Unsupervised representation learning for time series anomaly detection and beyond

Ferreira Correia; L.

Chapter 4

Datasets

Datasets play a fundamental role in the benchmarking aspect of research. Together with evaluation metrics, they allow for the quantification of a scientific contribution to the field and give readers and practitioners context to how any given approach performs. This chapter introduces the two main datasets used throughout this thesis, along with details on their generation and collection procedure and a preliminary analysis highlighting some key properties.

4.1 Simulation-based Publicly Available Dataset

As thoroughly discussed in Chapter 3.1, the most popular data sets suffer from several issues [31, 33]. While a few datasets were published that have attempted to address the identified issues, they solely represent continuous-sequence problems. Some real-world applications, however, involve discrete sequences. These applications might have patterns in the test data that do not appear in the training data, yet are still considered nominal. Research should aim to provide solutions for real-world problems, though if advances in the research field are disconnected from real-world settings, proposed solutions are of limited value. On the other hand, creating high-quality datasets from real-world problems can be difficult and costly, and hence there is a clear opportunity for generated but real-world inspired datasets to provide a cost-effective way to complete the anomaly detection dataset landscape [33]. This chapter details the generation of a non-trivial, and high-quality discrete-sequence dataset consisting of multivariate time series for online anomaly detection, named the *powertrain anomaly time series benchmark (PATH)* dataset. While primarily aimed at unsupervised anomaly detection, versions for semi-supervised anomaly detection, and time series generation and forecasting are provided as well. Despite the data being generated using simulation, the underlying electric vehicle simulation model is strongly motivated by the real world and is therefore complex and variable-state. The variable-state nature requires anomaly detectors to make the distinction between behaviour changes due to different states, like the high-voltage battery (HVB) state of charge (SoC) and behaviour changes due to an anomaly, further complicating detection.

4.1.1 Simulation Model

To create an extensive and diverse data set, the use of a physically inspired model from which data can be generated via simulation is proposed. MathWorks offers reference models for a variety of dynamic systems, one of which is the full electric vehicle (FEV) model [96] from the powertrain blockset in MathWorks Simulink. This choice is based on the author's familiarity with the domain, as generating data without any background knowledge may lead to systematic errors. The FEV model offered by MathWorks describes longitudinal dynamics and consists of six main subsystems: the drive cycle block, the driver block, the environment block, the controllers block and the vehicle block. The topology of the FEV model is illustrated in Figure 4.1.

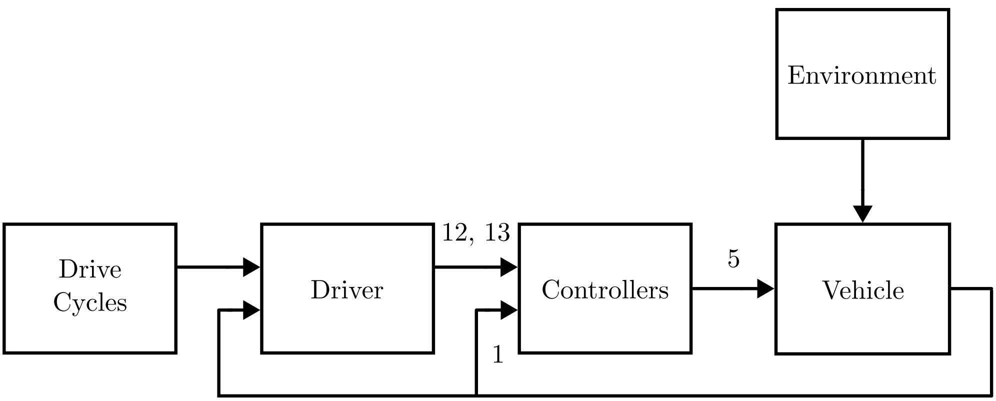

Figure 4.1: Simplified schematic of the FEV model used for the generation of the PATH data set. The numbers represent the indices of signal flow, whose reference is shown in Table 4.1. Only a subset of the signals in Table 4.1 are shown here, as the rest are signals exist exclusively within the vehicle block.

To represent system behaviour, $d_{\mathcal{D}} = 16$ signals are chosen to be logged during simulation and are summarised in Table 4.1. These signals are chosen based on do-

Table 4.1: Signals included in the PATH dataset, along with their physical units and persistent indices. For brevity, electric motor (EM) is abbreviated to EM, henceforth.

Index	Signal Name	Unit	Index	Signal Name	Unit
1	EM Angular Speed	rad/s	9	Axle Force Rear	N
2	EM Torque	N m	10	Axle Speed Front	rad/s
3	Axle Torque Front	N m	11	Axle Speed Rear	rad/s
4	Axle Torque Rear	N m	12	Accelerator Pedal	-
5	HVB SoC	%	13	Brake Pedal	-
6	HVB Current	A	14	HVB Temperature	°C
7	HVB Power	W	15	Cooling Pump Power	W
8	Axle Force Front	N	16	Refrigerator Power	W

main knowledge, with the goal of picking the features that are most representative of powertrain behaviour.

The drive cycle block of the FEV model defines the target vehicle speed and features a series of real-world drive cycles, i.e. profiles depicting the target vehicle speed as a function of time. From the list of speed profiles available in the original FEV model [96], a subset of them is eliminated due to their unrealistic nature, e.g. linearity, as they can be exclusively described using linear functions, or duplicity, as many cycles are present as sub-sequences in others. For example, the FTP72 drive cycle is present within FTP75 and the LA92Short drive cycle is present within LA92. In addition to that, drive cycles aimed at heavy vehicles, like trucks or buses, are also eliminated. The resulting subset of drive cycles chosen for simulation contains 33 different speed profiles of varying length, shown in Table 4.2, along with their lengths in seconds. Some drive cycles may be designed for specific types of powertrains such as diesel ones, but given that they only represent vehicle speed profiles, there is little reason why they cannot be driven by a vehicle with an electric powertrain, like the one modelled in this case.

The driver block of the FEV model regulates the dynamic system to maintain the target speed. Its inputs are the target vehicle speed and the actual vehicle speed, and its outputs are the acceleration and deceleration control commands, index 12 and 13 in Table 4.1, respectively, which are fed into the controllers block of the FEV model. This block takes said accelerator and brake pedal commands stemming as well as vehicle states like actual vehicle speed, EM angular speed, and HVB signals to calculate request signals for the powertrain, like the required EM torque and the brake signal, as well as battery management system signals like the HVB SoC, index

Table 4.2: Drive cycles used for the PATH data set generation, along with their respective lengths in seconds.

Drive Cycle	Length [s]
FTP75	2474
US06	600
SC03	600
HWFET	765
NYCC	598
HUDDS	1060
LA92	1435
IM240	240
UDDS	1369
WLTP Class 1	1022
WLTP Class 2	1477
WLTP Class 3	1800
Artemis Urban	993
Artemis Rural Road	1082
Artemis Motorway 130 kmph	1068
Artemis Motorway 150 kmph	1068
JC08	1204
JC08 Hot	1376
World Harmonized Vehicle Cycle (WHVC)	900
Braunschweig City Driving Cycle	1740
RTS 95	886
ETC FIGE Version 4	1800
CUEDC Petrol cycle	499
CUEDC SPC240 cycle	240
CUEDC diesel cycle - MC	1723
CUEDC diesel cycle - NA	1795
CUEDC diesel cycle - NB	1706
CUEDC diesel cycle - ME	1678
CUEDC diesel cycle - NC	1797
CUEDC diesel cycle - NCH	1676
China Light-Duty Vehicle Test Cycle for Passenger Cars	1800
China Light-Duty Vehicle Test Cycle for Commercial Vehicles	1800
China Worldwide Transient Vehicle Cycle	1799

5 in Table 4.1. Electric vehicles are capable of regenerative braking, meaning, the EM is used to decelerate the vehicle by acting as a generator, thereby charging the HVB if it is not already fully charged.

