| Hyperparameter                  | Set value range (optimum)        |
|---------------------------------|----------------------------------|
| **Decision Tree**               |                                 |
| Criterion                       | Gini, Entropy                    |
| Max depth                       | None, 10, 30, 50, 60, 100        |
| Min sample split                | 1, 10, 50, 100, 200, 300, 400   |
| Min samples leaf                | 1, 10, 100, 200, 300, 400, 500, 600 |
| Min impurity decrease           | 0, 0.1, 0.2, 0.5                 |
| **Logistic Regression**         |                                 |
| Regularization parameter        | 0.001, 0.01, 0.1, 1, 10, 100, 1000 |
| Penalty                         | L1, L2                           |
| Solver                          | Newton-cg, Liblinear, Sag        |
| Multiclass                      | Ovr, Multinomial                 |
| **Support Vector Machine**      |                                 |
| Regularization parameter        | 0.001, 0.01, 0.1, 1, 10, 100, 1000 |
| Kernel function                 | RBF, Sigmoid                     |
| Kernel coefficient              | 0.001, 0.01, 0.1, 1              |
| **Artificial Neural Networks**  |                                 |
| Batch size                      | 8                                |
| Epochs                          | 15                               |
| Loss function                   | Categorical cross-entropy        |
| Optimizer                       | Simple Adam                      |

#### 4.1 Cross-validation

The cross-validation of the classification models including their mean score is shown in Fig. 3.

Except for NB, all models show a mean score of over 90%. In particular, the sixth and ninth folds of the complete data show a relatively low accuracy as training and validation data, whereas the other folds show a high accuracy. ANN shows the highest discrepancy concerning the sixth folder. ANN is therefore sensitive to the allocation of the complete data as training and validation data.

In total, the SVM model shows the best results since the corresponding average accuracy of the cross-validation is the highest with a maximum of 92%.

#### 4.2 Learning curve

Learning curves show the training and validation accuracy of a classifier for a different number of training samples. Thus, it is possible to determine whether a model is overmatched or undermatched. According to Géron et al. (2020), overfitting means that the validation accuracy appears low if the training accuracy is very high. If underfitting, both of these accuracies are low. Learning curves can also check whether adding more training data is useful to achieve optimal model performance.

The learning curves of the classification models are shown in Fig. 4.

The results of the learning curves show that there is no overfitting or underfitting in the models of the algorithms. With a higher number of training samples, the model can better predict the validation data, and the validation accuracy increases. For the DT and SVM, the convergence to a value for training accuracy and validation accuracy is particularly pronounced. Here, the accuracy is also highest with the maximum number of training samples (DT: training accuracy of 94%, validation accuracy of 94%; SVM: training accuracy of 94%, validation accuracy of 93%). Fig. 4 shows the learning curve for ANN. From this figure, it can be seen that the training and validation accuracy does not converge to a value in set epochs. Therefore, using multiple dates or epochs for training is proven to be suitable.

#### 4.3 Confusion matrix

After the sensitivity of the models to the training data has been examined by means of cross-validation and learning curves, the accuracy of the models in relation to the test data is now evaluated with the help of confusion matrices.

![Fig. 3. Cross-validation for all models](image1)

![Fig. 4. Learning curves for all models](image2)
For this purpose, 20% of the total data is defined as the test data set. The confusion matrix determines the accuracy of the classification models for each process.

The confusion matrices of the classification models are shown in Fig. 5.

For the activity “Emptying” (0) DT achieves the highest accuracy with 93%. In comparison, it is minimal for NB with 85%. Likewise, for the activity “Training” (1), DT achieves the highest accuracy with 82% and NB the lowest accuracy with 58%. For the activity “Drilling” (2), the accuracies of all models are around 92%. For the activity “Pulling” (3), the accuracies of all models are higher than 95%. For the activity “Drilling in casing” (4), only NB has a low accuracy of 78%. All the other models achieve a high accuracy of at least 94%. Ultimately, they come to one conclusion: both the DT and SVM models demonstrate the best performance in the confusion matrix and, therefore, can consider being the best choice for our application.

5. DISCUSSION

After comparing the different metrics, the following statement is derived: DT appears to give the best results for process classification for the sensor data used. SVM can also have good accuracy for process classification. However, in practice, it is time intensive.

NB produces relatively low classification accuracy for real sensor data because the basic principle of this algorithm is based on ideal data with uncorrelated features.

In comparison to Ahn et al. (2015), the poor results with NB may indicate that the amount of data examined is too small. Ahn et al. (2015) show that the accuracy of NB increases significantly with a larger number of samples. Furthermore, ANN scores worse in this study than in the reviewed literature (Ahn et al., 2015; Akhavian and Behzadan, 2015; Rashid and Louis, 2020). The missing consideration of time, such as depth per time, is not covered by the selected hyperparameters from the Scikit-Learn library.

