

# Prozessdiagnose Waschmaschine

FH Aachen  
Modul 81307  
Wintersemester 2021/22  
Projektaufgabe

Matthias Simons  
Fabian Meyer

3104576  
3125420

[matthias.simons@alumni.fh-aachen.de](mailto:matthias.simons@alumni.fh-aachen.de)  
[fabian.meyer@alumni.fh-aachen.de](mailto:fabian.meyer@alumni.fh-aachen.de)

## Inhaltsverzeichnis

PROZESSBESCHREIBUNG .....	3
DATENERFASSUNG.....	3
DATENTYPEN, -MENGE UND BANDBREITE.....	4
KOMPRESSION .....	5
DATENBANKANBINDUNG UND -ZUGÄNGLICHKEIT.....	6
KLASSIFIKATION.....	7
BENUTZEROBERFLÄCHE.....	8
LESSONS LEARNED .....	9
BEDIENUNGSANLEITUNG.....	9

## Prozessbeschreibung

Die Waschmaschine durchläuft die folgenden Betriebszustände, welche mit LEDs dargestellt werden (siehe Abbildung 1):

1. Bereit
2. Waschen
3. Spülen
4. Schleudern
5. Fertig

In den Betriebszuständen werden mehrere Prozesse durchlaufen. Dabei werden die Zustände der Trommel und der Pumpe betrachtet. Die Trommel kann die Zustände Stillstand, Rotieren oder Schleudern annehmen. Beim Zustand Stillstand rotiert die Trommel nicht. Beim Rotieren rotiert die Trommel mit einer Drehzahl von etwa 30 Umdrehungen pro Minute 5-10-mal linksherum, wechselt in den Zustand Stillstand für einige Sekunden und rotiert dann wieder 5-10-mal rechtsherum. Schleudern zeichnet sich dadurch aus, dass die Trommel für einen längeren Zeitraum mit der eingestellten Drehzahl rotiert. Die Pumpe kann die Zustände an oder aus annehmen. In den Betriebszuständen waschen, spülen und schleudern kann die Trommel die Zustände stillstand, rotieren oder schleudern annehmen und die Pumpe ist an oder aus. In den Zuständen bereit und fertig ist die Trommel im Stillstand und die Pumpe ist aus. Aus diesem Grund werden beide zusammengefasst.



Abbildung 1 Waschmaschine (links) und Betriebszustände/Prozessschritte (rechts)

## Datenerfassung

Die Messeinrichtung besteht aus einem Beschleunigungs- und Lagesensor (GY-521 MPU-6050), einem Temperatur- und Luftfeuchtigkeitssensor (DHT22 AM2302) und einem Microcontroller (ESP 8266). Die einzelnen Komponenten sind auf einem Bredboard angeordnet. Der Temperatur- und Luftfeuchtigkeitssensor ist für die Datenerfassung nicht relevant und wird nicht ausgelesen.



Abbildung 2 Versuchsaufbau

Zur Erfassung der Vibrationsdaten wurde die Messeinrichtung an die Waschmaschine geklebt und über ein USB-Kabel mit Strom versorgt. Wichtig zu beachten ist, dass der Microcontroller am Standort der Waschmaschine eine Netzwerkverbindung aufbauen kann.

### Datentypen, -menge und Bandbreite

Die Berechnung der Datenübertragungsrate hängt unmittelbar von der Datenmenge und der Zeit ab. Die Messwerte werden in Textform als String in folgender Form versandt:

"(timestamp);(AX,AY,AZ)?"

```
timestamp=str(rtc.datetime())
AX,AY,AZ=accel.xyz
```

Zur Bestimmung der Datenmenge wurde untersucht, wie groß ein Datensatz ist. Dafür wurde die Größe eines typischen Datensatzes mit der Funktion „sys.getsizeof()“ untersucht. Die Größe beträgt 113 Byte.