Following is the vehicle block of the FEV model, which outputs how the vehicle reacts to any inputs and contains the electric plant subsystem and the drivetrain subsystem. Both take inputs from the controllers block, including the HVB SoC, and the environment block, as well as from each other, as shown in Figure 4.2. The

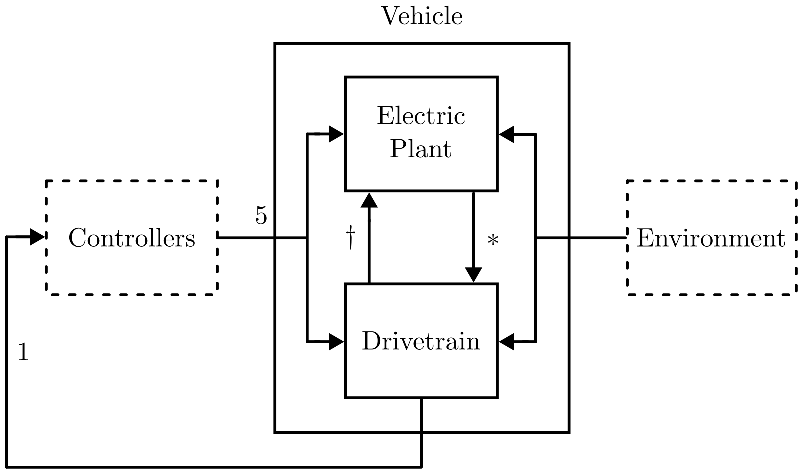

Figure 4.2: A more detailed schematic of the vehicle model depicted in Figure 4.1. Numbers represent the indices of signal flow, reference is shown in Table 4.1. For visual clarity, $*$ is used to represent signals with indices 2, 6, 7, 14, 15 and 16, while † signals with indices 1, 3, 4, 8, 9, 10, 11. Output signals of the vehicle model, which are not fed back into other subsystems, are not shown, for simplicity.

electric plant subsystem outputs the EM torque, the HVB current and power, the HVB temperature and the cooling pump and refrigerator powers, which correspond to indices 2, 6, 7, 14, 15 and 16 in Table 4.1, respectively. The EM torque is input into the drivetrain subsystem, which in turn outputs the EM angular speed, front and rear axle torques, forces and speeds, corresponding to indices 1, 3, 4, 8, 9, 10, 11 in Table 4.1, respectively. The EM angular speed is also fed back into the electric plant model, completing the control loop. Both subsystems also contain further subsystems within them which uncover the causal relationships between their respective signals, but diving as deep as the lowest abstraction level of the model is outside the scope of this thesis. By default, the HVB model features a static temperature model, however, to increase system complexity, a dynamic temperature model is added to the FEV model. The dynamic temperature model used is the electric vehicle HVB cooling system model [97], also from MathWorks.

The environment block of the FEV model dictates environmental conditions that affect the longitudinal dynamics of the FEV model. Parameters like atmospheric pressure, wind speed, road gradient and coefficient of friction can be modified within this subsystem.

By default, the signals are logged at a sampling frequency of 10 Hz and the solver used is the differential algebraic equations solver for Simscape (daessc). Physical simulations may encounter numerical under- and overflow, which slow down simulations drastically. To avoid simulations getting "stuck", a timeout counter of one hour is set in place to skip the current simulation if triggered. Simulation time depends on the length of the drive cycle, however, for the computer hardware used this is never longer than 20 minutes, and hence one hour is considered sufficient for simulations that do not stall.

Note that, unlike typical internal combustion engine powertrains, this powertrain does not feature discrete gear states, which would decouple the EM angular speed from the axle angular speeds, though a gearbox could be added in future versions of this powertrain for increased complexity. Additionally, further EMs could be added to the model to further increase the complexity of the simulation model.

Readers interested in more detail can refer to the full MathWorks Simulink model available in the GitHub repository under `github.com/lcs-crr/Thesis`. The entire PATH dataset in both its raw and ready-to-use forms can be found in the Zenodo repository under `zenodo.org/records/13255120`.

4.1.2 Data Set Generation

To generate a data set that is not only extensive but also diverse, 100 simulations are undertaken for each of the 33 drive cycles, each with random initial HVB temperatures and HVB SoCs. All simulations are run on a workstation equipped with an Intel Xeon Gold 6234 CPU running Windows 10 Enterprise LTSC version 21H2 with MATLAB 2023b. At this stage, all model properties are left as default, and hence all simulation results are considered *nominal*. For each simulation, the two states (HVB temperatures and HVB SoCs.) are sampled from uniform distributions $\mathcal{U}(10\,°\mathrm{C}, 30\,°\mathrm{C})$ and $\mathcal{U}(10\,\%, 100\,\%)$, respectively, to ensure no further bias is introduced. This also eliminates and reduces the effectiveness of simple threshold and control chart methods, since the HVB temperature and SoC, but also, by extension and to a lesser extent, other channels, exhibit nominal but high deviation from the average behaviour. As a precautionary measure, drive cycles with a minimum SoC value lower than 5 % are removed, as very low values are observed to result in abnormal behaviour. To add further complexity and to reflect real-world properties, *post-processing* is applied to all time series in this dataset. First, the beginning of time series is trimmed by random number of time steps so that time series representing the same drive cycle are rarely in sync. The amount by which a given time series is trimmed is sampled from uniform distribution $\mathcal{U}(0, 0.1T)$. This artefact can happen in the real world and means that, for the same drive cycle, any given time step is not comparable across different sequences, eliminating the viability of simple statistical methods such as control charts. In addition to that, noise sampled from Gaussian distribution $\mathcal{N}(0, 0.01\sigma_{\mathcal{D}})$ is added to further move the obtained data towards the real world, where $\sigma_{\mathcal{D}}$ is the feature-wise standard deviation of the data set. After simulation, $N_{\text{n}} = 3273$ highly diverse and unique nominal time series are collected.

For illustration purposes, a nominal time series is plotted in Figure 4.3.

For the generation of *anomalous* time series, six types of anomalies are considered. Some types can occur as both sub-sequence anomalies and whole-sequence anomalies, while others only in whole-sequence anomaly form, due to simulation model limitations. The distribution of sub-sequence and whole-sequence anomalies across the different anomaly types is shown in Table 4.3. Anomalies are caused by changing

Table 4.3: Number of sub-sequence anomalies N_{ss} and number of whole-sequence anomalies N_{ws} by anomaly type.

Anomaly Types	N_{ss}	N_{ws}
Regenerative Braking Off	33	32
Increased Headwind	33	31
Reduced Pump Displacement	1	23
Reduced EM Torque Request	32	33
Increased Wheel Diameter	0	33
Increased Driver Reaction Time	0	33

certain model properties *prior* to simulation, ensuring that any observed anomalous behaviour results from simulation rather than manual tampering of the data, like in the UCR data set [31], which reduces bias. All simulations are done with a fixed seed of 1.

To ensure that the different anomaly types actually lead to anomalous behaviour, control simulations with the same initial HVB states but with otherwise regular model properties are run in parallel. The data from the control simulations are solely used for identification of anomalous behaviour and are naturally excluded from the dataset.

For the first kind of anomaly, the regenerative braking is turned off, which leads to visibly different EM and axle torques, as well as HVB current and power, as these can no longer assume negative values. When regenerative braking is off, the HVB SoC

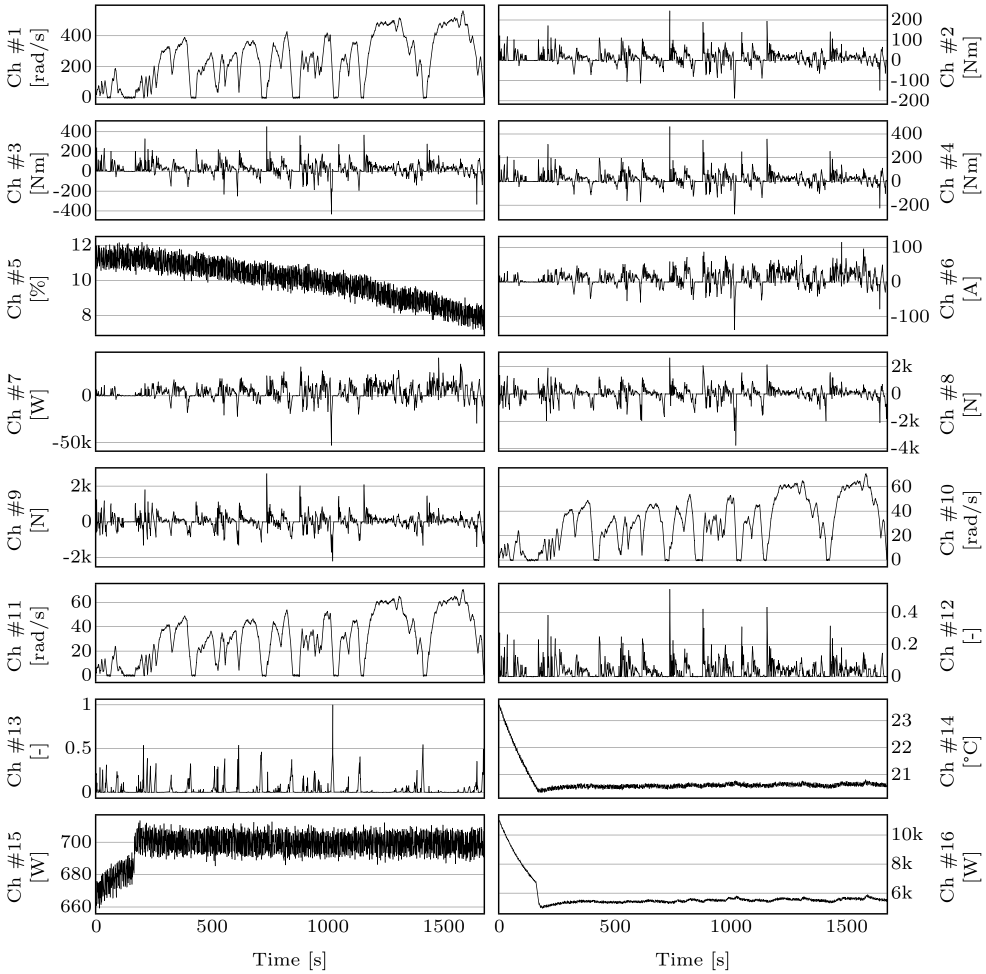

Figure 4.3: Sample plot of an anomaly free sequence with added noise and undergone trimming. The channel legend can be found in Table 4.1.

now has an exclusively negative gradient as it is no longer recharged via regenerative braking, and hence it decreases at a faster rate. The brake pedal is also used more to compensate for the missing EM torque for braking. For each of the cycles, this anomaly type is simulated in two different ways: without regenerative braking from the beginning and from a random point in time within the drive cycle. This random point in time is sampled from a uniform distribution $\mathcal{U}(0.2T, 0.8T)$, where T denotes the temporal length of the respective drive cycle, see Table 4.2. This distribution is used for all other sub-sequence anomaly types. Clearly, this introduces a positional bias, which always leads to a transition from nominal to anomalous behaviour. While this bias can easily be removed, it is kept, as this type of dynamic system would also most likely feature this bias in the real world. As discussed in Chapter 3.1, there are no metrics that are aware of detection delay and compatible with discrete-sequence problems. A set of metrics addressing the two short-comings are proposed, though they are also not perfect and can only be applied to problems with this positional bias. One of the anomalous time series for the CADC130 drive cycle and its control simulation counterpart are plotted in Figure 4.4.

