

Projektarbeit

im Modul

Datenmanagement, Leitechnik und statistische Prozesslenkung (81307)

angefertigt am

Fachbereich Maschinenbau und Mechatronik

der FH Aachen

vorgelegt von

Magnus Diepers 3307011 magnus.diepers@alumni.fh-aachen.de

Simon Röhser 3291574 simon.roehser@alumni.fh-aachen.de

Studiengang: Master Industrial Engineering

Betreuender Professor: Prof. Dr. Ing. Stephan Kallweit

Aachen, den 21. Februar 2022

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit ohne fremde Hilfe selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Quellen entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Die Arbeit besteht aus 9 Seiten (ohne Anhang).

Aachen, den 21. Februar 2022



Unterschrift, Diepers



Unterschrift, Röhser

Inhaltsverzeichnis

1	Projektaufgabe	1
2	Prozessbeschreibung	2
3	Theoretische Grundlage der Datenerfassung	3
4	Main-Programm des Mikrocontrollers	4
5	Programme über PyCharm	5
5.1	Datenempfang über TCP und Datenverarbeitung	5
5.2	Visualisierung mithilfe eines Graphical-User-Interface	5
6	Anwendung des maschinellen Lernens	7
7	Hinweise zur Verwendung der Programme	9

Abbildungsverzeichnis

Abbildung 2-1: Verwendete Waschmaschine	2
Abbildung 6-1: Klassifizierungsergebnis mit fünf Zuständen	7
Abbildung 6-2: Klassifizierungsergebnis mit drei Zuständen	8

Symbolverzeichnis

Symbol	Einheit	Bezeichnung
n_{max}	min^{-1}	maximale Drehzahl (Prozess)
f_{max}	Hz	maximale Drehfrequenz (Prozess)
$f_{Abtast_{min}}$	Hz	minimale Abtastfrequenz
f_{Abtast}	Hz	gewählte Abtastfrequenz

Formelverzeichnis

Formel 3-1: Berechnung der maximalen Drehfrequenz der Waschmaschine	3
Formel 3-2: Berechnung der minimalen Abtastfrequenz	3
Formel 3-3: Berechnung des Verhältnisses der Abtastfrequenz zur Eingangsfrequenz.....	3

1 Projektaufgabe

Im Rahmen des Projektes sollen verschiedene Betriebszustände einer Waschmaschine mithilfe eines maschinellen Lern-Algorithmus festgestellt und anschließend über ein Graphical-User-Interface visualisiert werden. Für die Diagnose des aktuellen Betriebszustandes müssen die fachlichen Aspekte, wie beispielsweise die Bestimmung der benötigten Abtastfrequenz, Komprimierungsaspekte und verschiedene Kommunikationsformen genutzt werden.

Die Beschleunigungsdaten sollen mithilfe eines Mikrocontrollers mit einem MPU6050 erfasst werden. Diese sollen einerseits in einer cloudbasierten Datenbank gespeichert werden, um Zugriffe außerhalb des internen Netzwerkes zu ermöglichen, und andererseits dem Graphical-User-Interface übergeben werden, um sie zu visualisieren. Im Rahmen der Visualisierung sollen die Daten mithilfe der Fast-Fourier-Transformation in ihr Frequenzspektrum umgerechnet werden. Hierdurch wird die Phasenverschiebung, die bei den Beschleunigungsdaten zu Schwierigkeiten bei der Anwendung des maschinellen Lernens führen könnte, eliminiert. Das Frequenzspektrum kann anschließend genutzt werden, um den aktuellen Programmstatus mithilfe eines maschinellen Lern-Algorithmus zu erkennen. In dem Graphical-User-Interface soll das Ergebnis des Algorithmus klar visualisiert werden.

Ziel ist es, dass die Studierenden sich mit den grundlegenden Aspekten der Prozessüberwachung und -diagnose befassen und in der Lage sind, einen Prozess von der Aufnahme der Daten bis zur Visualisierung der Untersuchungsergebnisse vollständig automatisiert zu analysieren und zu verstehen.

