

Eingebettetes Funksensornetzwerk zur Handlungserkennung beim Menschen mittels künstlicher Intelligenz

Zur Erlangung des akademischen Grades eines

**DOKTORS DER INGENIEURWISSENSCHAFTEN
(Dr.-Ing.)**

von der KIT-Fakultät für
Elektrotechnik und Informationstechnik
des Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

M.Sc. Marc Bruno Schroth

geboren in Pforzheim

Tag der mündlichen Prüfung:

06.03.2025

Hauptreferent:

Prof. Dr. rer. nat. Wilhelm Stork

Korreferent:

Prof. Dr.-Ing. Arne Rönnau

Kurzfassung

In der vorliegenden Arbeit werden Sensornetzwerke zur Handlungserkennung beim Menschen detailliert analysiert und ein darauf basierendes Konzept für ein Sensornetzwerk zur Prozesserfassung beim Kochen erarbeitet, vorgestellt und umgesetzt. Das Sensorkonzept berücksichtigt dabei verschiedene Aspekte der avisierten Zielgruppe der Senior*innen, unter anderem leichte Bedienbarkeit, detaillierte Erkennung einzelner Kochschritte und eine einfach verständliche Unterstützung während des Kochens.

Der inhaltliche Kern wird von einer Signalstärkemessung mittels Sensorknoten an verschiedenen Küchenutensilien gebildet, die während der Zubereitung eines Rezepts der Lokalisierung dienen, um damit die Objektinteraktionen abzubilden. Die Sensorknoten werden hierzu basierend auf einer Anforderungsanalyse eigens spezifiziert, entworfen und implementiert.

Des Weiteren werden Methoden der künstlichen Intelligenz eingesetzt, um aus diesen Signalstärkemessungen das Kochrezept, sowie den aktuellen und nächsten Kochschritt zu prädizieren. KI-Modelle werden hierbei an verschiedenen Stellen des Gesamtsystems eingesetzt: Einerseits serverseitig im Hintergrund, als auch eingebettet auf den Sensorknoten selbst. Auf diese Weise kann den Nutzenden eine unmittelbare Hilfestellung angeboten werden, wenn beispielsweise ein einzelner Teilschritt vergessen wird.

Am Ende der Arbeit wird das entwickelte System in einer realen Küche eingesetzt, um mittels Usability-Testing-Methoden zu überprüfen, in wie weit der Einsatz eines solchen Sensornetzwerks zielführende Unterstützung bieten kann. Hierzu werden von einem Kollektiv an Testpersonen verschiedene Rezepte mit Zutaten gekocht und ein Vergleich zwischen Personen gezogen, die mit und ohne dem entwickelten System kochen.

Abstract

In this thesis, sensor networks for activity recognition in humans are analyzed in detail and on this basis a concept for a sensor network for activity recognition during cooking is developed, presented and implemented. The sensor concept takes various aspects of the intended target group, including ease of use, detailed recognition of individual cooking steps and easy-to-understand support during cooking into account.

The content-related core is formed by a signal strength measurement using sensor nodes on various kitchen utensils, which are used for localization during a recipe in order to map the object interactions. The sensor nodes are specified, designed and implemented based on a requirements analysis.

Furthermore, artificial intelligence methods are used to infer the recipe and also the current and next cooking step from these signal strength measurements. AI models are used at various points in the overall system, both in the background on the server side and embedded in the sensor nodes themselves. In this way, users can be offered immediate assistance if, for example, an individual sub-step is forgotten.

At the end of the thesis, the developed system is used in a real kitchen. By applying usability testing methods, the extent to which the use of such a sensor network can provide targeted support is checked. For this purpose, a group of test subjects cooks various recipes with real ingredients and a comparison is made between people cooking with and without the developed system.

Danksagung

Die vorliegende Dissertation entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am FZI Forschungszentrum Informatik im Forschungsbereich Embedded Systems and Sensors Engineering. Besonders danken möchte ich Prof. Dr. rer. nat Wilhelm Stork sowohl für die Promotionsmöglichkeit, als auch für das entgegengebrachte Vertrauen in meine wissenschaftliche Arbeit. Der Freiraum bei der Gestaltung meiner Dissertation hat meine eigenverantwortliche Arbeitsweise und meine persönliche Entwicklung nachhaltig gestärkt.

Prof. Dr.-Ing. Arne Rönnau danke ich für die Übernahme des Korreferats, das damit verbundene Engagement und die langjährige Begleitung, auch in Themen über die Dissertation hinaus.

Hervorheben möchte ich die außerordentlich gute Zusammenarbeit mit Jens Juhl: Vielen Dank für die gemeinsamen Diskussionen, das Feedback und deinen unvoreingenommenen Blick. Unser Tandem hat den Unterschied gemacht!