In the case of the next anomaly type, a headwind of 5 m/s is simulated. This headwind acts as a resistive force on the frontal area of the vehicle and needs to be overcome to maintain the target vehicle speed. The driver model approaches this by using the accelerator pedal more than the norm, which leads to higher EM and axle torques and therefore axle forces. The higher EM torque requires a higher HVB current and power, which also causes accelerated discharging. Like previously, this anomaly type is simulated for each drive cycle, both from the beginning and from a random point in time within the cycle. One of the anomalous time series for the CLTCPassenger drive cycle and its control simulation counterpart are plotted in Figure 4.5.

Following that, the displacement of the cooling pump is reduced by 10 % to simulate another anomaly type. Evidently, this leads to a higher HVB temperature as the cooling capacity is reduced. This reduction is also visible in the pump power. Like with the previous two anomaly types, this anomaly type can start from the beginning and from a random point in time within the cycle. One of the anomalous time series for the CUEDCDieselME drive cycle and its control simulation counterpart are plotted in Figure 4.6.

For the next anomaly type, the requested EM torque value output by the powertrain control module is also reduced by 10 %. As a response to this, the driver model requests a higher acceleration pedal value and consequently a different brake pedal values as well. This anomaly type can also start from the beginning and from a random point in time within the cycle. One of the anomalous time series for the FTP75 drive cycle and its control simulation counterpart are plotted in Figure 4.7.

In the next case, the wheel diameter under weight is increased by 10 % which, for the same target vehicle speed, leads to a lower EM and axle angular speed. Furthermore, a larger wheel diameter leads to higher EM and axle torques, which are achieved using higher accelerator and brake pedal values. Here, the wheel diameter also has an effect on the HVB temperature, which, depending on its absolute magnitude, may also affect the cooling system. Due to model limitations, this anomaly can only be

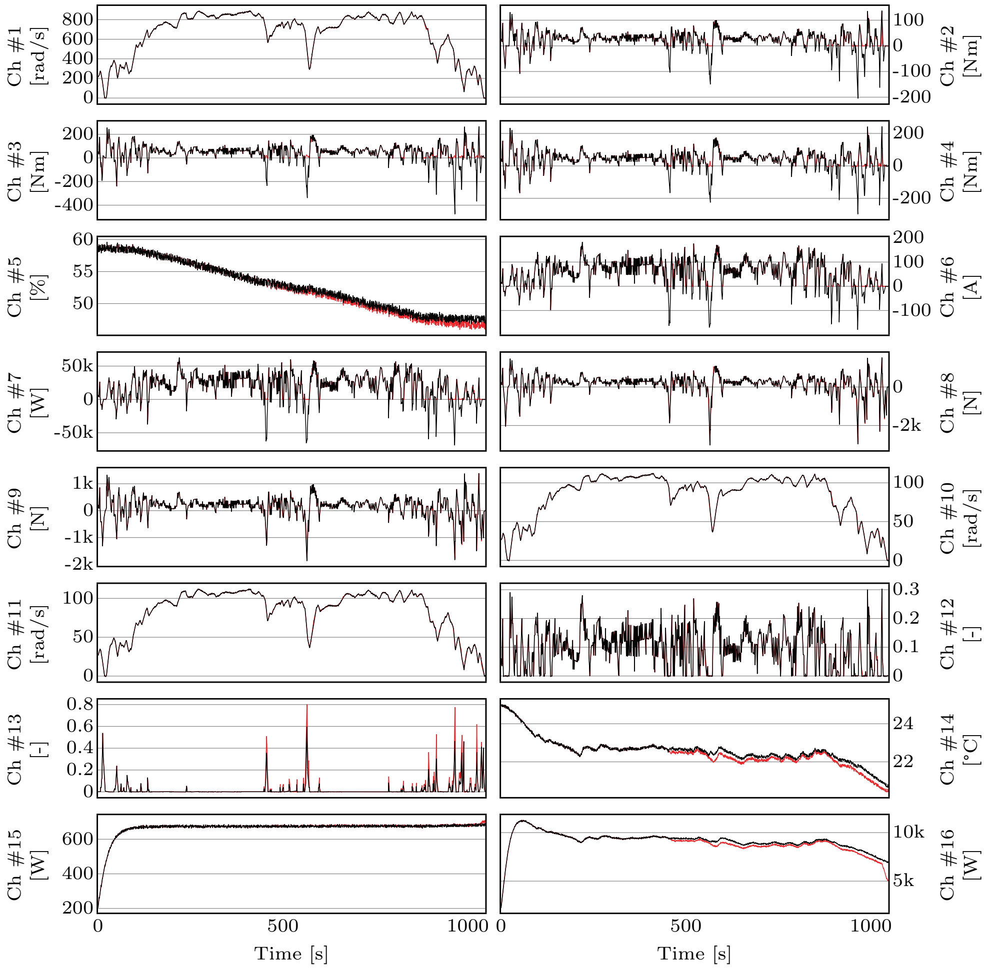

Figure 4.4: Plot of an anomalous sequence without regenerative braking (in red) and its control simulation counterpart (in black), both with added noise and undergone trimming. The anomalous sub-sequence starts after 384.6 s. The channel legend can be found in Table 4.1.

simulated for whole-sequence anomalies. One of the anomalous time series for the HUDDS drive cycle and its control simulation counterpart are plotted in Figure 4.8.

The last anomaly is recorded by increasing the driver response time by a factor of 4. This is one of the more subtle anomalies types, but manifests itself in all channels, except for the HVB temperature and cooling. Like for the wheel diameter anomaly, this anomaly can only be simulated for whole-sequence anomalies. One of the anomalous time series for the LA92 drive cycle and its control simulation counterpart are plotted

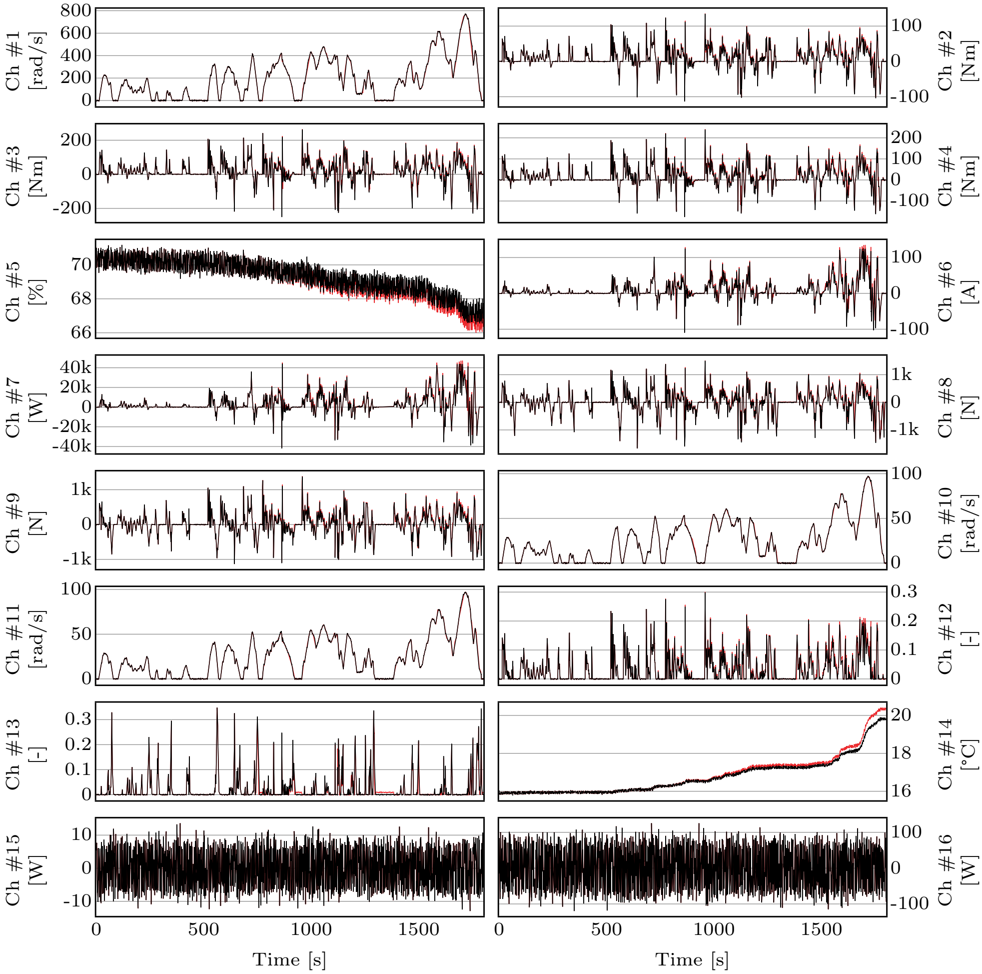

Figure 4.5: Plot of an anomalous sequence with an added headwind (in red) and its control simulation counterpart (in black), both with added noise and undergone trimming. The anomalous sub-sequence starts after 738.0 s. The channel legend can be found in Table 4.1.

in Figure 4.9.