Die Bestimmung der Zeit ergibt sich aus der Abtastrate. Diese wird aus der maximal auftretenden Frequenz bestimmt, in diesem Fall die maximal einstellbare Schleuder-Drehzahl.

$$f_{max} = f_{Schleudern} = 1200 \frac{1}{Minute} = 20 \text{ Hz}$$

Unter Berücksichtigung des Nyquist-Kriteriums wird die doppelte Abtastrate gewählt.

$$f_s = 2 \cdot f_{schleudern} = 40 \text{ Hz}$$

Aus dem Kehrwert der Abtastrate ergibt sich die Periode bzw. die Zeit zur Berechnung der Datenübertragungsrate.

$$T_s = \frac{1}{f_s} = 0,025 \text{ s} = 25 \text{ ms}$$

Mit der Datenmenge und der Zeit kann nun die Datenübertragungsrate berechnet werden.

$$D = 113 \text{ Byte} = 0,113 \text{ kB}$$

$$C = \frac{D}{T_s} = \frac{0,113 \text{ kB}}{0,025 \text{ s}} = 4,52 \frac{\text{kB}}{\text{s}}$$

Die theoretische Datenübertragungsrate beträgt  $4,52 \frac{\text{kB}}{\text{s}}$ . Die vorhandene Bandbreite von  $50,17 \frac{\text{MB}}{\text{s}}$  ist ausreichend hoch (Abbildung 3).

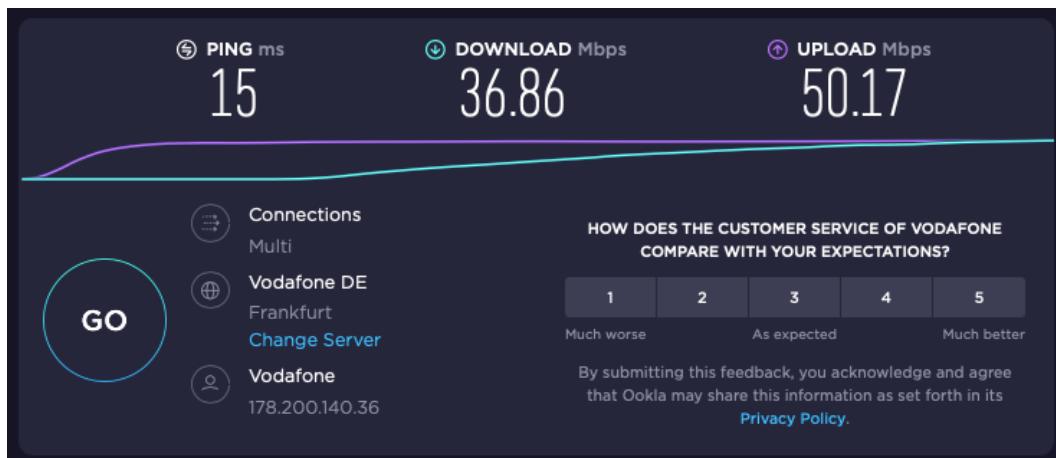


Abbildung 3 Bandbreite am 08.02.2022 um 16:27 geprüft bei speedtest.net

Zur Datenübertragung wird das Socket (TCP/IP) Kommunikationsprotokoll mit einem Client (Messeinrichtung) und einem Server (Gateway) genutzt.

### Kompression

Um die angestrebte Abtastrate von  $25 \text{ ms}$  zu erreichen, muss das Programm auf dem Microcontroller so schnell wie möglich durchlaufen werden können. Dies wurde dadurch erreicht, dass nicht auf globale Variablen zugegriffen wurde und keine nicht notwendigen Transformationen durchgeführt wurden. Außerdem wurden die benötigten Funktionen direkt aufgerufen und in das String Format transformiert, ohne diese als Variabel zwischen zu speichern. Dadurch wird eine Abtastrate von durchschnittlich  $28,33 \text{ ms}$  erreicht.