6. CONCLUSION

This paper shows an approach to how sensor data sent via telematics can be evaluated using machine learning. The subject of the investigation is a rotary drilling rig. The aim is to detect the state of the machine to be able to follow construction progress. At first, the data is collected, processed and the relevant features are selected. For the evaluation with machine learning, five supervised classification models are considered. The target values correspond to the process steps and thus to the activities during drilling: “Emptying”, “Lowering”, “Drilling”, “Pulling”, and “Drilling in casing”. Overall, satisfactory results are achieved with all five models used, with the DT and SVM models being superior in terms of learning performance and accuracy.

The data in this paper refers to a real case study. By breaking down the drilling process into five activities, the presented activity detection allows a more detailed prediction (e.g., per pile). Besides drilling, it is then interesting to extend this study to include all side-processes in the pile production process, such as reinforcement and concrete pouring. The selected algorithm will be applied to other construction sites in subsequent studies. In the future, we aim at investigating other influencing factors and including them in the activity detection approach. The influence of the geology or the identification of the geology based on the mask data is especially of interest for further investigations. The next step is to implement the evaluation of the sensor data using the algorithm as an application. A middleware automates the retrieval of sensor data and its forwarding to a suitable simulation model via TCP/IP protocols (Fischer et al., 2020). The implementation of algorithms as microservice can automatically assign activity data to the sensor data. This information then serves as input for the duration of the activities in the simulation model. Therefore, it is necessary to update the simulation model with data continuously (Fischer et al., 2021).

ACKNOWLEDGEMENTS

This research was funded by the German Federal Ministry of Education and Research within the research project “Building 4.0”. We also like to thank the project partners for providing the data.

REFERENCES

Abdelmegid, M.A., González, V.A., Poshdar, M., O’Sullivan, M., Walker, C.G., and Ying, F. (2020). Barriers to adopting simulation modelling in
construction industry. *Automation in Construction*, 111, 103046. doi:10.1016/j.autcon.2019.103046.

AbouRizk, S., Halpin, D., Mohamed, Y., and Hermann, U. (2011). Research in modeling and simulation for improving construction engineering operations. *Journal of Construction Engineering and Management*, 137(10), 843–852. doi:10.1061/(ASCE)CO.1943-7862.0000288.

AbouRizk, S. (2010). Role of simulation in construction engineering and management. *Journal of Construction Engineering and Management*, 136(10), 1140–1153. doi:10.1061/(ASCE)CO.1943-7862.0000220.

Ahn, C.R., Lee, S., and Peña-Mora, F. (2015). Application of low-cost accelerometers for measuring the operational efficiency of a construction equipment fleet. *Journal of Computing in Civil Engineering*, 29(2). doi:10.1061/(ASCE)CP.1943-5487.0000337.

Akhavian, R. and Behzadan, A.H. (2015). Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers. *Advanced Engineering Informatics*, 29(4), 867–877. doi:10.1016/j.aei.2015.03.001.

Das, S. and Cakmak, U. (2018). *Hands-On Automated Machine Learning: A beginner’s guide to building automated machine learning systems using automl and python*. PACKT Publishing Limited.

Fischer, A., Balakrishnan, G., Kessler, S., and Fottner, J. (2020). Begleitende Prozesssimulation für das Kellybohrverfahren [Accompanying process simulation for the Kelly drilling process]. In *8. Fachtagung Baumaschinenotechnik 2020*. Dresden, Germany.

Fischer, A., Li, Z., Wenzler, F., Kessler, S., and Fottner, J. (2021). Cyclic update of project scheduling by using equipment activity data. In *17th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2021)*. Budapest, Hungary.

Géron, A., Rother, K., and Demmig, T. (2020). *Prazisinstieg Machine Learning mit Scikit-Learn, Keras and TensorFlow: Konzepte, Tools und Techniken für intelligente Systeme [Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools and Techniques for Intelligent Systems]*, 2nd edition.

Golparvar-Fard, M., Heydarian, A., and Niebles, J.C. (2013). Vision-based action recognition of earthmoving equipment using spatio-temporal features and support vector machine classifiers. *Advanced Engineering Informatics*, 27(4), 652–663. doi:10.1016/j.aei.2013.09.001.

Kim, S. and Choi, J.H. (2018). Convolutional neural network for gear fault diagnosis based on signal segmentation approach. *Structural Health Monitoring*, 18. doi:10.1177/1475921718805683.

Köhler, T. and Lorenz, D. (2005). A comparison of denoising methods for one dimensional time series. In *Technical Report DFG SPP 1114*. Braunschweig, Germany.

Leite, F., Cho, Y., Behzadan, A.H., Lee, S., Choe, S., Fang, Y., Akhavian, R., and Hwang, S. (2016). Visualization, information modeling, and simulation: Grand challenges in the construction industry. *Journal of Computing in Civil Engineering*, 30(6), 04016035. doi:10.1061/(ASCE)CP.1943-5487.0000604.

Lundberg, S.M. and Lee, S.I. (2017). A unified approach to interpreting model predictions. In *31st Conference on Neural Information Processing System (NIPS 2017)*, 4765–4774. Long Beach, CA, USA.