Recall the findings by Wenig et al. [34] in Chapter 3.1. As outlined above, all anomaly types manifest themselves in more than one channel, making the generated dataset truly multivariate with multivariate anomalies. This further underlines the real-world-motivated nature of the dataset, since properties of complex real-world systems also correlated with one another.

Given the uniform distribution from which the HVB temperature is sampled from, half of the simulated anomaly types start with a HVB temperature below 20 °C. In

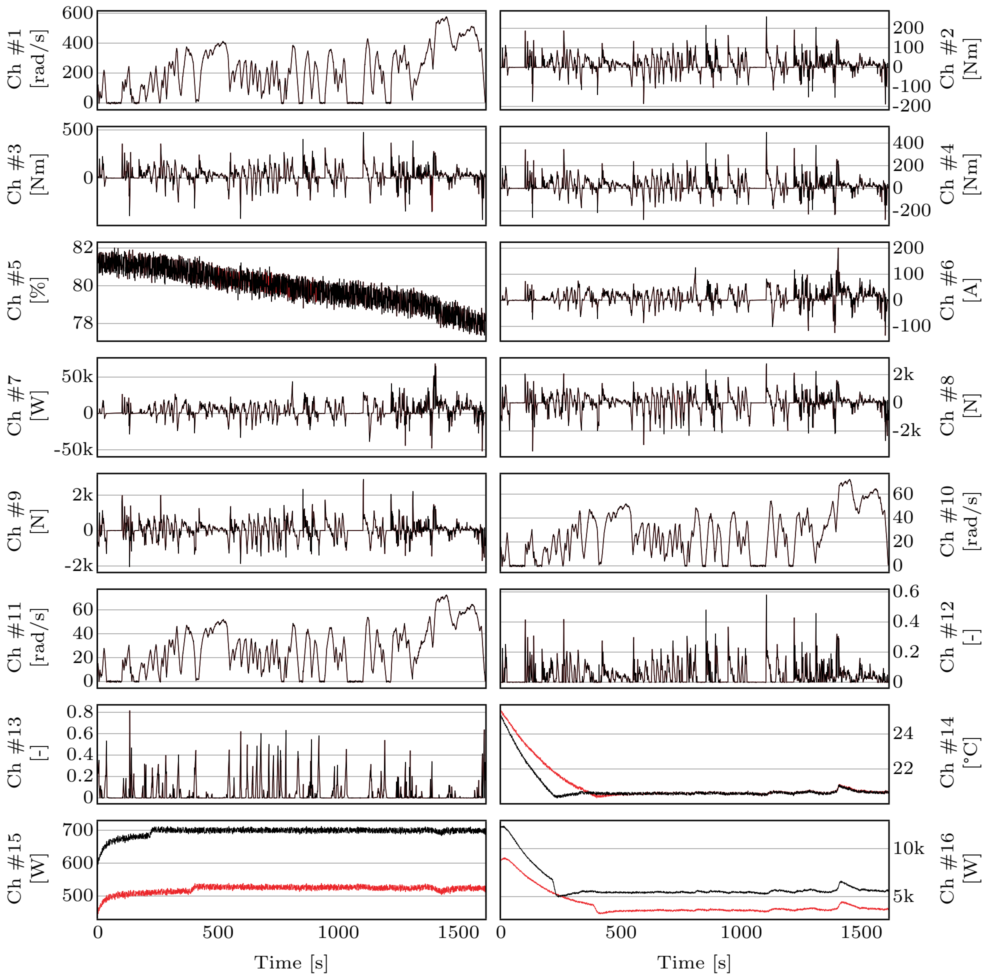

Figure 4.6: Plot of an anomalous sequence with a reduced cooling pump displacement (in red) and its control simulation counterpart (in black), both with added noise and undergone trimming. It is a whole-sequence anomaly, and hence the anomalous behaviour starts from the first time step. The channel legend can be found in Table 4.1.

these cases, the HVB will naturally heat up as it is being used, and hence the cooling system does not play a role in short drive cycles. Therefore, in the case of the reduced cooling pump displacement, often no anomalous behaviour can be observed because the simulated anomaly is identical with the corresponding control simulation. For these cases, the simulated anomaly is discarded.

Finally, this results in $N_{\mathrm{a}} = 284$ successful simulations yielding anomalous time series, where $N_{\mathrm{a}} = N_{\mathrm{ss}} + N_{\mathrm{ws}}$, N_{ss} denotes the number of sub-sequence anomalies and

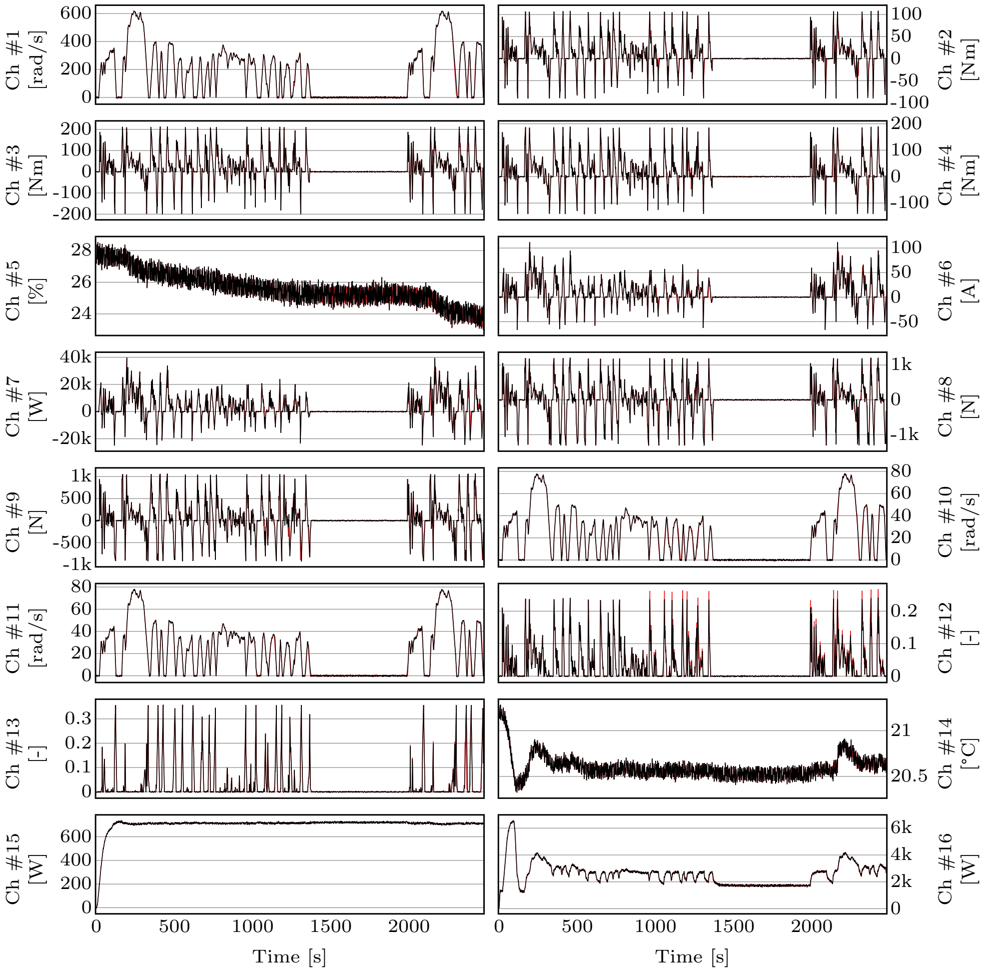

Figure 4.7: Plot of an anomalous sequence with a reduced requested EM torque (in red) and its control simulation counterpart (in black), both with added noise and undergone trimming. The anomalous sub-sequence starts after 940.2 s. The channel legend can be found in Table 4.1.

N_{ws} the number of whole-sequence anomalies. Hence, the entire data set $\mathcal{D}$ consists of $N_{\mathrm{n}} + N_{\mathrm{ss}} + N_{\mathrm{ws}} = 3557$ unique (nominal and anomalous) multivariate time series, with an anomalous sequence ratio of $N_{\mathrm{a}}/N \approx 8\,\%$. This amount of anomalous data in relation to anomaly-free data is used as it approximately matches the anomaly ratio in public data sets [2–4, 32] and fulfils the assumption that anomalous behaviour is rare by nature [33]. $\mathcal{D}$ is then shuffled and divided into three separate folds for cross-validation, which corresponds to a 67/33 split for training and test, respectively.