2 Prozessbeschreibung

Das Projekt wird an einer Siemens-Waschmaschine des Modells IQ500 durchgeführt (siehe Abbildung 2-1). Diese steht auf Gummimatten, die sie von dem Fliesenboden trennen und einen Teil der beim Waschen entstehenden Schwingungen dämpfen, da sie nicht vollends starr sind.



Abbildung 2-1: Verwendete Waschmaschine

Bei dem betrachteten Waschprozess handelt es sich um ein Programm mit einer Länge von 3 Stunden und 3 Minuten. Das Programm beinhaltet eine Vorwäsche, Schleudervorgänge, Einspülen von Wasser und einen langen Waschprozess. Ebenfalls sind zahlreiche Stillstände vorhanden, die einige Sekunden dauern. Daher sollen diese fünf Zustände im maschinellen Lernen erkannt werden. Die maximale Drehzahl der Waschmaschine beträgt 1400 min^{-1} .

3 Theoretische Grundlage der Datenerfassung

Wie im vorangegangenen Kapitel erwähnt, beträgt die maximale Drehzahl der Waschmaschine 1400 min^{-1} .

Formel 3-1: Berechnung der maximalen Drehfrequenz der Waschmaschine

$$f_{\max} = \frac{n_{\max} \left[\frac{1}{\text{min}} \right]}{60 \left[\frac{\text{s}}{\text{min}} \right]} = \frac{1400 \left[\frac{1}{\text{min}} \right]}{60 \left[\frac{\text{s}}{\text{min}} \right]} = 23, \overline{33} \text{ [Hz]} \quad \text{Formel 3-1}$$

Die maximale Drehfrequenz des Waschprogramms beträgt dementsprechend 23,33 Hz.

„Beim Prozess des Abtastens von Signalen kann es zum Aliasing kommen, wenn die Abtastrate kleiner als das Doppelte der höchsten Frequenz ist. Das Eingangssignal kann dann nicht mehr vollständig aus dem abgetasteten Signal rekonstruiert werden.“ [1]

Um die Problematik von fehlerhaften Abtastdaten durch Aliasing zu umgehen, muss die Abtastfrequenz mindestens das doppelte der maximalen Drehfrequenz der Waschmaschine betragen. Es ist aber dennoch sinnvoll, die Abtastfrequenz höher als die doppelte Frequenz des Eingangssignals anzusetzen, da auch bei dem Verhältnis 2:1 (Abtastfrequenz zu Eingangsfrequenz) die Genauigkeit leiden kann.

Formel 3-2: Berechnung der minimalen Abtastfrequenz

$$f_{\text{Abtast}_{\min}} = f_{\max} [\text{Hz}] * 2 = 23, \overline{33} [\text{Hz}] * 2 = 46, \overline{66} \text{ [Hz]} \quad \text{Formel 3-2}$$

Für das Projekt beträgt die minimale Abtastfrequenz 46,66 Hz. Um eine möglichst genaue Abtastung des Eingangssignals zu erreichen, wurde entschieden, mit einer Frequenz von 100 Hz abzutasten.

Formel 3-3: Berechnung des Verhältnisses der Abtastfrequenz zur Eingangsfrequenz

$$\text{Abtastverhältnis} = \frac{f_{\text{Abtast}} [\text{Hz}]}{f_{\max} [\text{Hz}]} = \frac{100 [\text{Hz}]}{23, \overline{33} [\text{Hz}]} = 4,29 \text{ [#]} \quad \text{Formel 3-3}$$

Dadurch beträgt das Verhältnis von Abtastfrequenz zu Eingangsfrequenz 4,29. Der Aliasing-Effekt kann dadurch ausgeschlossen werden.