Für die unzähligen fachlichen Diskussionen und den wertvollen Austausch möchte ich insbesondere Dennis Birkenmaier und Christina Erler danken. Des Weiteren danke ich Christoph Zimmermann, Dr.-Ing. Lukas Kohout, Dr.-Ing. Johannes Schneider, dem ganzen Team von MIT und den von mir betreuten Studierenden für das kollegiale und inspirierende Arbeitsumfeld, auch gleich zum Start am FZI. Einen ganz besonderen Dank an meine Schwester Julia für ihr offenes Ohr und an meine Eltern Claudia und Heinz, die mich liebevoll bei jedem Schritt meiner Ausbildung unterstützt haben und bereits früh meine Faszination für Wissenschaft und Technik entfachten. Von ganzem Herzen danke ich meiner Frau Nadine, die durch ihre Gelassenheit und Lebensfreude stets ein wichtiger Ruhepol für mich ist und auf deren Unterstützung ich jederzeit voll vertrauen kann. Vielen Dank.

Karlsruhe, im November 2024

Marc Bruno Schroth

Inhaltsverzeichnis

Kurzfassung	i
Abstract	iii
Danksagung	v
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	4
1.3 Umfeld	5
1.4 Gliederung	5
2 Grundlagen und domänen spezifische Rahmenbedingungen	7
2.1 Begriffsgrundlagen	7
2.1.1 Die Handlung als Prozess	8
2.1.2 Aktivitäten messbar machen: Activities of Daily Living	8
2.1.3 Human Activity Recognition zur Erkennung von Handlungen	10
2.1.4 Verknüpfung von Technologie und Umfeld über Ambient Assisted Living	11
2.2 Usability Testing	11
2.2.1 Quantitative und qualitative Befragungen	12
2.2.2 Die System Usability Scale zur Beurteilung der Bedienbarkeit	13
2.2.3 Messung der Belastung mittels Task Load Index	15
2.3 Elektrotechnische Grundlagen und Messtechnik	17
2.3.1 Elektromagnetische Wellen	17

2.3.2	Signalstärke und RSSI	18
2.4	Sensoren und Sensornetzwerke	19
2.4.1	Sensorik	19
2.4.2	Sensornetzwerke	22
2.5	Künstliche Intelligenz	26
2.5.1	Feature Engineering	27
2.5.2	Performanzmetriken	28
2.5.3	Klassische Methoden des maschinellen Lernens	30
2.5.4	Künstliche neuronale Netze	34
3	Einordnung in den Stand der Wissenschaft und Technik	37
3.1	Sensorsysteme zur Handlungserkennung	38
3.1.1	Sensoren und Systeme	38
3.1.2	Sensorsysteme mit direktem Nutzendenfeedback	41
3.2	Künstliche Intelligenz für smarte Sensorsysteme	44
3.2.1	Datensets	44
3.2.2	Architekturen und Methoden	48
3.3	Usability Methoden zur Testung der Gebrauchstauglichkeit	52
3.4	Diskussion und Fazit	55
4	Forschungsfrage zur Überwindung des aktuellen Stands der Wissenschaft und Technik	59
5	Entwurf eines KI-gestützten Sensor-Netzwerks zur Prozessoptimierung	61
5.1	Konzeptionierung des Gesamtsystems	61
5.2	Anforderungen an das Gesamtsystem	65
6	Entwicklung des Sensor-Netzwerks	69
6.1	Auswahl einer geeigneten Lokalisierungstechnologie	69
6.2	Auswahl des Feedbackkanals	72
6.3	Spezifizierung der Sensorknoten	74
6.4	Design und Implementierung der Sensorknoten	76
6.4.1	Energieversorgung und Ladeschaltung	76
6.4.2	Accelerometer und Feedbackkanal	79
6.4.3	Antennen	80

6.4.4	Gesamtschaltung	81
6.4.5	Sensorknotensoftware	85
6.5	Vorabtestung der Sensorknoten	88
7	Entwicklung der Machine Learning Komponenten	93
7.1	Einsatzgebiete von Machine Learning Algorithmen im System . .	94
7.2	Konzeptionierung des Machine Learning Hintergrundsystems . .	94
7.3	Implementierung des Hintergrundsystems	97
7.4	Erster technischer Pratest	99
7.5	Optimierung der Prozessanalyse	105
7.6	Konzeptionierung des eingebetteten ML-Systems	108
7.7	Implementierung des eingebetteten ML-Systems	110
7.8	Zweiter technischer Pratest	115
8	Evaluation und Ergebnisdiskussion	119
8.1	Messaufbau und Ablauf	119
8.1.1	Rezept 1: Gednste Karotten	122
8.1.2	Rezept 2: Gemsesuppe	124
8.1.3	Rezept 3: Nudeln mit Tomaten	125
8.2	bersicht der Messungen	128
8.3	Ergebnisse der Befragung	128
8.3.1	Auswertung und Diskussion der System Usability Scale .	129
8.3.2	Auswertung und Diskussion des nicht-standardisierten Fragebogens	133
8.3.3	Auswertung und Diskussion zum Raw Task Load Index .	136
8.3.4	Sonstige Beobachtungen	142
9	Zusammenfassung und Ausblick	143
9.1	Zusammenfassung	143
9.2	Ausblick	147
A	Anhang	149
A.1	Testrezepte	150
A.2	SUS Fragebogen	155
A.3	RTLX Fragebogen	156
A.4	Nicht-standardisierter Fragebogen	157