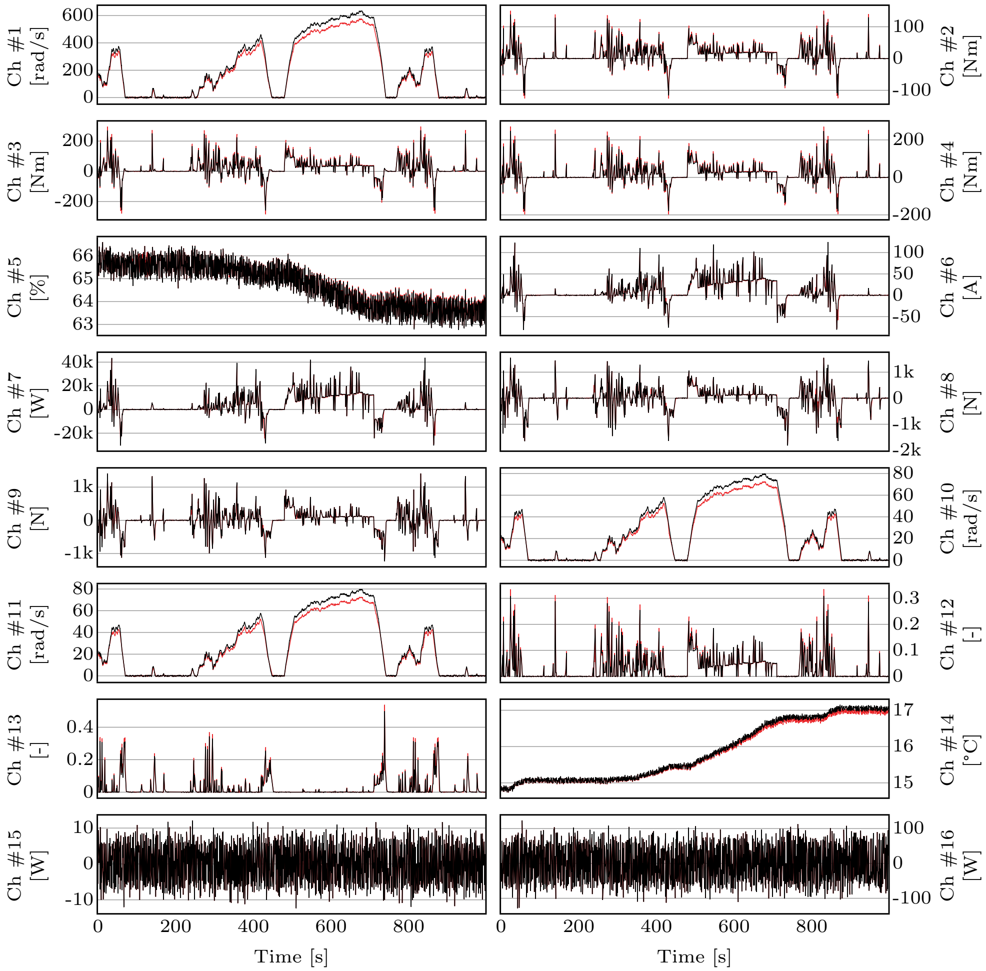

Figure 4.8: Plot of an anomalous sequence with an increased loaded wheel diameter (in red) and its control simulation counterpart (in black), both with added noise and undergone trimming. It is a whole-sequence anomaly, and hence the anomalous behaviour starts from the first time step. The channel legend can be found in Table 4.1.

Formally, the training subset $\mathcal{D}^{\text{train}} = \{\mathcal{S}_1, ..., \mathcal{S}_m, ..., \mathcal{S}_M\}$ then consists of $M = 2371$ multivariate time series on average, where $\mathcal{S}_m \in \mathbb{R}^{T_m \times d_{\mathcal{D}}}$, where T_m is the number of time steps in sequence $\mathcal{S}_m$. Likewise, the test subset $\mathcal{D}^{\text{test}} = \{\mathcal{S}_1, ..., \mathcal{S}_n, ..., \mathcal{S}_N\}$ consists of $N = 1186$ multivariate time series on average, where $\mathcal{S}_n \in \mathbb{R}^{T_n \times d_{\mathcal{D}}}$, where T_n is the number of time steps in sequence $\mathcal{S}_n$. For benchmarking purposes, the use of the prescribed training and test split are strongly recommended.

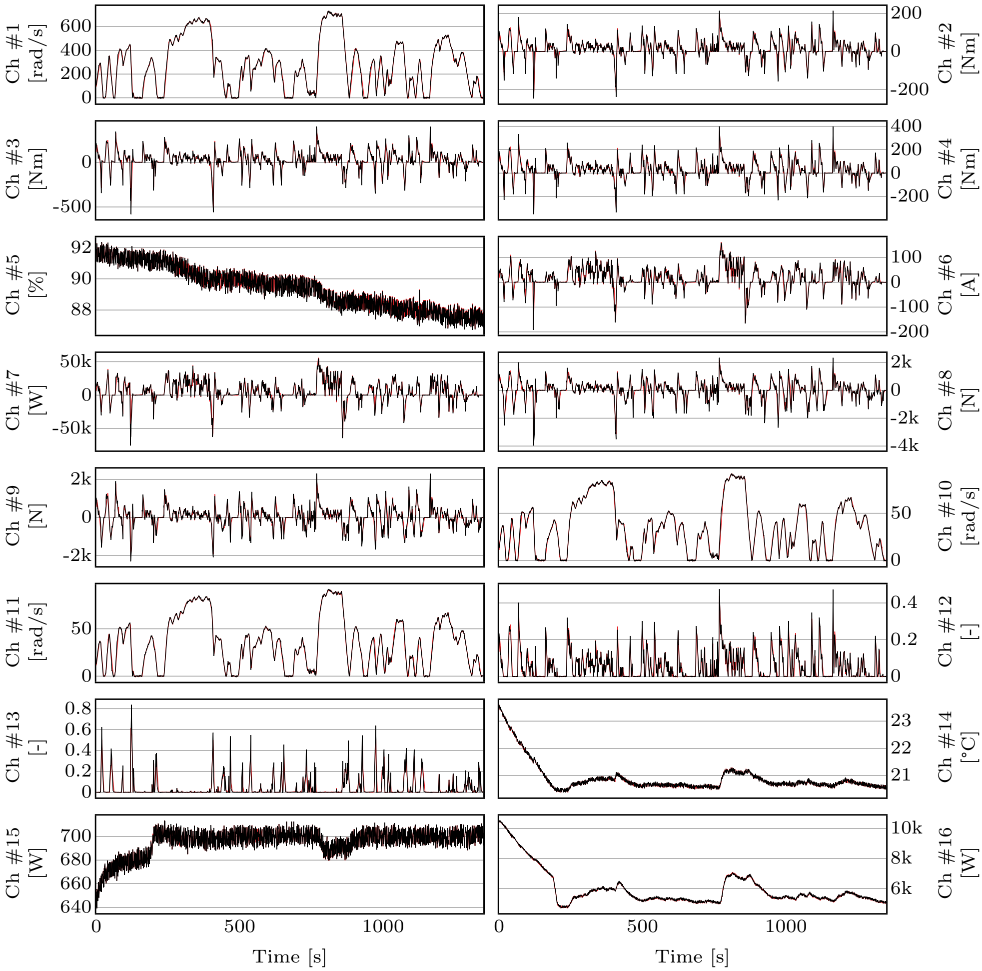

Figure 4.9: Plot of an anomalous sequence with increased an driver response time (in red) and its control simulation counterpart (in black), both with added noise and undergone trimming. It is a whole-sequence anomaly, and hence the anomalous behaviour starts from the first time step. The channel legend can be found in Table 4.1.

4.1.3 Usage of the Data Set

Clearly, both $\mathcal{D}^{\text{train}}$ and $\mathcal{D}^{\text{test}}$, as specified above, contain nominal and anomalous time series, though in an unsupervised setting the labels for $\mathcal{D}^{\text{train}}$ should be disregarded. This is because, this data set is aimed at *unsupervised* time series anomaly detection, which requires approaches especially robust to contaminated training data.

Apart from the unsupervised setting, the underlying properties of the data set can be useful in other research areas. The same $\mathcal{D}^{\text{train}}$ and $\mathcal{D}^{\text{test}}$ subsets can also be used for *imbalanced time series classification*. For *semi-supervised* anomaly detection, a clean $\mathcal{D}^{\text{train}}$ with on average $M = 2182$ nominal time series and the same labelled $\mathcal{D}^{\text{test}}$ is provided in the data set, where clean refers to the absence of anomalous sequences in the subset. Furthermore, for time series *forecasting* or *generation*, clean versions of both $\mathcal{D}^{\text{train}}$ and $\mathcal{D}^{\text{test}}$ are provided, where $M = 2182$ and $N = 1091$ on average, respectively. Despite being targeted at *online* time series anomaly detection, the PATH data set can just as well be used in *offline* time series anomaly detection. For an explanation of online and offline anomaly detection, please refer to Chapter 2.2.

