Multilevel fault diagnostics for railway applications using limited historical data

Osarenren Kennedy Aimiyekagbon[1], Alexander Löwen[2], Raphael Hanselle[3], Thomas Rief [4], Maximilian Beck[5], Walter Sextro[6]

[1,2,6] *Paderborn University, Faculty of Mechanical Engineering, Chair of Dynamics and Mechatronics, Warburger Str. 100, 33098 Paderborn, Germany*
osarenren.aimiyekagbon@uni-paderborn.de

[3] *Ostwestfalen-Lippe University of Applied Sciences and Arts, Institute for Energy Research (iFE), 32657 Lemgo, Germany*

[4,5] *Wölfel Engineering GmbH, Höchberg, Germany*

ABSTRACT

This study proposes a fault diagnostics methodology that addresses the challenges posed by highly imbalanced datasets typical of railway applications, where faulty conditions constitute the minority class. Fault diagnostics is performed from the component level upward, considering each sensor's proximity to its respective critical component. Advanced signal analysis, feature engineering, and automated data-driven model generation techniques were explored to achieve comprehensive diagnostics, such that the model development process accounts for variations in the operating conditions and differing levels of information availability. The proposed methodology is evaluated on datasets from the MONOCAB, for scenarios with limited faulty instances and on the Beijing 2024 IEEE PHM Conference data challenge, which focused on fault diagnostics of railway systems under various fault modes and operating conditions.

1. INTRODUCTION

Advancements in technology have contributed to closing the transportation gap between urban and rural areas, exemplified by the development of the MONOCAB. The MONOCAB is conceptualized as a compact, on-demand autonomous gyro-stabilized monorail vehicle for the reactivation of rail routes in rural regions and for the establishment of seamless connectivity with urban rail networks (Griese & Schulte, 2025; Hanselle, Griese, Rasche, & Schulte, 2023). Its advantage lies in the narrow width of the vehicle in combination with its stabilization system, such that the vehicle is able to drive bidirectionally on a single railway track. Operating at Grade of Automation 4, the MONOCAB functions without onboard personnel, such as attendants or drivers. Therefore, to meet stringent requirements for reliability and safety, the MONOCAB bundles sensors, actuators and information processing units in a complex mechatronic system that supports autonomous localization, communication, perception, object and obstacle detection, intelligent control, condition monitoring, and predictive maintenance.

Currently, two technology demonstrators of the MONOCAB exist. Although several onboard sensors are integrated into the system, they primarily provide data representing healthy instances. This presents several challenges in developing diagnostic and predictive models, such as data imbalance and limited generalization capability. This study specifically addresses the challenges posed by highly imbalanced datasets typical of railway applications, where faulty conditions constitute the minority class. It focuses on leveraging condition monitoring data to facilitate early fault detection, isolation, and identification, thereby sustaining the availability, reliability, and safety of autonomous railway applications.

2. THEORETICAL FRAMEWORK

Rolling stocks typically operate under normal healthy conditions over long periods of time, while faulty conditions often rarely occur. Furthermore, although vehicle manufacturers might instrument condition monitoring systems on newly developed rolling stocks, the acquired data might comprise only of healthy instances and is not available to train operators. Particularly, in the field of data-driven diagnostics, such data imbalance poses a challenge. Although several techniques, such as undersampling, oversampling have been proposed, they are not directly applicable under cases, where only limited healthy data is available.

The flowchart in Figure 1 presents the framework for data-driven diagnostics, under consideration of three levels of data

imbalance. In the first level, only healthy instances at distinct operating conditions (OCs) are available. In the second level, healthy instances at several distinct OCs and limited faulty instances at a single OC are available. Finally, in the third level, healthy and faulty instances at several distinct OCs are available. The several steps within the framework are described in the following.

Identify key signals

One of the first steps to consider before implementing a condition monitoring system, according to the ISO 17359 standard (ISO 17359:2018(en), 2018), is a failure modes and effects analysis (FMEA) to identify possible fault modes and their influence on critical components of a system. Subsequently, appropriate measurement techniques are selected to monitor the fault modes, if they are measurable via physical parameters (ISO 17359:2018(en), 2018). In complex systems, like rolling stock, with multiple interconnected components, it is advisable to place appropriate sensors close to potential sources of faults, so as to capture the fault signature or symptoms before been masked or attenuated by other neighboring components. Thus, for effective fault detection of critical components, it is essential to prioritize sensor signals that are in close proximity to these components.

Explore / analyze data

The data exploration and analysis step is essential to uncover signal deviations from pristine condition, even under limited data or distinct OCs. This step comprises data visualization and two steps of the Prognostics and Health Management (PHM) cycle (Clements, 2011), namely data pre-processing and feature extraction. In general, data, and time series in particular, can be analyzed directly in the time domain, or transformed into the frequency domain to reveal amplitude variations at characteristic frequencies or to detect frequency shifts, due to changes in the internal properties of a system, such as stiffness or damping. Additionally, visualizing data in the time-frequency domain can uncover local changes in frequency or amplitude over time (Aimiyekagbon, Bender, Hemsel, & Sextro, 2024). Data pre-processing typically involves data denoising and data cleansing, such as outlier removal and filling missing numbers. After the optional data pre-processing, features can then be extracted in the time-, frequency- and time-frequency domains. Several toolboxes, such as highly comparative time-series analysis (HCTSA) (Fulcher, Little, & Jones, 2013; Fulcher & Jones, 2017), electromyography (EMG) feature extraction toolbox (Too, Abdullah, & Saad, 2019; Too, Abdullah, Mohd Saad, & Tee, 2019), time series feature extraction on basis of scalable hypothesis tests (tsfresh) (Christ, Braun, Neuffer, & Kempa-Liehr, 2018), and flexible time series processing & feature extraction (tsflex) (Van Der Donckt, Van Der Donckt, Deprost, & Van Hoecke, 2022) can automate the feature extraction process. Feature selection is often conducted after feature extraction to pre-select relevant features to facilitate efficient training of machine learning algorithms.

When data regarding different fault modes and underlying OCs is scarce, then the need for expert knowledge becomes more significant. For instance, in the first level, where only healthy data is available, establishing a threshold for healthy conditions requires expert insight. This is because changes in OCs can affect condition monitoring signals, making a healthy threshold or features valid in one OC potentially unsuitable in another. As more information about the different fault modes and underlying OCs become available, they should be incorporated in the data exploration and analysis to improve fault detection and subsequently fault isolation and identification.