Abbildungsverzeichnis	159
Tabellenverzeichnis	161
Eigene Veröffentlichungen	163
Konferenzbeiträge	163
Literaturverzeichnis	165

Abkürzungsverzeichnis

AAL	Ambient Assisted Living	ID	Identität
ADL	Activities of Daily Living	IMU	Inertial Measurement Unit
AmI	Ambient Intelligence	KI	Künstliche Intelligenz
BLE	Bluetooth Low Energy	KNN	K-Nearest-Neighbour
CART	Classification And Regression Tree	LED	Light-Emitting Diode
		LDO	Low Dropout
CNN	Convolutional Neural Network	LiPo	Lithium-Polymer
dBm	Dezibel Milliwatt	LSTM	Long Short-Term Memory
DT	Decision Tree	mcd	Milicandela
EU	Europäische Union	MEMS	Mikroelektromechanisches System
FN	False Negatives		
FP	False Positives	ML	Machine Learning
FZI	FZI Forschungszentrum Informatik	MLP	Multi-Layer-Perzeptron
		NASA	National Aeronautics and Space Administration
NB	Gaussian Naive Bayes		
HAR	Human Activity Recognition / Human Action Recognition	RF	Random Forest

RFECV	Recursive Feature Elimination with Cross Validation
RNN	Recurrent Neural Network
RTLX	Raw Task Load Index
SoC	System on Chip
SUS	System Usability Scale
SVM	Support Vector Machine
TLX	Task Load Index
TN	True Negatives
TP	True Positives
UART	Universal Asynchronous Receiver / Transmitter
μC	Mikrocontroller

1 Einleitung

Zur Einordnung dieser Dissertation beginnt die Arbeit mit einer Motivation, die sich im demografischen Wandel begründet und die damit zusammenhängenden Herausforderungen für die Gesellschaft und insbesondere der Pflegekassen aufzeigt. Darauf aufbauend folgt die Zielsetzung der Arbeit und die Einbettung in das Umfeld, in der diese Arbeit angefertigt wurde. Das Kapitel schließt mit einer Übersicht zur Gliederung.

1.1 Motivation

Die Vereinten Nationen sagen eine Verdreifachung der Personen über 65 Jahren bis zum Jahr 2050 voraus, in absoluten Zahlen entspräche dies 1,5 Milliarden Menschen [137]. Mit einer zunehmend älteren Gesellschaft steigt auch die Zahl der chronisch erkrankten Personen und die Anzahl der Gesundheitsnotfälle [145]. Durch eine sinkende Geburtenrate [24], vor allem in Industriestaaten, steigt damit weiterhin der Abhängigenquotient, d.h. der Anteil der nicht-erwerbsfähigen Personen zu den erwerbsfähigen Personen vergrößert sich [33].

Um dieser Herausforderung zu begegnen wurden innerhalb der Europäischen Union verschiedene Förderprogramme aufgelegt und Arbeitsgruppen gegründet, beispielsweise die *Ambient Assisted Living (AAL) Joint Association*. Im Unterschied zu anderen Ansätzen, bei denen z.B. sozio-ökonomische Anreize zu Verbesserung des Abhängigenquotienten im Mittelpunkt stehen, stellt bei der AAL Joint Association die technologische Entwicklung ein zentraler Lösungsansatz dar. Mittels *Ambient Assisted Living* bzw. *Ambient Intelligence* soll eine Umgebung mit Technologie angereichert werden [34], um so kontextsensitive Lösungen

[40] zu schaffen, die unmittelbar dazu beitragen sollen, eine Reihe von im AAL Europe-Programm ausgerufenen Zielen zu lösen. Hierbei werden unter anderem die folgenden genannt [2]:

- „extending the time people can live **in their preferred environment** by increasing their **autonomy, self-confidence**, and mobility“ (1),
- „**maintaining health and functional capability** of the elderly individuals“ (2),
- „enhancing security, **prevent social isolation** and support the maintenance of a multifunctional network around the individual“ (3).