4.1.4 Preliminary Analysis

To provide a more complete picture to the reader, some preliminary analysis on the generated dataset is provided. As mentioned above, the simulation model has a sampling frequency 10 Hz, meaning that time steps of the resulting time series are 0.1 s apart. According to the Nyquist–Shannon theorem [98], a signal sampled at an arbitrary sampling frequency can only represent dynamic process frequencies of at most half the sampling frequency, which, in this dataset, is 5 Hz.

Fourier transformations allow for the decomposition of time series data into its components in the frequency domain. For a brief explanation of the Fourier transformation, please refer to Chapter 2.1. To find which frequencies contain the most information, a Fourier transformation is performed for each feature and for each driving cycle. Summing the amplitudes of all features for each driving cycle reveals that, most information is represented in the lower frequency range, as exemplified in Figure 4.10. Another observation is how complex the drive cycles are. Consider the simple combination of sine waves at 1 Hz and 2 Hz, a cosine wave at 5 Hz, and some minor noise at 20 Hz, as shown in the top half of Figure 4.11. Its Fourier transform, shown in the bottom half of Figure 4.11, clearly uncovers the underlying frequencies within the data; see the three peaks at the individual component frequencies as well as the minor peak caused by noise at 20 Hz. In contrast, Figure 4.10 underlines how much more complex the underlying time series is, as illustrated by all the different frequencies that make up the signal.

Since most information is present in the lower frequencies of the data, computational effort can be reduced through downsampling. To avoid aliasing and maintain signal integrity, downsampling is performed by low-pass filtering the time series data. The most important parameter of a low-pass filter is the cut-off frequency, above which

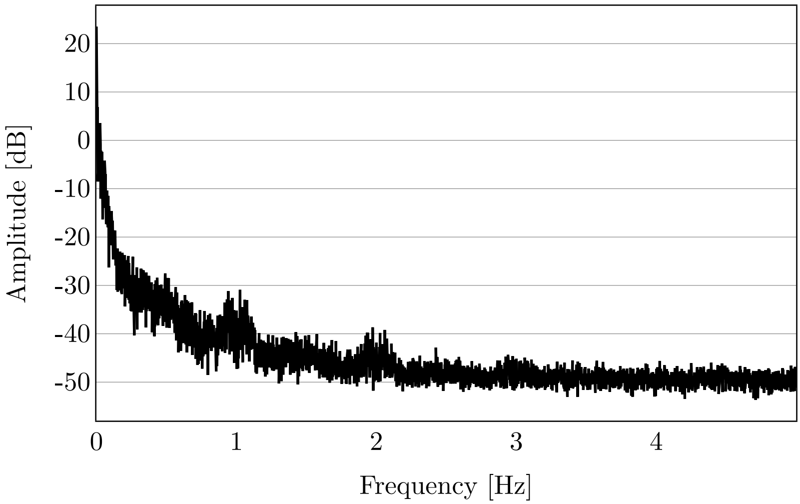

Figure 4.10: Fourier plot of the WLTP1 drive cycle from the PATH dataset. The amplitude seen is the sum of the amplitudes for each feature.

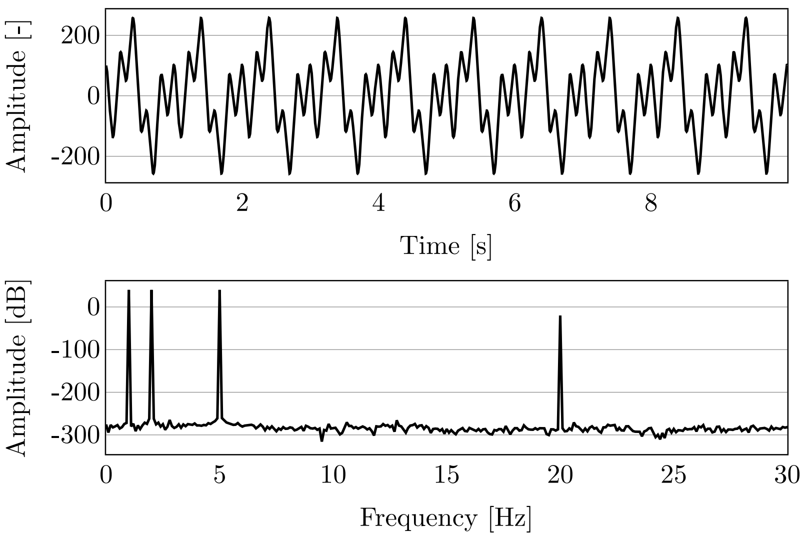

Figure 4.11: Fourier plot of a simple combination of a sine waves.

all frequencies attenuated and, in case of an ideal low-pass filter, removed. The cut-off frequency should be set to at most half of the sampling rate to be downsampled to, as to comply with the Nyquist–Shannon theorem [98]. To quantify the impact of downsampling on the data, various cut-off frequencies are iterated through, and the mean-squared error (MSE) between the original and the filtered time series is computed. The goal is to find a cut-off frequency which strikes the balance between computational effort, but also fidelity compared to the original data. Figure 4.12 shows the average MSE across all sequences in all folds. As expected, the MSE is

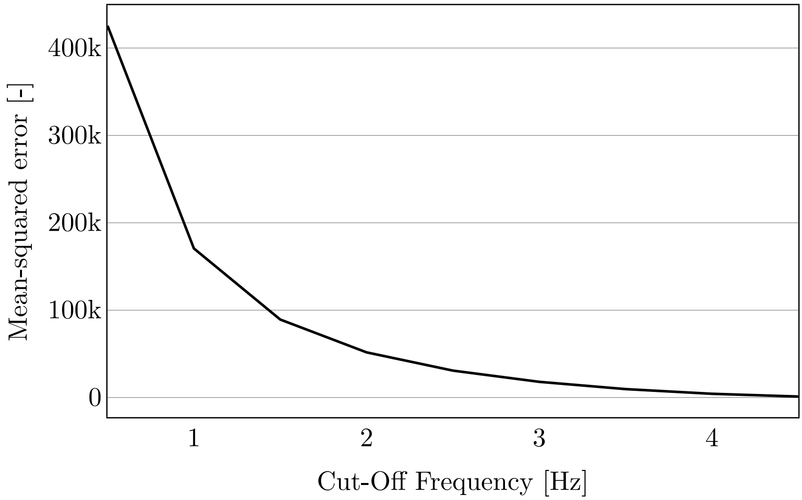

Figure 4.12: Mean-squared error as a function of cut-frequency used in the low-pass filter.

highest at very low cut-off frequencies and decreases rapidly as the cut-off frequency increases. Parameterising the low-pass filter with a cut-off frequency below 1 Hz is therefore undesired, and hence downsampling the dataset to 2 Hz is recommended.

Once the time series data is downsampled, the variable-length sequences can be windowed to create a set of fixed-length sub-sequences, also known as *windows*. The reasoning behind using windows rather than entire sequences is that most of the dynamic processes in the time series data are fairly fast and hence only have a short-term effect that lasts for a small period of time. Modelling entire variable-length sequences would be possible, but much of the model learning capacity would be wasted on internal state mechanisms carrying information for longer periods of time than necessary. This therefore allows for more efficient model training on windows that are only as long as they need to be to represent all present dynamics in the time series data. To find the effect length of the slowest dynamics, an autocorrelation analysis is undertaken for each of the drive cycles and each of the features within them. The pseudocode representing the windowing procedure is shown in Algorithm 1 and is also denoted as the `window` function.

This contrasts with approaches taken in the literature, which treat window size as a hyperparameter that is tuned. An autocorrelation analysis yields the correlation

Algorithm 1 Windowing Process (`window`)

Require: Sequence $\mathcal{S}_n \in \mathbb{R}^{T_n \times d_\mathcal{D}}$, Window Size $w \in \mathbb{N}$, Window Shift $k \in \mathbb{N}$

$\lambda \leftarrow (T_n - w)/k + 1$ $\triangleright$ Find the total number of windows in $\mathcal{S}_n$

for $i = 1 \rightarrow \lambda$ **do** $\triangleright$ Iterate through total number of windows

  $\mathbf{X}[\lambda] \leftarrow \mathcal{S}_n[i * k : i * k + w]$ $\triangleright$ Assign current window

end for

return Window Matrix $\mathbf{X} \in \mathbb{R}^{\lambda \times w \times d_\mathcal{D}}$

between the time series and a range of lagged versions of the time series for a specified number of lags and does not require labels. For a more detailed explanation of the autocorrelation analysis, please refer to Chapter 2.1. The autocorrelation plot for the slowest signal of all sequences in the dataset is shown in Figure 4.13.