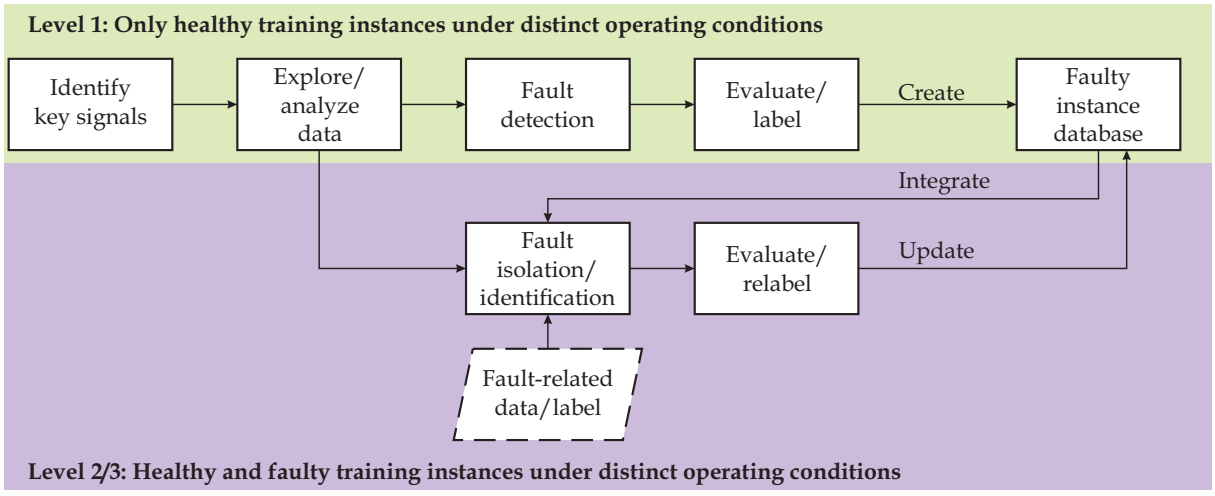


Figure 1. Framework for fault diagnostics under consideration of limited faulty instances

Fault detection, isolation and identification

Fault detection is essentially a binary decision of determining whether a fault exists or not. Fault isolation, on the other hand, involves distinguishing among multiple fault modes, making it a multi-class classification problem. Additionally, identifying a fault through quantifying its size is akin to solving a regression problem. (Isermann, 1994) Consequently, supervised, semi-supervised, or unsupervised machine learning techniques can be employed to address these challenges.

In cases where only healthy instances are available, unsupervised techniques for anomaly detection, such as autoencoder, one-class support vector machine (SVM) and isolation forest are applicable. However, as previously noted, expert insight may be necessary for improved performance. For instance, experts may be required to select distinctive features, when determining the reconstruction error threshold for autoencoders, or throughout the algorithm training process in general. In cases where faulty instances corresponding to some or all anticipated fault modes are available, they should be included with their respective labels during the training phase of semi-supervised and supervised techniques, respectively.

Model performance evaluation

Evaluating model performance is a crucial step in developing diagnostic models. For fault identification, regression metrics like root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) are used, whereas fault isolation is assessed using classification metrics such as recall, F1 score, and the Matthews correlation coefficient, which are obtained from the confusion matrix comparing predicted and actual classes (Luque, Carrasco, Martín, & de Las Heras, 2019). However, because previously unseen fault modes may be misclassified, expert evaluation is also necessary. Additionally, when models are trained solely on healthy data, any detected anomaly can be accurately attributed to a specific fault mode and subsequently labeled by experts.

Faulty instance database

At the first level, only healthy data is available. Thus, the faulty instance database is initially empty. As mentioned earlier, once an anomaly is accurately associated with a particular fault mode, it is labeled and stored in a faulty instance database. As a result, the task shifts from anomaly detection to fault isolation, with the database, and eventually the model, being updated incrementally as each new fault mode gets correctly identified. In the case of the second and third levels, the faulty instance database and eventually the model are also updated, when previously unseen and unlabeled fault modes are precisely linked to a particular fault mode and eventually labeled.

3. APPLICATION EXAMPLES

Two application examples are considered to demonstrate the effectiveness of the proposed framework. The first application focuses on the MONOCAB dataset, which contains only healthy data. This scenario emphasizes the framework's ability to operate under limited and challenging data conditions. The second application involves the Beijing Jiaotong University (BJTU) bogie dataset provided for the PHM-Beijing 2024 Data Challenge (Ding et al., 2024). This dataset includes both healthy and faulty instances, allowing for a comprehensive evaluation of the framework's performance in a more conventional setting.

3.1. MONOCAB dataset

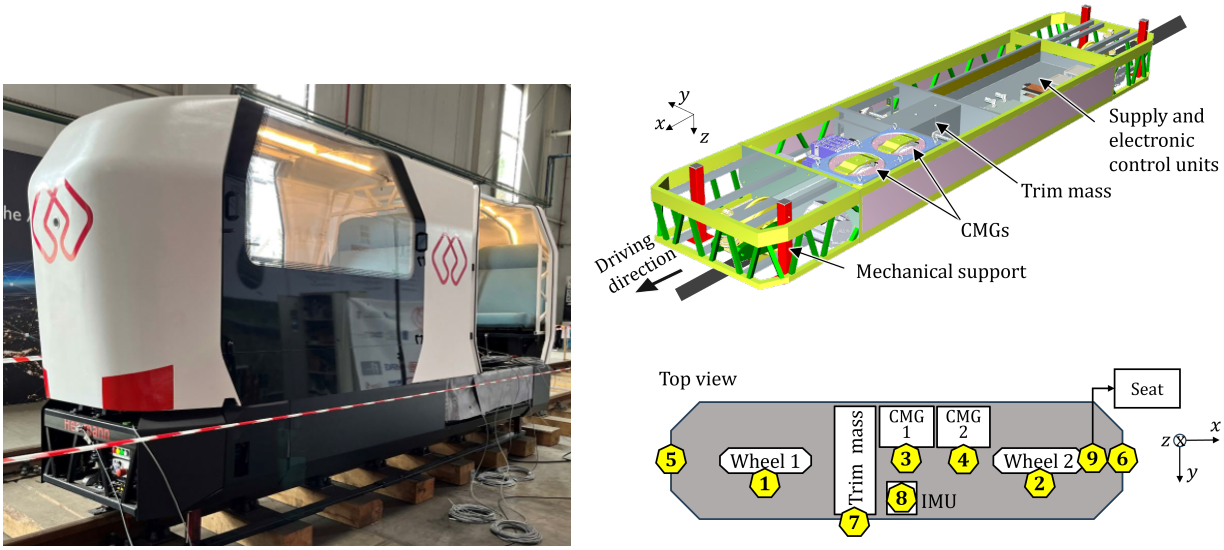
In the following, the different steps of the proposed framework will be elaborated using the MONOCAB dataset as a case study.