4 Main-Programm des Mikrocontrollers

Im Main-Programm, das auf dem Mikrocontroller läuft, wird zunächst eine Print-Ausgabe der aktuell auf dem Mikrocontroller geflashten Dateien für eine möglicherweise verbundene REPL erzeugt. Dies soll dem Verwender die Möglichkeit geben, jederzeit einsehen zu können, dass nur die Dateien geflasht sind, die benötigt werden. Anschließend verbindet sich der Mikrocontroller mit dem WLAN und synchronisiert die aktuelle Zeit aus dem Netzwerk. Nachdem die WLAN-Verbindung hergestellt wurde, wird der I²C-Bus initialisiert, mit dem der MPU6050-Sensor angesprochen werden kann. Der Befehl für das Auslesen der Beschleunigungsdaten wird auch bereits in der Initialisierung der Variablen und Funktionen adressiert. Anschließend im Rahmen einer While-Schleife mithilfe einer Try-&Except-Funktion darauf gewartet, dass im Try-Teil eine TCP-Verbindung mit der Server-Seite hergestellt wird. Der Mikrocontroller ist in diesem Fall die Client-Seite der TCP-Kommunikation. Wenn die Verbindung steht, wird ein Timer initialisiert, dessen Callback-Funktion mit der zuvor definierten Abtastfrequenz eine vordefinierte Funktion aufruft. In dieser Funktion zählt ein Datacounter die Anzahl der Durchläufe. Wenn der Datacounter den Wert 200 erreicht hat, wird er auf den Wert 0 zurückgesetzt. Wenn er den Wert 0 hat, wird ein Zeitstempel mit der aktuellen Zeit via TCP gesendet. Bei jedem Aufruf der Funktion wird unabhängig vom Datacounter der aktuelle Beschleunigungswert in allen drei Achsen ermittelt, direkt auf 5 Nachkommastellen formatiert und ebenfalls via TCP übermittelt. Sämtliche über TCP gesendeten Daten werden vor dem Senden als String formatiert und verschlüsselt. Das Programm ist so konzipiert, dass es über die Try-&Except-Funktion automatisch stoppt, wenn die TCP-Verbindung von der Server-Seite terminiert wird.

5 Programme über PyCharm

Parallel zu dem Main-Programm auf dem Mikrocontroller laufen zwei separate Programme über PyCharm, die in den folgenden beiden Unterkapiteln erläutert werden. Damit die Programme fehlerfrei laufen, mussten zunächst verschiedene Packages in PyCharm installiert werden. Diese sind im Ordner „venv“ des PyCharm-Projektes gespeichert.

5.1 Datenempfang über TCP und Datenverarbeitung

Das erste der beiden Programme (DLSP_Datenverarbeitung.py), die über PyCharm laufen, dient dazu, die Serverseite der TCP-Kommunikation zu initialisieren, die vom Mikrocontroller gesendeten Daten zu empfangen, zu verarbeiten und anschließend im gewünschten Format in die Datenbank hochzuladen.

Zunächst wird der Socket erzeugt, gebunden und darauf gewartet, dass der Mikrocontroller sich mit dem Server verbinden möchte. Sobald dies geschehen ist, wird die Verbindung akzeptiert. Anschließend startet eine While-Schleife, die solange durchlaufen wird, bis das Programm manuell geschlossen wird. In der Schleife werden zu nächsten die vom Mikrocontroller gesendeten Daten empfangen und entschlüsselt. Im nächsten Schritt werden die Daten vorverarbeitet. Hierbei wird geschaut, ob die Beschleunigungsdaten im richtigen Format und vollständig übermittelt wurden. Alle defekten Datensätze werden an dieser Stelle rausgefiltert. Für die verbleibenden Datensätze, die keinen im Main-Programm erzeugten Zeitstempel haben, wird ein zugehöriger Zeitstempel errechnet und eingefügt. Nun sind die Daten vorverarbeitet und können in die cloudbasierte MongoDB Atlas Datenbank hochgeladen werden.

5.2 Visualisierung mithilfe eines Graphical-User-Interface

Das Programm besteht aus verschiedenen definierten Funktionen, die ineinander aufgerufen werden. Nach dem Start des Programms wird ein Diagramm der letzten 2.000 in der Datenbank enthaltenen Datensätze in einem