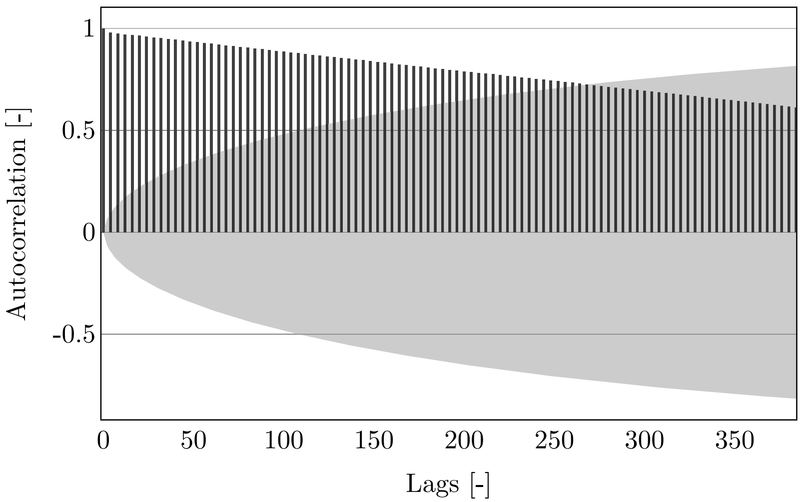

Figure 4.13: Autocorrelation of the SoC signal for the FTP75 as a function of lags. The band shown represents the confidence interval, below which the autocorrelation is no longer statistically significant.

As is evident, the autocorrelation decreases with the number of lags, which is consistent with theory as dynamic effects dampen with time. Therefore, the slowest dynamics present that is still significant is at the lag where the autocorrelation intersects the confidence interval. It is no surprise that the SoC signal features the slowest dynamics; recall Figure 4.3, where the slow dynamics is especially visible. Therefore, the window size parameter should be set as the temporally furthest-away but statistically significant lag of the signal with the slowest dynamics in all sequences, which for the PATH dataset is 268 time steps. Given that some approaches in the benchmarking section strictly require windows to be a multiple of 2, the window size is therefore rounded to $w = 256$, representing 128 s at a 2 Hz sampling rate. Of course, the decision on a suitable sampling rate could also be done manually by a domain expert, who is interested in anomalous behaviour at specific time scales.

4.2 Real-world Application: Endurance Testing

To complement the results obtained with the generated PATH dataset, a dataset from the real world is also experimented with. The dataset stems from the endurance testing of an automotive battery-electric powertrain, a procedure which aims to simulate real-world use by the customer over several years by running tests with little downtime for a prolonged period of time, thereby accelerating powertrain ageing in a shorter period of time. Given the compressed nature of this type of testing, large amounts of data can be collected. Current monitoring systems are rule-based, which can only detect anomalies outside the operating range, do not take signal dynamics into account and represent a high level of manual parameterisation. Therefore, such a real-world scenario is especially suited for the application of a time series anomaly detection algorithm.

4.2.1 Automotive Powertrain Testing Setup

During endurance testing, a set of exclusively proprietary drive cycles is run, which differs from the public drive cycles used in the PATH dataset in Chapter 4.1. The reason proprietary drive cycles are used for endurance runs is that they allow for more extensive loading of the powertrain than public drive cycles, which are mainly used for fuel or energy consumption certification. The portfolio consists of eight dynamic drive cycles representing short, long, fast, slow and dynamic trips ranging from 5 to 30 minutes. Like the PATH dataset, this dataset also contains variable-state behaviour caused by the presence of a HVB in the system.

Aside from the HVB, the testee consists of other components too. The inverter converts the direct current from the HVB into alternating current, which leads to the EM. The EM is then connected via drive shafts to resistance machines which simulate the road load. The EM, inverter, and powertrain controller are cooled via the same loop, and hence their temperatures have some correlation, while the HVB is cooled using a separate loop. A simplified schematic of the components and their relationship is shown in Figure 4.14.

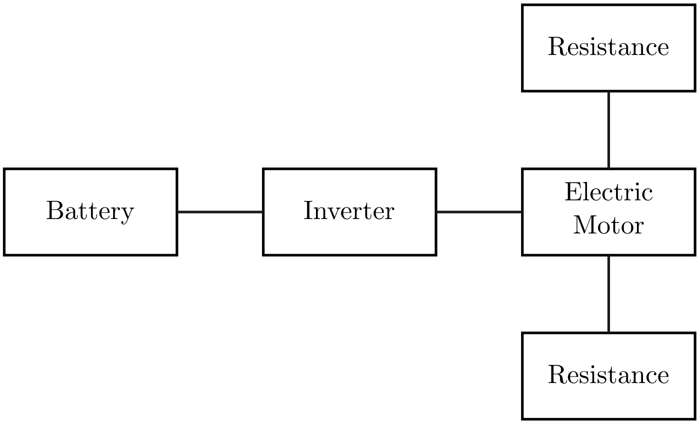

Figure 4.14: High-level schematic of powertrain. Direct current flows from the HVB to the inverter, which in turn converts it to alternating current for the EM. The EM drives two axles, each connected to a resistance.

4.2.2 Data Set

The nominal subset of this real-world data set consists of 1875 of variable-length sequences, which are divided into three folds to enable cross-validation, where 2/3 are used for training and 1/3 for testing. Therefore, the unlabelled training subset $\mathcal{D}^{\text{train}}$ in each fold consists of $M = 1250$ unlabelled time series, where $\mathcal{D}^{\text{train}} = \{\mathcal{S}_1, ..., \mathcal{S}_m, ..., \mathcal{S}_M\}$. For this work, a list of representative $d_{\mathcal{D}} = 22$ channels is hand-picked in consultation with test bench engineers. The chosen features along with their indices are as shown in Table 4.4, while a plot of one anomaly-free measurement is shown in Figure 4.15.

The testing subset $\mathcal{D}^{\text{test}}$ consists of N labelled time series such that $\mathcal{D}^{\text{test}} = \{\mathcal{S}_1, ..., \mathcal{S}_n, ..., \mathcal{S}_N\}$. Recall that the nominal portion of the testing subset $\mathcal{D}^{\text{test}}$ accounts for $N_{\text{n}} = 625$ measurements. Due to the absence of labelled anomalies in the dataset, realistic anomalous events are intentionally provoked and recorded following the advice of test bench engineers. To this end, four anomaly types are recorded. Every anomaly type is recorded for every drive cycle at least once, leading to $N_{\text{a}} = 47$ anomalous measurements that are all used as the labelled anomalous portion of the testing subset $\mathcal{D}^{\text{test}}$. Hence, the testing subset $\mathcal{D}^{\text{test}}$ in each fold is made up of $N = N_{\text{n}} + N_{\text{a}} = 672$ measurements on average, representing an anomaly sequence ratio of around $N_{\text{a}}/N \approx 7\,\%$, putting it in line with the PATH dataset and other public datasets [2–4, 32].

In the first type, the virtual wheel diameter is changed, such that the resulting

Table 4.4: Channels (features) chosen for modelling testee behaviour. The indices correspond to Figure 4.15.

Index	Signal Name	Index	Signal Name
1	Vehicle Speed	12	Inverter Inlet Temperature
2	HVB Voltage	13	Inverter Outlet Temperature
3	HVB Current	14	Powertrain Controller Temperature
4	HVB Temperature	15	Left Axle Torque
5	HVB SoC	16	Right Axle Torque
6	EM Voltage	17	Left Axle Angular Speed
7	EM Current	18	Right Axle Angular Speed
8	EM Torque	19	Cooling Loop 1 Inlet Temperature
9	EM Angular Speed	20	Cooling Loop 1 Outlet Temperature
10	EM Stator Temperature	21	Cooling Loop 2 Inlet Temperature
11	Inverter Temperature	22	Cooling Loop 2 Outlet Temperature

vehicle speed deviates from the norm. The wheel diameter is an adjustable parameter, rather than a physical quantity, as resistances are connected to the shafts rather than actual wheels. This accounts for 16 whole-sequence anomalies and the only channel that demonstrates anomalous behaviour is the vehicle speed, since:

$$v_{\text{vehicle}} = r \cdot \omega \tag{4.1}$$

Here, r is the wheel radius and ω the angular speed of the drive shaft. Logically, the anomalous behaviour is most visible at higher speeds.

In the case of the next anomaly type, the recuperation level is turned from maximum to zero, and hence the minimum EM, and therefore left and right axle torques are always non-negative and the HVB SoC experiences a higher drop. This anomaly type accounts for another eight whole-sequence anomalies.

For the following anomaly, the HVB is swapped for a battery simulator, where the voltage behaviour deviates from a real HVB. Considering that operation works by requesting a given amount of power, a different voltage behaviour will also result in a change in current, since power P is the product of voltage across the HVB V and the current I:

$$P = V \cdot I \tag{4.2}$$

Therefore, this type of anomaly will be evident in the HVB and EM voltage channels, as well as the respective current channels, representing 15 whole-sequence anomalies.