3.1.1. Identify key signals

In the context of developing a condition monitoring system for the MONOCAB, an FMEA is carried out to reveal possible failures. A concise description of the FMEA of various components is presented in Table 7, in the appendix. Unlike conventional rail vehicles, the MONOCAB runs on one rail with the aid of a comprehensive stabilization system. As seen in top sub-figure of Figure 2, two counter-rotating control moment gyroscopes (CMGs) counteract dynamic disturbances on the rolling axis of the vehicle, while a laterally movable trim mass counteract stationary disturbances. Finally, a mechanical support system is meant to keep the vehicle upright during emergencies or halt. The wheelset of the MONOCAB is constructed differently, because it drives on one rail. However, since similar failure modes also occur for conventional train wheelsets, the FMEA is similar. Thus, in addition to the stabilization system comprising of the CMGs and trim mass based on the previous works (Rezvanizaniani, Valibeigloo, Asghari, Barabady, & Kumar, 2008; Müller, 2009; Szkoda & Satora, 2019; Pang, Dai, & Chen, 2024), the wheelset, brake and suspension are considered very important.

The next step after the FMEA is the selection of appropriate sensors for continuous monitoring of the critical components. There are several onboard sensors on the MONOCAB used to control and monitor the condition of the vehicle. The translational accelerations, rotational angles and rotation rates are measured using two redundant inertial measurement unit (IMU) sensors mounted on the frame of the MONOCAB near the CMGs. They are used to determine the current roll angle for the stabilization control and also the translational acceleration of the vehicle for the longitudinal control.

The vehicles velocity and relative position are measured using two inductive proximity sensors, located at the front and rear



(a) (b)

Figure 2: (a) MONOCAB in the Dörentrup test site, and corresponding (b) main components description adapted from (Griese et al., 2023; Griese & Schulte, 2025) and sensor placement of the accelerometers

wheel shaft of the vehicle. They are pointing at the sprocket on the wheel shaft to detect if a tooth is present or not. Based on these discrete signals, the velocity is calculated. The relative position is a basic integration of the velocity value. The direction of the velocity is based on the driving stage of the vehicle. Both values are needed for the longitudinal control of the MONOCAB.

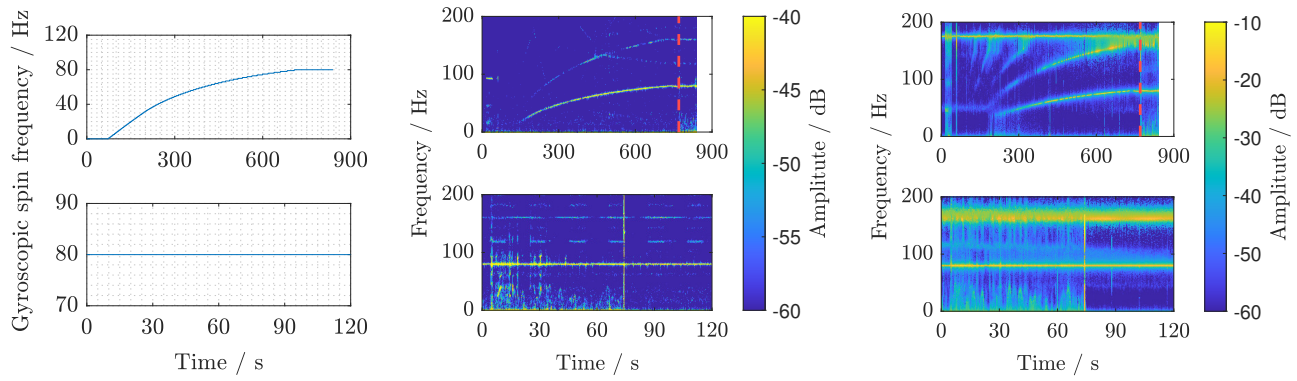
Furthermore, the motor current and power consumption are continuously monitored using the frequency inverter of the drive motors, eliminating the need for additional external sensors. This enables fault diagnostics of the motor itself as well as neighboring components.

Vibration sensors are particularly suitable for detecting failure modes in mechanical components at an early stage (Bernal, Spiryagin, & Cole, 2018). They are capable of capturing even subtle changes in a component's vibration behavior, which may be caused by material defects such as cracks or plastic deformation. Specifically, cracks and axle deformation result in local changes in stiffness, which manifest as shifts in resonance frequencies or the appearance of additional vibration modes. Thus, external accelerometers are mounted on critical components of the MONOCAB as further sensors.

3.1.2. Explore / analyze data

To investigate the dynamic behavior of the MONOCAB's stabilization system, a comprehensive measurement campaign was conducted at the Dörentrup test site. The measurement setup included nine ±6 g triaxial accelerometers, mounted magnetically at key structural and functional components of the MONOCAB. As seen in the bottom sub-figure of Figure 2, the sensor positions were strategically selected, such that sensors 1 and 2 were mounted on the wheel suspension, sensors 3 and 4 on the gyroscopic stabilization units, sensors 5 and 6 on the main frame, sensor 7 on the sliding mass bearing (integral to the stabilization mechanism), sensor 8 adjacent to the Inertial Measurement Unit (IMU), and sensor 9 beneath a seat, representing passenger exposure. All sensors were connected to a central data acquisition unit housed in a measurement case, powered via a 230 V power supply, and after synchronized with the MONOCAB's onboard data acquisition system for comparative analysis.