Der limitierende Faktor im Programm ist die Abfrage der Uhrzeit (`rtc.datetime()`). Da die Uhrzeit für die weitere Zuordnung der Daten relevant ist, wird auf eine weitere Komprimierung verzichtet. Allerdings sind somit nur Drehzahlen bis  $1.058,95 \frac{1}{\text{min}}$  messbar (Tabelle 1). Aus diesem Grund wurden die Messreihen mit einer Schleuderdrehzahl von  $800 \frac{1}{\text{min}}$  (Sicherheitsfaktor: 1,32) durchgeführt.

Tabelle 1 Maximal auftretende Drehzahlen und Berechnung der maximal abtastbaren Drehzahl

Betriebszustand_Trommel	Rotieren	Schleudern	Tatsächlich
fmax 1/min	32,43	1.200,00	1.058,95
fmax (Hz)	0,54	20,00	17,65
fs (Hz)	1,08	40,00	35,30
Ts (ms)	925,12	25,00	28,33

### Datenbankanbindung und -zugänglichkeit

Als zentrale Schnittstelle zwischen Server, GUI-Backend und GUI-Frontend dient eine NoSQL-Datenbank. Umgesetzt wurde die Datenbank in MongoDB Atlas. In Abbildung 4 ist die Architektur zwischen den einzelnen Komponenten dargestellt.

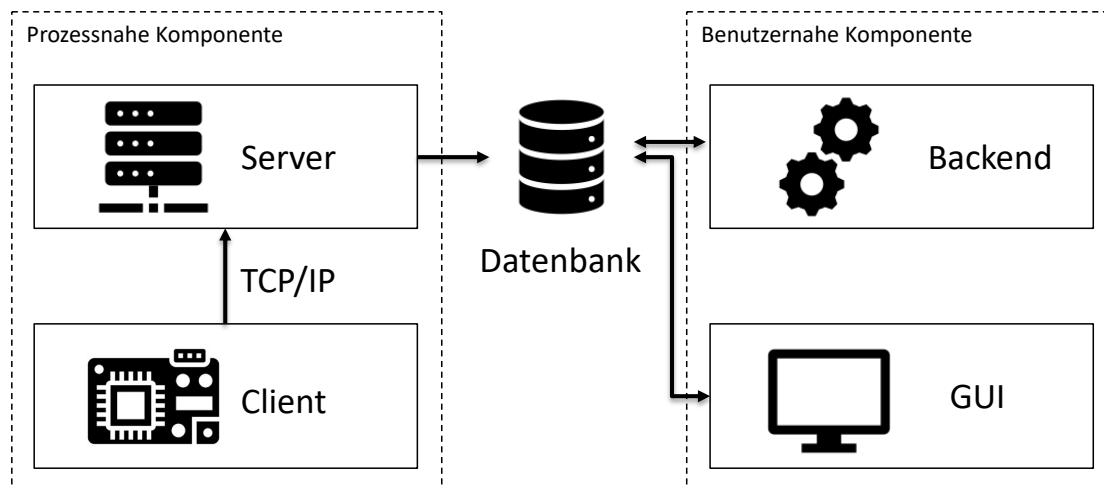


Abbildung 4 Architektur der Prozessnahen-, Benutzernahenkomponenten und der Datenbank

Die Datenbank besteht aus vier Collections mit verschiedenen Zwecken (Abbildung 5). In den Collections „training“ und „process\_data“ werden die Messdaten gespeichert. In „backend“ werden aus die in „Backend“ berechneten für die GUI notwendigen Daten gespeichert. In „labels“ werden die Prozesszustände aus der Trainingsphase gespeichert.