The inverter and EM share a cooling loop, whose cooling capacity is reduced at the

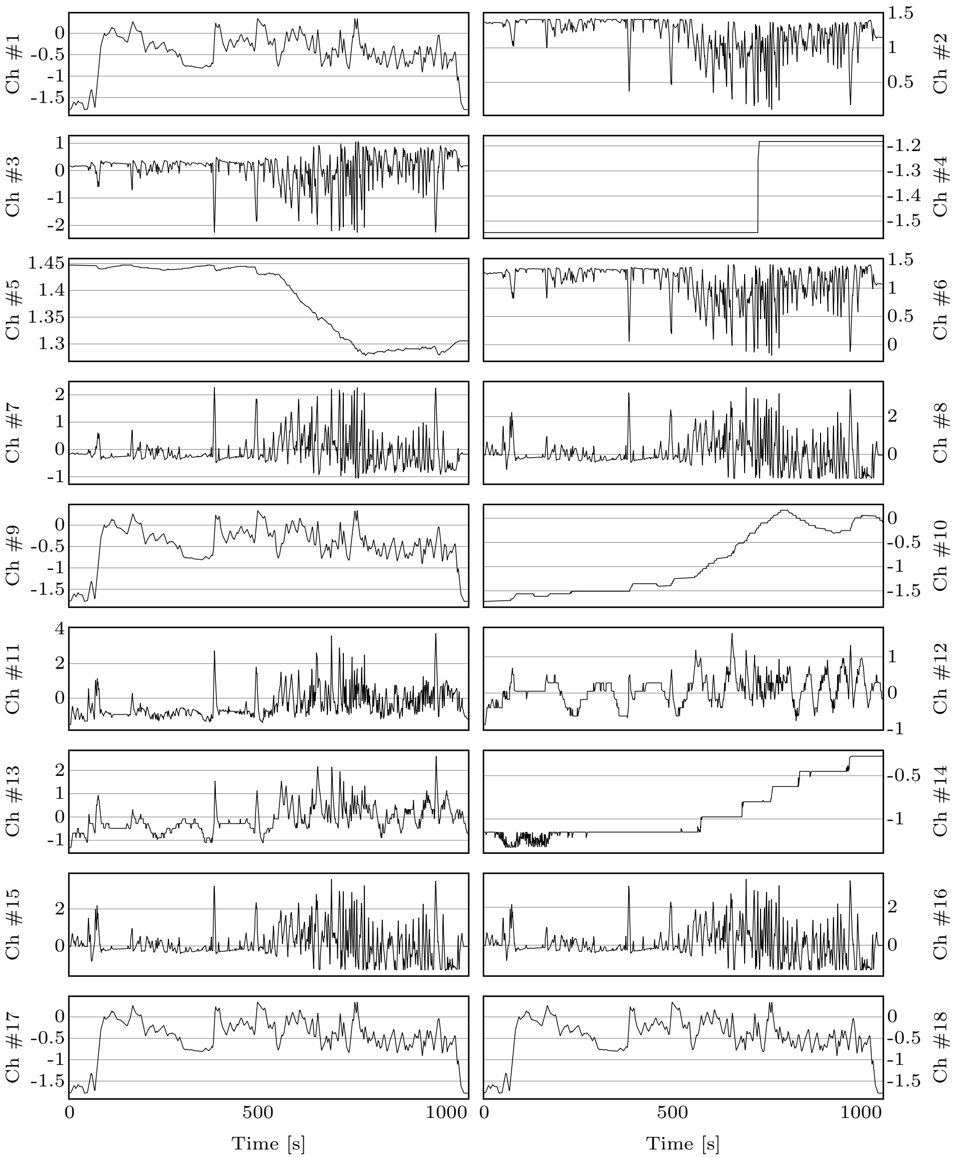

Figure 4.15: Sample plot of an anomaly free sequence. The channel legend can be found in Table 4.4. Note that, the cooling loop inlet and outlet temperatures are omitted due to space constraints. Unlike Figure 4.3, the y-axis ticks are removed due to comply with confidentiality requirements.

beginning or middle of the measurement for the last type of anomaly. This leads to, for instance, higher EM rotor, EM stator, and inverter temperatures than normal. For three of the sequences, the capacity is reduced at the midpoint of the measurement, whereas another five sequences are recorded with reduced cooling capacity from the very beginning.

In an operative environment, it is desirable to find out whether the previously recorded sequence had any problems to trigger an inspection before the next measurement is recorded. Furthermore, a model that performs as well as possible with as little data as possible translates to faster deployment. Good performance is indicated by a model that can detect as many anomalies as possible and rarely labels anomaly-free measurements wrongly.

4.2.3 Preliminary Analysis

Analogous to Chapter 4.1, a preliminary analysis is also undertaken on the real-world dataset.

Performing a Fourier transformation for each feature and summing their amplitudes reveals that the same observations are made as with the PATH dataset. For the real-world dataset, most information is also represented in the lower frequency range, as shown in Figure 4.16.

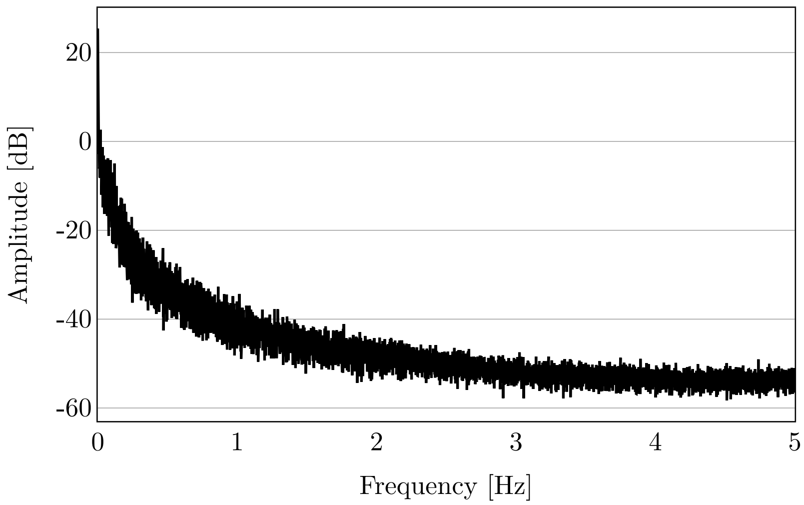

Figure 4.16: Fourier plot of an example drive cycle from the real-world dataset. The amplitude seen is the sum of the amplitudes for each feature.

The fact that the same observations are made in both a real-world dataset and a generated one further supports the claim of the PATH dataset being real-world motivated despite being synthetic.

Like in Chapter 4.1, the impact of downsampling on the data is quantified by iterating through various cut-off frequencies and computing the MSE between the original and the filtered time series. Figure 4.17 shows the average MSE across all sequences in all folds.

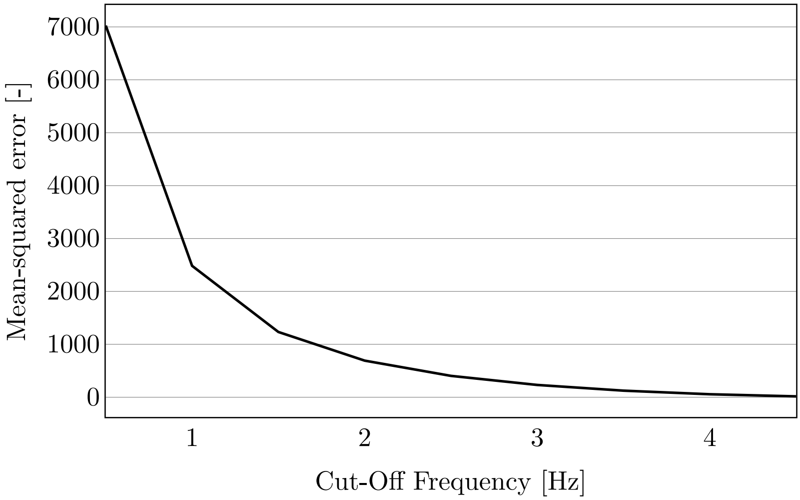

Figure 4.17: Mean-squared error as a function of cut-frequency used in the low-pass filter.

Similarly, the MSE is also highest at very low cut-off frequencies, and hence downsampling the dataset to 2 Hz is recommended as well.

Lastly, a suitable window size is also computed using an autocorrelation analysis. Again, the autocorrelation plot of the slowest channel of all sequences in the dataset is shown in Figure 4.18.

Unlike the PATH dataset, the cooling loop 1 inlet temperature has even slower dynamics than the SoC. Its autocorrelation is statistically significant up to 304 lags, and hence the window size is, again, rounded to $w = 256$ time steps.

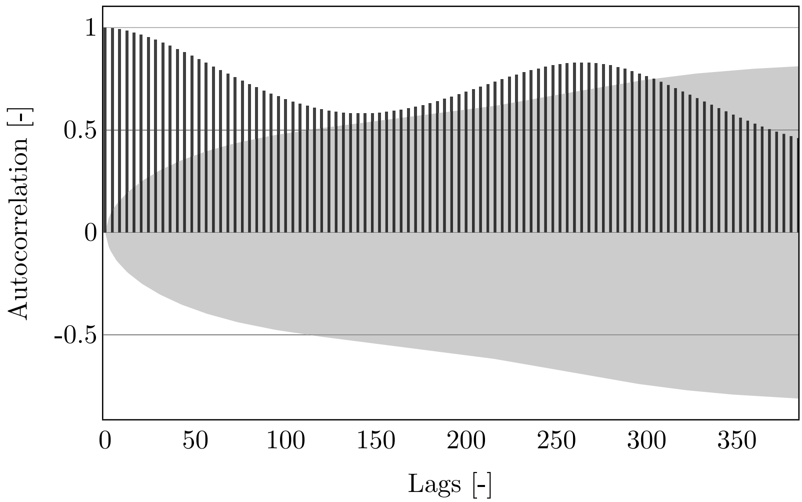

Figure 4.18: Autocorrelation of the SoC signal for drive cycle B as a function of lags. The band shown represents the confidence interval, below which the autocorrelation is no longer statistically significant.