Four experimental measurement data, namely M01 through M04 are considered for the MONOCAB dataset. They consist of sensor measurements collected under different operational settings. The measurement data M01 and M04 were collected during the run-up phase of the CMGs, while measurement data M02 and M03 were collected after reaching the precessional speed of the CMGs and during the shutdown of the trim mass stabilization system, in the sense that the control system stops sending actuation signals, causing no further displacement from that point onward. The sampling frequency of the internal and external sensor signals considered in this study are 2 kHz and 5 kHz, respectively. The gyroscopic spin frequency is depicted in Figure 3(a) for measurement data M01 and M04. As seen in the top sub-figure of Figure 3(a), the CMGs reach the precessional speed of 4800 rpm, that is 80 Hz after approximately 12 min. Furthermore, as seen in the bottom sub-figure of Figure 3(a) for measurement data M02, the CMGs remain at the precessional speed of 4800 rpm during operation. The corresponding time-frequency domain analysis results of the onboard accelerometer of IMU rear sensor



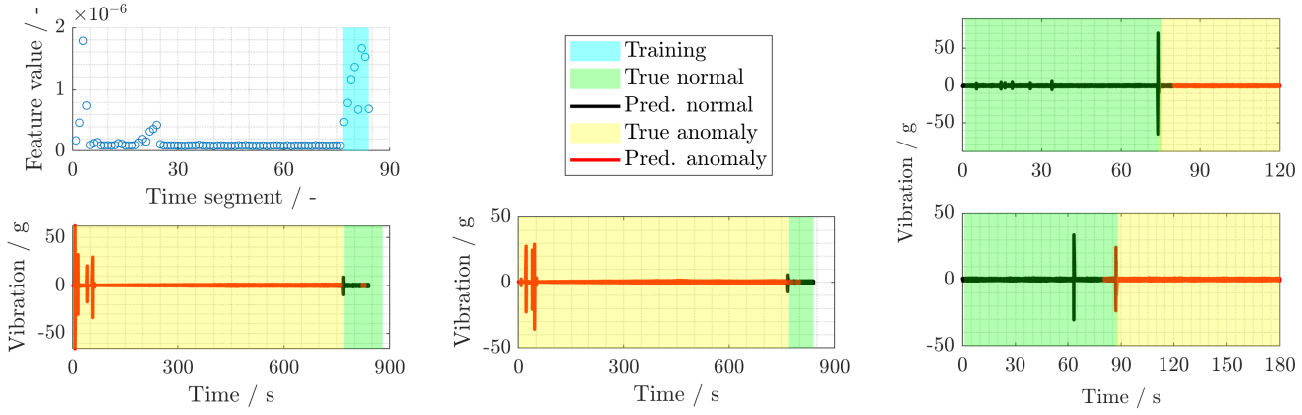
(a) (b) (c)

Figure 3: Comparative analysis of experiments M01 (top) and M02 (bottom) of the MONOCAB (a) Gyroscopic spin frequency in the time domain (b) Time-frequency domain analysis of the internal accelerometer of IMU rear sensor in the y-direction (c) Time-frequency domain analysis of external accelerometer at position 7 in the y-direction

and the external accelerometer at position 7 along the y-axis are presented in Figures 3(b) and 3(c), respectively. As seen in the top sub-figures, the harmonics of the spin frequency are dominant. Furthermore, after reaching the precessional speed, as indicated by the dashed red line, the intensity of the amplitudes at distinct frequencies and especially at frequencies below 100 Hz. The dominant amplitudes at distinct frequencies, resulting from the active trim mass stabilization, are seen better in the bottom sub-figures of Figures 3(b) and 3(c). After the shutdown of the trim mass stabilization at about 75 s, the intensity of the amplitudes at the distinct frequencies also clearly decreases.

3.1.3. Fault detection, isolation and identification

Based on the findings from the preceding section, four statistical features, namely the variance, standard deviation, range and interquartile range were exemplarily extracted each from the amplitudes in the frequency range [10; 40] Hz of 10 s measurement segments. The variance is exemplarily shown in the top sub-figure of Figure 4(a) for the measurement data M01. As seen in the figure, the variance is comparatively higher when the trim mass stabilization is active, except for the initial segments with increased variance. Thus, the features derived in the time segment during the active trim mass stabilization were utilized to exemplarily train a one-class SVM. The resulting prediction result is mapped to the underlying vibration signal and shown in the bottom sub-figure of Figure 4(a), where an anomaly represents an intentional or unintentional



(a) (b) (c)

Figure 4: Fault detection based on one-class SVM ultilizing (a) features derived from 10 s time segments for experiment M01 (top). Corresponding predicted anomaly for the experiment M01 (bottom), (b) Predicted anomaly for experiment M04, (c) Predicted anomaly for experiments M02 (top) and M03 (bottom)

inactive trim mass stabilization. As seen in the Figure, the one-class SVM model is capable of detecting an inactive trim mass stabilization. The prediction results obtained from the same model are presented in Figures 4(b) and 4(c), where the trim mass stabilization was inactive before and after reaching the precessional speed, respectively.

3.1.4. Model performance evaluation

The confusion matrix is employed to evaluate the model performance, where class 0 represents an active trim mass stabilization, while class 1 represents an intentional or unintentional inactive trim mass stabilization. Due to safety-critical reasons, an undetected inactive trim mass stabilization is more dire. Hence, the precision and recall, corresponding to the second-row value of the first column on the far right and the first-row value of the second column at the bottom of the plots, respectively, are particularly emphasized. As seen in Figure 5, the model performs good with at least a precision of 80% and at least a recall of 90% over all considered measurement cases. Although the detected anomaly can be pinpointed to the inactive trim mass stabilization, further amount of data, also with sufficient faulty instances are required for a comprehensive fault isolation and identification for this presented case study. As more faulty instance become available, they can be incorporated in the faulty instance database to address data imbalance and improve the performance of the diagnostic models.

3.2. Beijing Jiaotong University (BJTU) bogie dataset

The BJTU bogie dataset is comprised of experimental data collected at different health states and OCs of several critical components of a real subway bogie (Ding et al., 2024). In the following, the distinct steps of the proposed framework will be elaborated upon.

3.2.1. Identify key signals

As considered in the foregoing application example, previous studies (Rezvanizaniani et al., 2008; Müller, 2009; Szkoda & Satora, 2019) have investigated the FMEA of a railway bogie. Furthermore, it is assumed that the sensors, as listed in Table 1 have been optimally selected and positioned close to the component of interest. Thus, although the effect of a fault in one component could be reflected in neighboring components, the sensor with the closest path to a probable faulty component should be prioritized, under consideration of the fault mode. For example, for the diagnostics of a motor short circuit fault, the motor current signals should be analyzed.