Pop-Up-Fenster geladen, um den Nutzer genauere Einblicke in diese zu ermöglichen. Nachdem der Nutzer dieses Fenster geschlossen hat, lädt das Graphical-User-Interface. Dieses besteht aus zwei Tabs (Fotos befinden sich im Ordner „GUI“). Im ersten Tab (dem standardmäßig auftauchenden nach Programmstart) ist nur ein Dialogfenster zur Kommunikation wichtiger Statusmeldungen und Informationen sowie eine Visualisierung des aktuellen Programmstatus. Das zweite Tabs enthält zusätzlich tiefergehende Informationen. Es verfügt über eine Anzeige der letzten gemessenen Beschleunigungswerte und ein Diagramm zur Visualisierung der letzten 500 Datensätze der Beschleunigungsdaten. Im Dialogfenster werden zusätzlich zu den Informationen, die auch im ersten Tab eingeblendet werden, diagrammspezifische Informationen mit dem Nutzer geteilt. Somit hat der Nutzer selbst die Wahl, wie viel Informationsgehalt er haben möchte. Da die Anzeige des Graphical-User-Interface den Hauptthread, der über PyCharm erzeugt wird, vollständig beansprucht, wird für die restlichen Funktionen ein Nebenthread initialisiert, der durch eine While-Schleife durchgehend aufrechterhalten wird. Das Programm kann über den roten Programmstop-Button gestoppt werden. Das Fenster mit der Graphical-User-Interface muss allerdings manuell zusätzlich geschlossen werden.

6 Anwendung des maschinellen Lernens

Bei der Anwendung des maschinellen Lernens wurden zunächst fünf Zustände klassifiziert und trainiert.

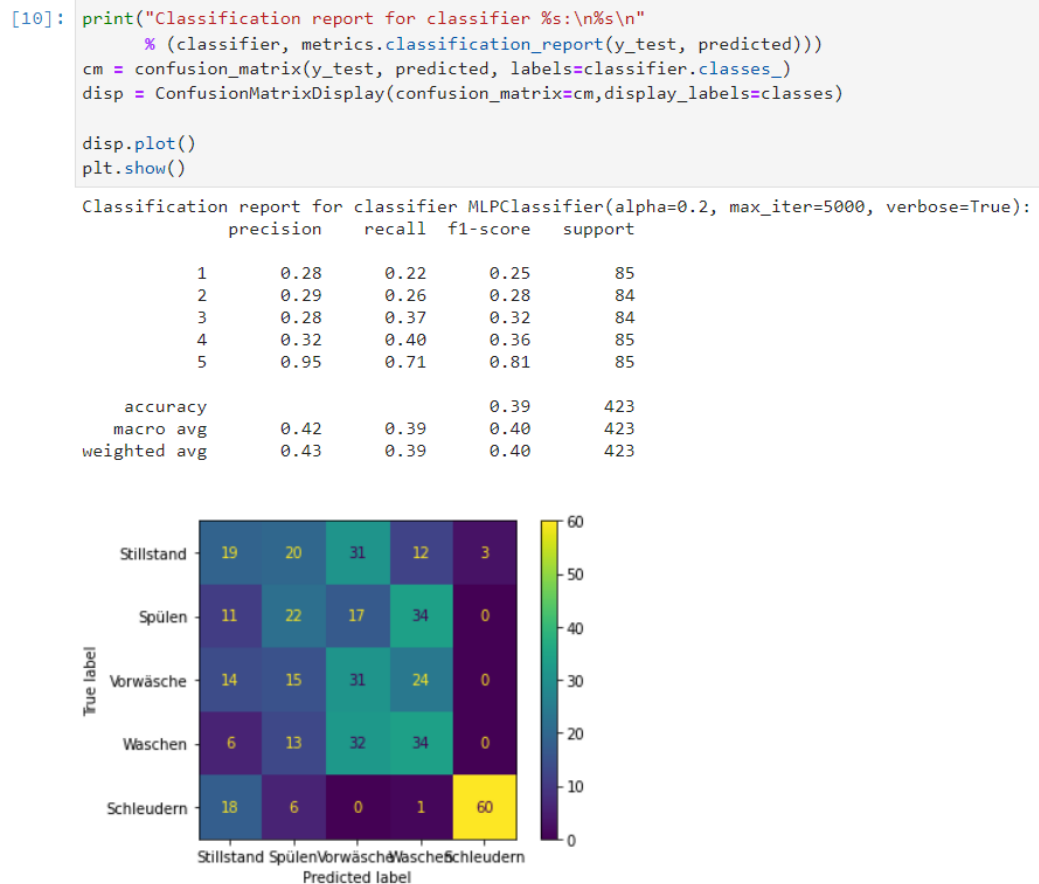


Abbildung 6-1: Klassifizierungsergebnis mit fünf Zuständen

Bei Betrachtung der Ergebnisse fällt auf, dass erhebliche Schwierigkeiten bei der Unterscheidung von Waschen und Vorwäsche auftreten. Zusätzlich ist das Spülen nicht gut klassifizierbar. Aus diesem Grund wurde entschieden, für die Umsetzung im Programm nur drei Zustände zu nutzen. Dies soll die Genauigkeit des Algorithmus erhöhen. Das Ergebnis der Anpassung zeigt diese Steigerung der Genauigkeit deutlich.