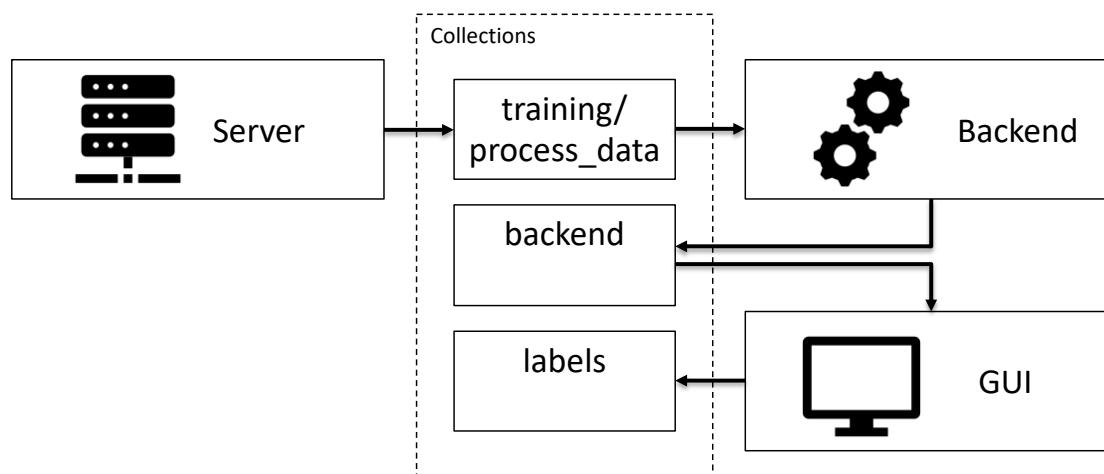


Abbildung 5 Darstellung der Datenbankanbindung

## Klassifikation

Zur Bildung der Klassifizierungsmodelle werden die im Training aufgenommenen Daten verwendet. Eine Messreihe eines Waschgangs ist in Abbildung 6 dargestellt.

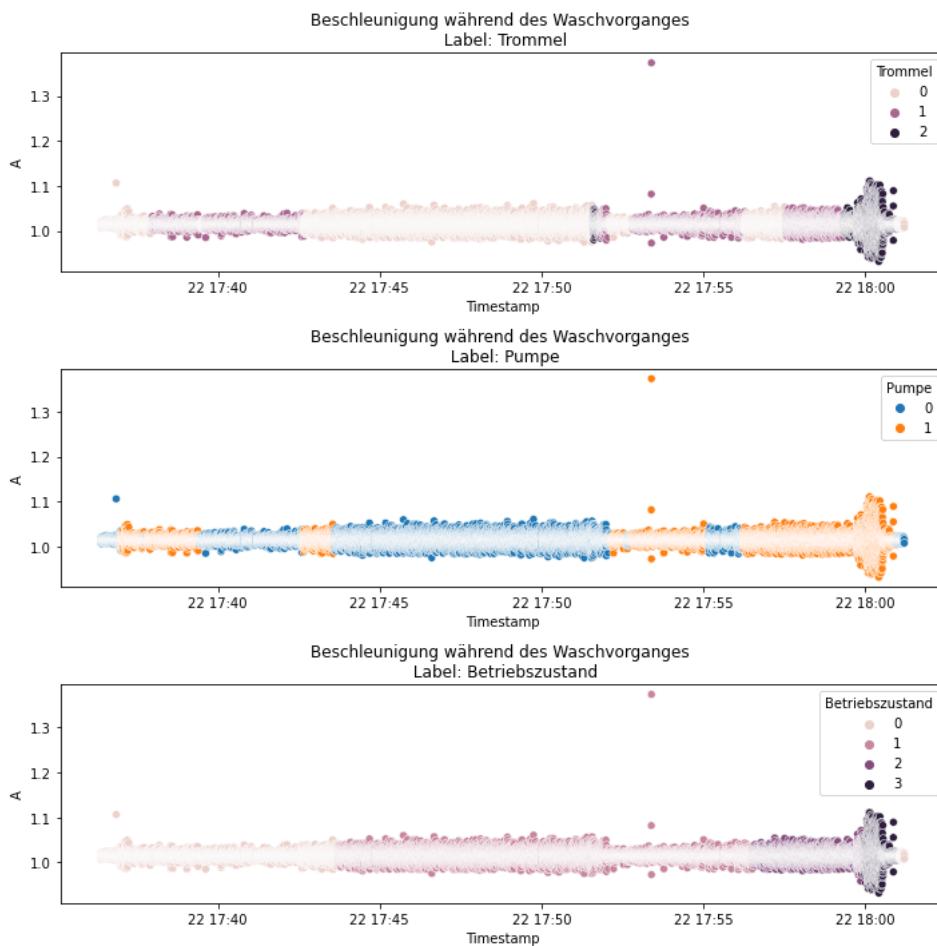
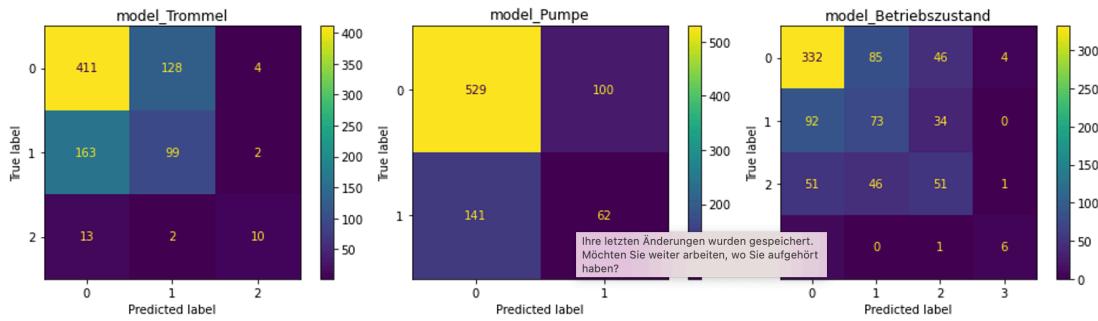