Channel	Component	Signal Type
1 - 3	Motor (drive end)	Tri-axial ACC
4 - 6	Motor (fan end)	Tri-axial ACC
7 - 9	Motor (cable)	3-phase current
10 - 12	Gearbox (input axle)	Tri-axial ACC
13 - 15	Gearbox (output axle)	Tri-axial ACC
16 - 18	Axle box left (end cover)	Tri-axial ACC
19 - 21	Axle box right (end cover)	Tri-axial ACC

Table 1. Overview of sensor channels (Ding et al., 2024)

3.2.2. Explore / analyze data

As seen in Table 2, three motor speed and three lateral load settings, which simulate different train speeds and the exerted forces as a train moves through a curve, are considered as OCs in the dataset (Ding et al., 2024). Furthermore, as listed in Table 2, at a constant OC and a certain health state, sensor signals

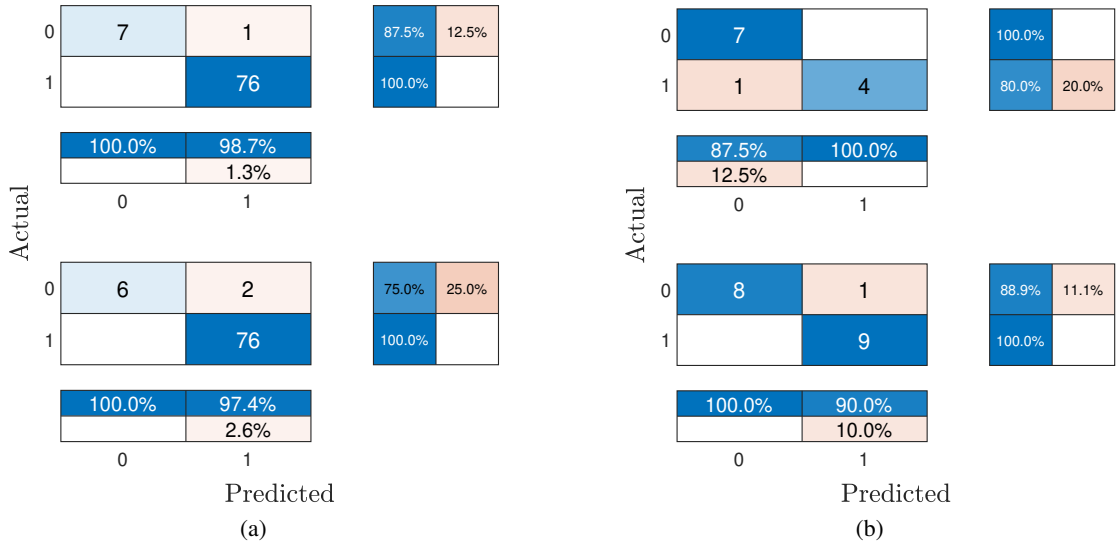


Figure 5: Confusion matrix for (a) experiments M01 (top) and M04 (bottom), and (b) M02 (top) and M03 (bottom)

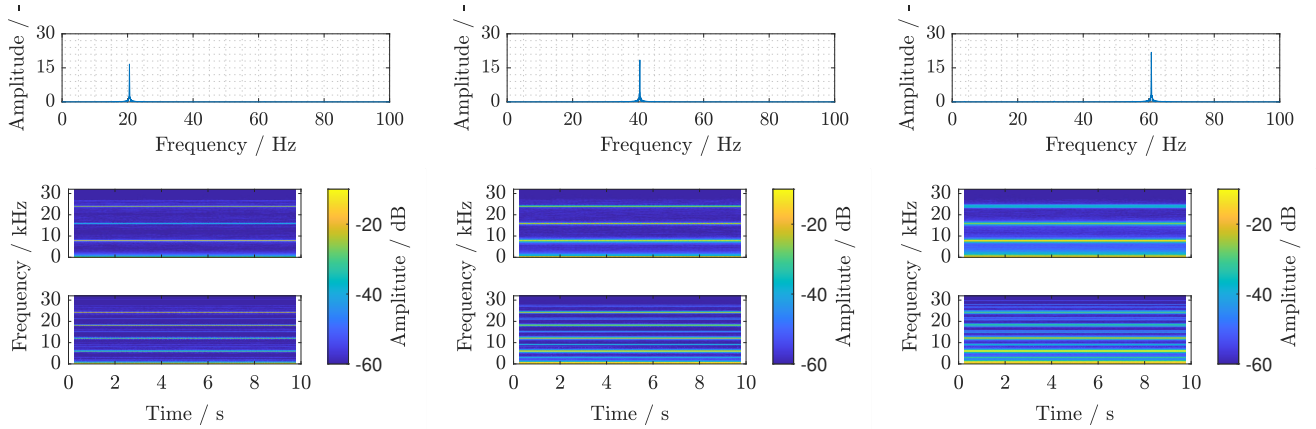


Figure 6: Frequency (top) and time-frequency domain analysis (bottom) of one of the three-phase current (channel 7) for the motor rotating frequency of (a) 20 Hz, (b) 40 Hz and (c) 60 Hz

OC	Motor speed / Hz	Lateral load / kN
1	20	0
2	20	10
3	20	-10
4	40	0
5	40	10
6	40	-10
7	60	0
8	60	10
9	60	-10

Table 2. Overview of OCs (Ding et al., 2024)

from the 21 channels were acquired at a sampling frequency of 64 kHz. One of the three-phase current, namely channel 7, is exemplarily analyzed in the frequency- and time-frequency domain for the data indices $1, 18, 35, 2, 19$ and 36, that is for the normal condition and for motor short circuit. As seen in the first row of Figure 6 for the data indices $1, 18$ and 35, the distinct motor rotating frequencies are visible in the frequency spectrum. As also seen in the second row of Figure 6, for the data indices $1, 18$ and 35, resonance and the corresponding harmonics occur at approximately, 8 kHz, 16 kHz and 24 kHz, irrespective of the rotating frequency, but the intensity increases with increasing rotating frequency. However, as seen in the third row of Figure 6, in the event of a motor fault, that is for data indices $2, 19$ and 36, sub-harmonics occur at approximately 3 kHz, 6 kHz, 9 kHz and so on. Thus, features can be extracted specifically from the sub-harmonics to diagnose a motor fault. However, to minimize the manual effort and associated costs of analyzing individual sensor signals for each component, the following subsections present an automated, data-driven modeling approach.