Anwendung des maschinellen Lernens

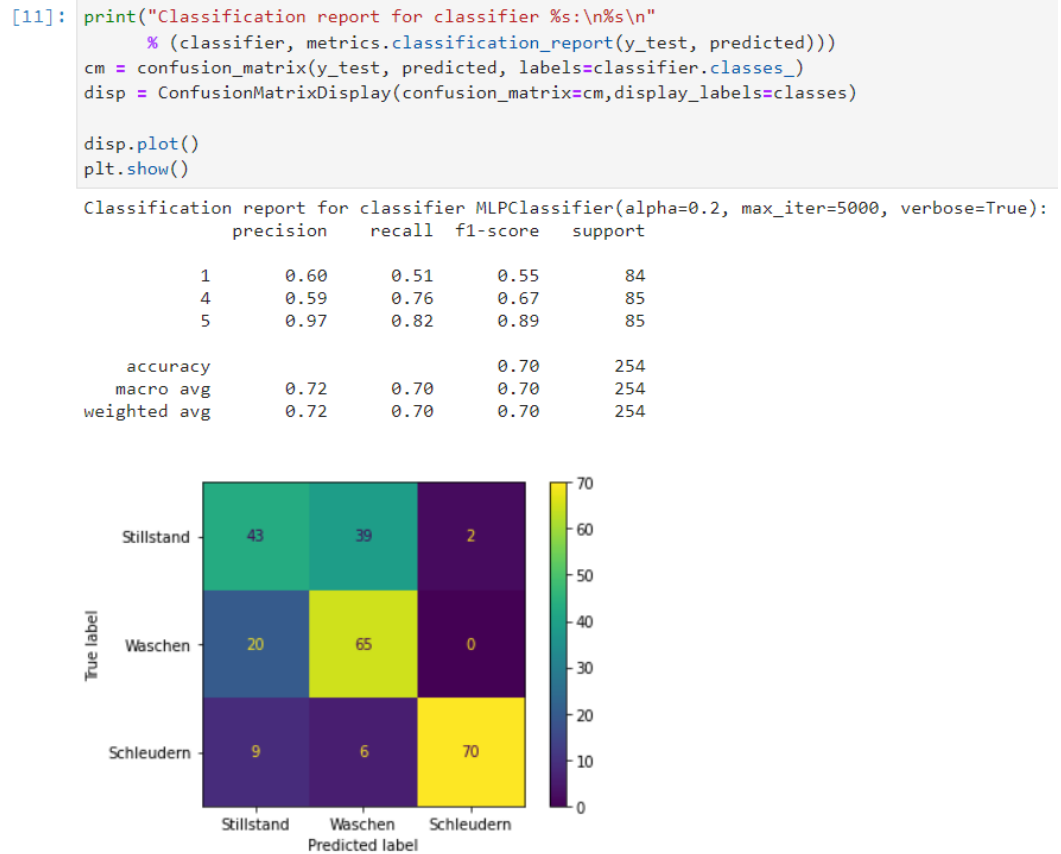


Abbildung 6-2: Klassifizierungsergebnis mit drei Zuständen

7 Hinweise zur Verwendung der Programme

Die Reihenfolge, in der die Programme gestartet werden, ist irrelevant. Alle Programme sind so geschrieben, dass sie auf die benötigten Daten der Schnittstellen warten, ohne in einen Fehler zu laufen. Unter Umständen kann es einige Sekunden dauern, bis das Graphical-User-Interface die Daten im Diagramm visualisiert. Vor dem Start müssen sowohl die Daten für die TCP-IP-Adresse als auch die WLAN-Verbindung eingegeben werden. Diese sind im hochgeladenen Code durch `***` ersetzt.