Abbildung 6 Beschleunigung während eines Waschgangs

Es werden drei Klassifikatoren eingesetzt. Jeweils einer zur Erkennung des Trommel-, Betriebs- und Pumpenzustands. In allen drei Fällen wird ein „Multi-layer Perceptron classifier“ (MLPClassifier) des scikitlearn Moduls verwendet.

Die Daten werden im ersten Schritt in Batches mit einer Größe von 64 Datensätzen unterteilt, als zweites Fourier-Transformiert und im dritten Schritt skaliert.

Die Skalierungs- und Klassifizierungsinstanz werden mithilfe von dem Modul „joblib“ gespeichert, damit bei der Anwendung des Models auf diese zugegriffen werden kann.



**62.5 % score for model\_Trommel**

**71.0337 % score for model\_Pumpe**

**55.5288 % score for model\_Betriebszustand**

Abbildung 7 Confusion Matrix und Genauigkeit der Klassifizierungs-Modelle

In Abbildung 7 ist für die drei Modelle jeweils die Confusion Matrix und die erreichte Genauigkeit dargestellt. Beim Trommel-Klassifizierungsmodell (model\_Trommel) fällt auf, dass die Zustände „0“ (Stillstand) und „1“ (Rotieren) häufig verwechselt werden. Dies kann zum einen daran liegen, dass die Daten nicht sauber genug aufgenommen worden sind, da diese beiden Zustände häufig wechseln.

### Benutzeroberfläche

Die Benutzeroberfläche besteht aus zwei Tabs. Das erste Tab „Überwachung“ dient der Prozessüberwachung und zeigt die einzelnen Zustände. Vom ML-Modell wird geschätzt in welchen Zustand sich die Waschmaschine befindet. Die Wahrscheinlichkeiten werden in der Oberfläche als Prozesszustandsdiagramm (links) und farblich abgestuft sowie prozentual (rechts) dargestellt (Abbildung 8).