3.2.3. Fault detection, isolation and identification

To automatically detect faults based on the healthy instances in the training data set, a Long Short-Term Memory Autoencoder (LSTM AE) is employed, which is already applied successfully to time series anomaly detection, such as in (Wei et al., 2023), able in capturing long-term relationships in sequential data. The LSTM AE is integrated into a toolbox for facilitated generation of data-driven models which ensures standardized interfaces between data processing steps, model re-usability, and simplified deployment (Löwen, Quirin, Hesse, & Aimiyekagbon, 2025). The LSTM AE is using all 21 sensors as inputs, simulating a scenario where no prior knowledge is used to generate a model for fault detection as automated as possible. The input is followed by hidden layers with dimensions 16 and 8 in the encoder, symmetrically mirrored in the decoder. The data is compressed into a latent representation of size 8, thereby facilitating the effective learning of temporal dependencies inherent in the data, and reconstructed. The reconstruction error is calculated as the mean squared error (MSE) through the signals j between the data x and the reconstruction $\hat{x}$ at each time i as depicted in Equation 1.

$$\text{MSE}_i = \frac{1}{21}\sum_{j=1}^{21}(x_{i,j} - \hat{x}_{i,j})^2 \tag{1}$$

The three-sigma rule is applied to automatically determine the threshold on the reconstruction error to detect anomalies within a measurement, see Equation 2, as it is frequently and successfully employed, such as in (Nicholaus, Park, Jung, Lee, & Kang, 2021). For each measurement, the detected anomalies are aggregated and expressed as a percentage. Very crucial is to set another threshold to identify a measurement as anomalous or not. To automate this decision, the percentage

			Training data index			Test data index					
Label	**Fault Component**	**Description** \ **OC**	3	6	9	1	2	4	5	7	8
N0	All components	Normal condition	1	18	35	52	69	86	103	120	137
M1	Motor	Short circuit	2	19	36	53	70	87	104	121	138
M2		Broken rotor bar	3	20	37	54	71	88	105	122	139
M3		Bearing fault	4	21	38	55	72	89	106	123	140
M4		Bowed rotor	5	22	39	56	73	90	107	124	141
G1	Gearbox	Gear cracked tooth	6	23	40	57	74	91	108	125	142
G2		Gear worn tooth	7	24	41	58	75	92	109	126	143
G3		Gear missing tooth	8	25	42	59	76	93	110	127	144
G4		Gear chipped tooth	9	26	43	60	77	94	111	128	145
G5		Inner race fault	10	27	44	61	78	95	112	129	146
G6		Outer race fault	11	28	45	62	79	96	113	130	147
G7		Rolling element fault	12	29	46	63	80	97	114	131	148
G8		Bearing cage fault	13	30	47	64	81	98	115	132	149
LA1	Left Axle Box	Inner race fault	14	31	48	65	82	99	116	133	150
LA2		Outer race fault	15	32	49	66	83	100	117	134	151
LA3		Rolling element fault	16	33	50	67	84	101	118	135	152
LA4		Bearing cage fault	17	34	51	68	85	102	119	136	153

Table 3. Training and test data indices with respective OCs and fault labels (Ding et al., 2024)

is considered as the model's confidence. This means if more than 50 % of the data points in a measurement are identified as anomalies, the measurement itself is classified as such

$$\text{Threshold} = \mu(\text{MSE}) + 3 \cdot \sigma(\text{MSE}). \tag{2}$$

If data regarding the faults to be recognized is provided, a classifier can be trained to recognize them, e.g. after a LSTM AE detects one. The toolbox that has already been applied to a classification problem can be used again for this purpose (Löwen et al., 2025). The data processing steps employed there can be reapplied, which consists of feature extraction, selection, standardization, and model training.

3.2.4. Model performance evaluation

To set up LSTM AE, the knowledge about the motor speed is used such that the OC regarding the motor speeds are considered separately, i.e., motor speeds of 20 Hz (OC 1, 2, 3), 40 Hz (OC 4, 5, 6), and 60 Hz (OC 7, 8 ,9). This means that a separate model is constructed for each motor speed and trained on training data 1, 18 and 35, respectively. A normalization process is fitted to adjust the input signals of the training data to the range [0,1], which is then separately applied on the test data of OC 1 and 2 (20 Hz), 4 and 5 (40 Hz), 7 and 8 (60 Hz). The Table 4 provides an overview of the hyperparameters utilized for the LSTM AE.

Figure 7 shows exemplary the percentages related to the data sets from OC 1, 2 and 3. There, most of the considered faults show an anomaly percentage of over 70 %. Similar results shows the LSTM AE trained on data 18. The LSTM AE trained on data 35 shows a clear but less distinct separation.

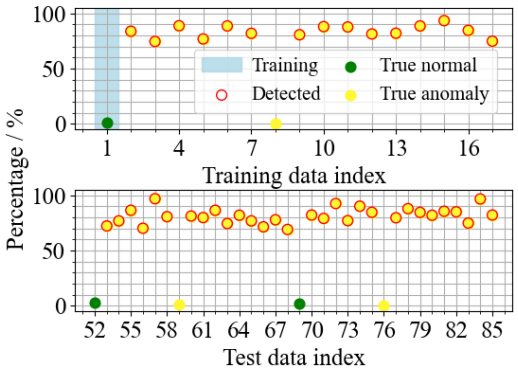


Figure 7. Percentages of detected anomalies in the data. Training on healthy condition OC 3. Test on OC 1, 2 and 3

Applying the percentage threshold at 50 % as mentioned before, the accuracy, precision and recall of each LSTM AE achieved are shown in Table 5, where an anomaly is considered as the positive class. Since there are only two healthy instances in each test data sets, the negative predictive value is very low if e.g. three faults are not classified as such, which is the case for the LSTM AE considered in Figure 7. However, the results show that the models can detect the majority of faults across different OCs with the condition that the motor

Hyperparameter	Value
Batch size	64
Number of epochs	1000
Layer configuration	21-16-8-16-21
Learning rate	0.01

Table 4. Utilized hyperparameters for the LSTM AE.

OC	Accuracy	Precision	Recall
OC 1, 2, 3	94.0 %	100.0 %	93.8 %
OC 4, 5, 6	94.0 %	100.0 %	93.8 %
OC 7, 8, 9	74.0 %	100.0 %	72.9 %

Table 5. Accuracy, Recall and Precision of the LSTM AE on the test data. A measurement is classified as faulty when more than 50 % of anomalies in it are detected.

speed is known. If more data were available for training including faulty data, the results could potentially be further improved and generalized through hyperparameter optimization and cross-validation, which is already integrated in the aforementioned toolbox (Löwen et al., 2025).

As mentioned before, a classifier can be trained to learn known faults to isolate and identify recurring faults, if the corresponding data is available, where the training data, data index 1 to 51, is treated as such. The following data processing steps orient on them employed in (Löwen et al., 2025). Firstly, to extract features, the HCTSA toolbox is applied. Subsequently, features which contain constant values or contain entries such as not a number (NaN) or infinite values are removed. If such entries appear in the test data, NaN values will be replaced by 0, while infinite values will be substituted with the largest representable number. To select the most significant features, analysis of variance is applied, to select the 50 most informative features, considering that highly correlated features are sorted out. Since the number of samples per fault class is very limited, a comparatively less complex algorithm, the k-nearest neighbors classifier (KNN) from scikit-learn (Pedregosa et al., 2011), is employed. Hyperparameter are optimized within a cross-validation procedure using scikit-optimize (Head, Kumar, Nahrstaedt, Louppe, & Shcherbatyi, 2021), where a StratifiedKFold strategy ensures that class distributions are preserved across the splits. The resulting values for various hyperparameters are listet in Table 6.

Hyperparameter	Value
Distance metric	Manhattan
Number of neighbors	4
Searching algorithm	Brute-force
Weight function	distance

Table 6. Optimized hyperparameters for the KNN algorithm.

The result is shown in Figure 8, where an F1 score of 0.79 is reached. Notable is, that the model has difficulties in the recognition of faults G2, G4 and G6. However, in cases of misclassifications, the instances are usually assigned to the correct component. When focusing on predictions at the component level, an F1 score of 0.97 is achieved.

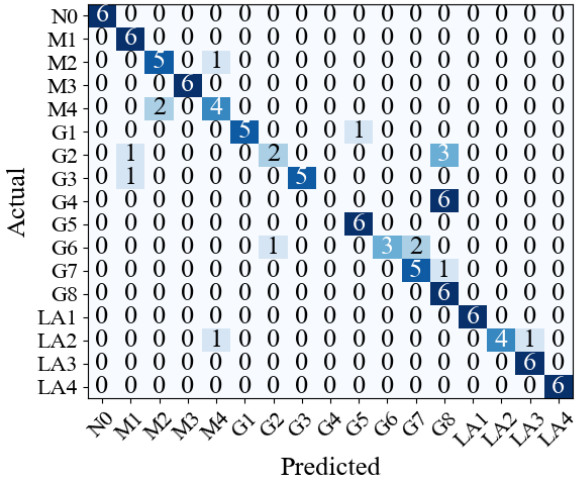


Figure 8. Confusion matrix for predicted classes of test data sets 52 to 153

4. Conclusion

The current study underlines the potential of advanced signal analysis, feature engineering, and automated data-driven model generation techniques for comprehensive diagnostics, even in scenarios with limited faulty instances. The methodology was validated using datasets from the MONOCAB and the Beijing 2024 IEEE PHM conference data challenge, which focused on fault diagnostics of a railway bogie under distinct constant operating conditions and different fault modes. The findings of this study are in line with current research, indicating that as the level of automation increases in data-driven model generation, the amount of required information, for training, testing and validation, also increases. However, models trained solely on healthy instances proved sufficiently effective for fault detection and expert-informed identification of faulty components. These findings underscore the potential of the presented approach for deployment in prototype systems or technical systems where faulty instances are scarce or unavailable, such as in the case of the MONOCAB. Although only evaluated on a relatively small dataset, the proposed methodology is applicable to larger datasets or real-time monitoring scenarios, depending on data quality. However, the expert-informed identification is identified as the bottleneck during scalability.

ACKNOWLEDGMENT

This research was accomplished within the research and development project **enableATO** funded by the Federal Ministry for Digital and Transport in Germany (BMDV).

The demonstrator monorail vehicle MONOCAB used for the measurements was developed within the framework of the research project MONOCAB OWL funded by the European Regional Development Fund (ERDF) and the Ministry of Transport of the state of North Rhine-Westphalia (Germany).

We extend our sincere appreciation to Daniel Reger for conducting the failure modes and effects analysis (FMEA) of the MONOCAB as part of his master's thesis.

REFERENCES

Aimiyekagbon, O. K., Bender, A., Hemsel, T., & Sextro, W. (2024). Diagnostics of piezoelectric bending actuators subjected to varying operating conditions. *Electronics*, *13*(3), 521.

Bernal, E., Spiryagin, M., & Cole, C. (2018). Onboard condition monitoring sensors, systems and techniques for freight railway vehicles: a review. *IEEE sensors journal*, *19*(1), 4–24.

Christ, M., Braun, N., Neuffer, J., & Kempa-Liehr, A. W. (2018). Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). *Neurocomputing*, *307*, 72–77.

Clements, N. S. (2011). Introduction to prognostics. In *tutorial, proceedings of the annual conference of the prognostics and health management society.*

Ding, A., Qin, Y., Wang, B., Guo, L., Jia, L., & Cheng, X. (2024). Evolvable graph neural network for system-level incremental fault diagnosis of train transmission systems. *Mechanical Systems and Signal Processing*, *210*, 111175.

Fulcher, B. D., & Jones, N. S. (2017). hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. *Cell systems*, *5*(5), 527–531.

Fulcher, B. D., Little, M. A., & Jones, N. S. (2013). Highly comparative time-series analysis: the empirical structure of time series and their methods. *Journal of the Royal Society Interface*, *10*(83), 20130048.

Griese, M., Döding, P., & Schulte, T. (2023). Analysis of mechanical eigenmodes of a self-stabilizing monorail vehicle. In *The iavsd international symposium on dynamics of vehicles on roads and tracks* (pp. 107–116).

Griese, M., & Schulte, T. (2025). Gyroscopic effects in the structural dynamics of monorail vehicles. *Vehicle System Dynamics*, 1–22.

Hanselle, R., Griese, M., Rasche, R., & Schulte, T. (2023). Hil simulation of the positioning control for an automated driving monorail vehicle. In *2023 ieee 21st international conference on industrial informatics (indin)* (pp. 1–6).