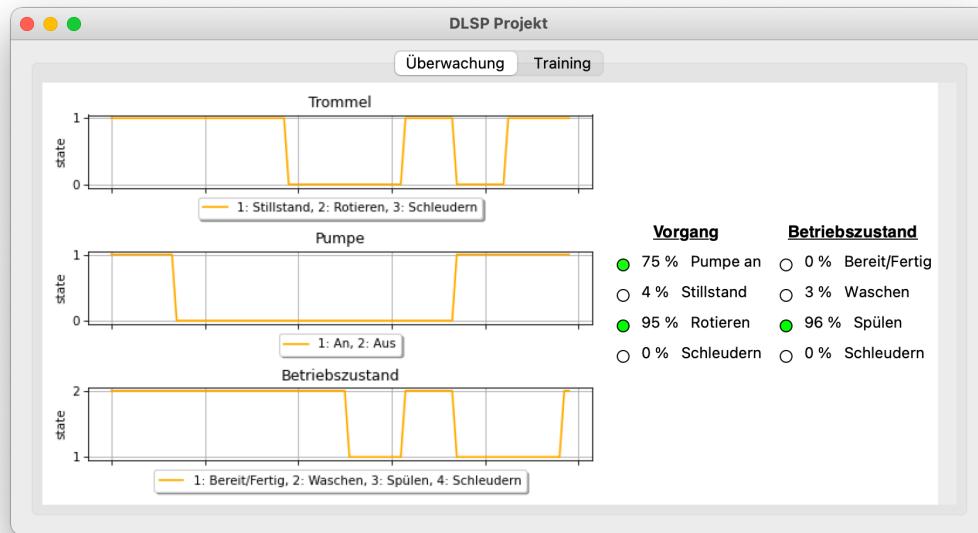


Abbildung 8 GUI Überwachung Tab

Das zweite Tab (Training) dient zur Datenaufnahme für die Trainingsphase des ML-Modells. Hier können die aufgenommenen Daten unmittelbar gelabelt werden.

Außerdem werden die Beschleunigung und die Schwingungen der zuletzt aufgenommenen Messreihe grafisch dargestellt (Abbildung 9).

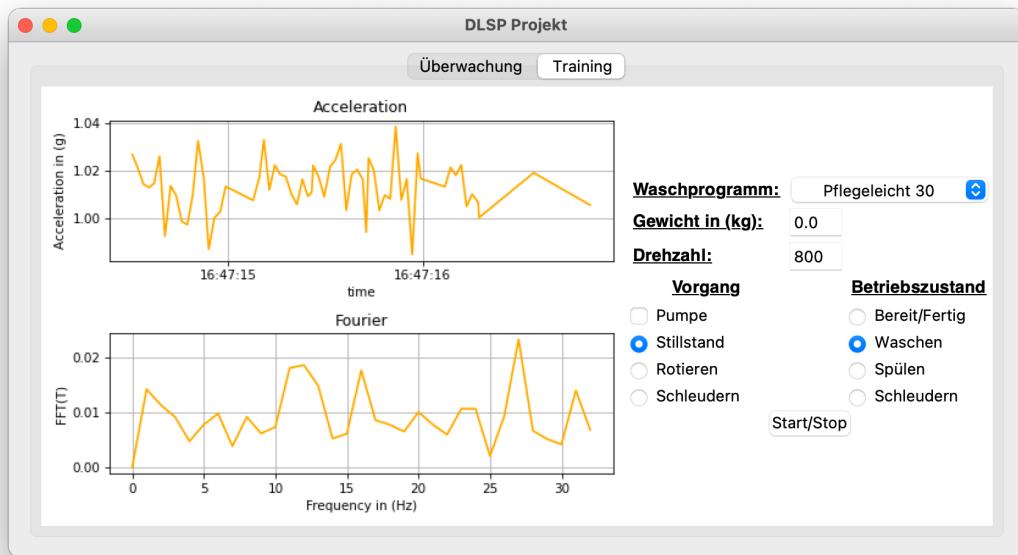


Abbildung 9 GUI Training Tab

### Lessons Learned

1. Strukturierung von Softwareprojekten
2. Entwicklung von NoSQL Datenbanken (MongoDB)
3. Performance Optimierung von Code
4. Anwendung von git

### Bedienungsanleitung

- Installiere requirements.txt
- Den Ordner client auf den ESP8266 flashen
- Zeile "ssid, pwd, ip = "", "", "" anpassen
- Anpassen von folgenden Parametern in src/gateway.py:
  - o self.connection\_string = ""
  - o self.tcp\_ip = ""
- starte main.py on ESP8266
- starte src/processrelated/server.py
- starte src/userrelated/backend.py
- starte src/userrelated/frontend.py