Head, T., Kumar, M., Nahrstaedt, H., Louppe, G., & Shcherbatyi, I. (2021, October). *scikit-optimize: Sequential model-based optimization in python.* `https://zenodo.org/records/5565057`. Zenodo. (Last accessed: April 29, 2025)

Isermann, R. (1994). Integration of fault detection and diagnosis methods. *IFAC Proceedings Volumes*, *27*(5), 575–590.

ISO 17359:2018(en). (2018). *Condition monitoring and diagnostics of machines - General guidelines* (Standard). International Organization for Standardization (ISO), Geneva, Switzerland.

Löwen, A., Quirin, D., Hesse, M., & Aimiyekagbon, O. (2025). Facilitating the automated generation of data-driven models for the diagnostics and prognostics of technical systems. In *2025 ieee 30th international conference on emerging technologies and factory automation (etfa).* (accepted for publication)

Luque, A., Carrasco, A., Martín, A., & de Las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, *91*, 216–231.

Müller, T. (2009). *Integration von verlässlichkeitsanalysen und-konzepten innerhalb der entwicklungsmethodik mechatronischer systeme* (Unpublished doctoral dissertation). Paderborn, Universität, Diss., 2009.

Nicholaus, I. T., Park, J. R., Jung, K., Lee, J. S., & Kang, D.-K. (2021, oct). Anomaly detection of water level using deep autoencoder. *Sensors (Basel)*, *21*(19), 6679. doi: 10.3390/s21196679

Pang, J., Dai, J., & Chen, Y. (2024). A fmea optimization method based on sm-ewm and vikor method in intuitionistic fuzzy environment for risk assessment of electromagnetic brake. In *2024 4th international conference on electrical engineering and mechatronics technology (iceemt)* (pp. 54–59).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Rezvanizaniani, S., Valibeigloo, M., Asghari, M., Barabady, J., & Kumar, U. (2008). Reliability centered maintenance for rolling stock: A case study in coaches' wheel sets of passenger trains of iranian railway. In *2008 ieee international conference on industrial engineering and engineering management* (pp. 516–520).

Szkoda, M., & Satora, M. (2019). The application of failure mode and effects analysis (fmea) for the risk assessment of changes in the maintenance system of railway vehicles. *Technical Transactions*, *2019*(Year 2019 (116)), 159–172.

Too, J., Abdullah, A. R., Mohd Saad, N., & Tee, W. (2019). Emg feature selection and classification using a pbest-guide binary particle swarm optimization. *Computation*, *7*(1), 12.

Too, J., Abdullah, A. R., & Saad, N. M. (2019). Classification of hand movements based on discrete wavelet transform and enhanced feature extraction. *International Journal of Advanced Computer Science and Applications*, *10*(6), 83–89.

Van Der Donckt, J., Van Der Donckt, J., Deprost, E., & Van Hoecke, S. (2022). tsflex: Flexible time series processing & feature extraction. *SoftwareX*, *17*, 100971.

Wei, Y., Jang-Jaccard, J., Xu, W., Sabrina, F., Camtepe, S., & Boulic, M. (2023). Lstm-autoencoder-based anomaly detection for indoor air quality time-series data. *IEEE Sensors Journal*, *23*(4), 3787-3800. doi: 10.1109/JSEN.2022.3230361

Main component	Sub-component	Failure mode	Cause	Effect
Wheelset	Wheel profile	Crack formation	Material fatigue, Overload, Wear, Corrosion, Overheating	Safety risks, Derailment
		Flat spots on wheel	Emergency braking, Slipping	Increased noise and vibrations, Reduced ride comfort
	Axle	Crack formation	Material fatigue, Wear, Corrosion	Safety risks
		Breakage	Overload, Ongoing crack formation, Excessive deformation	Safety risks, Derailment
		Deformation	Uneven load, Overload	Safety risks
	Bearing	Overheating	Insufficient lubrication, Contamination	Increased wear
		Increased noise and vibrations	Wear	Noisy operation, Reduced ride comfort
		Bearing damage	Wear, Overload	Safety risks, Axle blockage
Brake	Electromagnet	Extreme electromagnetic iron gap	Wear of electromagnetic components	Insufficient braking force
		Electromagnetic interference	Excessive electromagnetism in working environment	Brake malfunction, Safety risks
		Electromagnetic Overheat	Excessive coil current	Coil may burn out, Brake malfunction, Safety risks
	Spring	Spring failure	Fatigue, Wear, Corrosion	Safety risks, Brake malfunction
	Brake control	Electronic control failure	Software defect	Uncontrolled braking
	Emergency brake system	No release	Sensor fault, Software error	Collision risk
		Delayed activation	Electronic disturbance, Delay	Increased braking distance
		Sensor malfunction	Sensor failure, Environmental effects	Incorrect or missing brake reaction
Suspension	Spring	Spring breakage	Wear, Overload, Corrosion	Reduced ride quality, Vibrations
		Deformation	Excessive loading	Same as above
	Damper	Leakage	Wear, Corrosion, Fatigue	Reduced damping, More vibrations and wear
Gearbox	Frame / Chassis	Cracking / breakage	Wear, Corrosion, Cyclic loads	No stability
	Gear	Gear tooth breakage	Material fatigue, Overload, Corrosion	Power loss, Load on other teeth
		Tooth wear	Poor lubrication, Wear, Corrosion	Power loss, Noise
	Bearing	Overheating or damage	Poor lubrication, Wear	Increased wear, Power loss
Motor	Housing	Leakage	Material fatigue, Corrosion, Wear	Oil loss, Overheating
	Rotor	Short circuit, Cracks	Insulation fault, Fatigue, Wear	Power loss, Overheating
		Winding damage	Fatigue, Overload	Power loss, Overheating
	Stator	Winding damage	Fatigue, Overload	Power loss, Overheating
Control moment gyroscopes (CMGs)	Bearing	Bearing damage	Insufficient lubricant, Wear, Overload	Overheating, Increased wear, Failure, Derailment
	Motor	Motor damage	Wear, Material fatigue	Failure, Derailment
	Toothed belt	Crack	Wear	Uncontrolled deflection
		Poor tooth contact with gears	Fatigue	Belt skipping, Reduced driving comfort
Trim mass	Threaded rod	Crack, breakage	Fatigue, Overload	Derailment, Misalignment
		Deformation	Uneven load, Overload	Safety risk
		High sliding effort	Wear	Late alignment control

Table 7. Failure modes and effects analysis of the MONOCAB