Zentrale Ziele sind demnach die *Verlängerung der Zeit*, in der eine Person in ihrer gewünschten Umgebung leben kann, bei gleichzeitiger Beibehaltung oder Erhöhung der *Autonomie und Selbstbestimmtheit* (1), das *Aufrechterhalten der Gesundheit und der funktionalen Fähigkeiten* (2) und die *Vermeidung von sozialer Isolation* (3). Das Erreichen dieser Ziele würde für betroffene Personen eine ganzheitliche Verbesserung ihrer Situation bedeuten, da diese nicht nur für einzelne Domänen des Alltags, wie Gesundheit und Wohlbefinden gesteckt sind, sondern verschiedene Aspekte und Dimensionen des täglichen Lebens berücksichtigt werden. Diese Ziele decken sich mit Erhebungen von Reissmann und Wagner, nach denen sich Hochaltrige unter anderem eine Förderung der Selbstbestimmung, Möglichkeiten zur sozialer Teilhabe und sozialen Kontakts und Barrierereduzierungen wünschen [85].

Zur objektiven Messung dieser Ziele können die *Activities of Daily Living (ADL)* herangezogen werden [61]. ADLs stammen aus den Pflegewissenschaften und beschreiben die Eigenständigkeit einer Person in Bezug auf verschiedene Aktivitäten. Speziell die *Instrumental ADLs* (in Abgrenzung zu den *Basic ADLs*) können quantifizieren, inwiefern eine Person in der Lage ist, ein unabhängiges Leben zu führen [15] [143].

Ein zentraler Punkt dieser ADLs ist das Zubereiten von Nahrung; verliert eine Person diese Fähigkeit, leidet die Selbstständigkeit stark - es ist also wünschenswert, eine Person beim Kochen zu unterstützen, um ihre Selbstbestimmtheit und

Autonomie zu erhalten, wodurch die Person länger in ihrer gewünschten Umgebung, z.B. Zuhause, leben kann. Da Kochen oftmals auch eine soziale Tätigkeit darstellt, beispielsweise im familiären Umfeld oder in Wohnheimgemeinschaftsküchen, kann eine Unterstützung auch helfen, sozialer Isolation vorzubeugen [55]. Dies ist von besonderer Bedeutung, da gerade soziale Aktivitäten dabei nach Sun et al. in aktuellen AAL-Systemen vernachlässigt werden [131]. Weiterhin erfordert das Kochen Konzentration und verschiedene motorische Fähigkeiten, eine Unterstützung stärkt also diese Aspekte und trägt so zur Gesunderhaltung bei [115].

Zusätzlich ergeben sich in einer alternden Gesellschaft direkte ökonomische Aspekte: Eine Person ab Pflegegrad 2 kann in Deutschland aus der Pflegekasse im Jahr 2024 monatlich 761 € Zuschuss für ambulante Pflegesachleistungen erhalten, in Pflegegrad 5 sogar 2200 € [3]. Zu diesen Leistungen zählt auch die Unterstützung beim Zubereiten von Mahlzeiten [126]. Bereits im Jahr 2023 waren über 3,5 Millionen ambulant pflegebedürftige Personen mit Pflegegrad 2 oder höher eingestuft [25]. Daher ist zu erwarten, dass sehr viele Pflegebedürftige einen solchen Zuschuss insbesondere für die Nahrungszubereitung erhalten, wodurch Kosten in Millionenhöhe für die Pflegekassen entstehen. Ein System, welches den Bedarf an Betreuung bei der Nahrungszubereitung verringern kann, ermöglicht es somit unmittelbar, die volkswirtschaftlichen Kosten zu senken.

1.2 Zielsetzung

Diese Arbeit soll einen Beitrag dazu leisten, die Selbstständigkeit von Personen im Alltag zu erhöhen. Der Schwerpunkt wurde hierfür auf Kochszenarien gelegt, welche, wie bereits in Kapitel 1.1 diskutiert, einen zentralen Bestandteil hinsichtlich der Selbstständigkeit einnehmen. Hierfür wird ein Sensornetzwerk entwickelt, welches die Aktivitäten beim Kochen erfassen kann und Hilfestellung anbietet. Die Sensorknoten sind zu diesem Zweck an verschiedenen Küchenutensilien angebracht. Die Sensorplattform wird im Rahmen dieser Arbeit eigens entwickelt und umgesetzt, um Randbedingungen und Herausforderungen der Zielszenarien, wie beispielsweise Energieverbrauch und Abmessungen, bestmöglich begegnen zu können.

Weiterhin hat die Arbeit zum Ziel, die Daten aus dem Netzwerk mittels künstlicher Intelligenz für eine Lokalisation der Sensoren zum Zwecke einer Interaktionsanalyse nutzbar zu machen. Dabei werden bestehende KI-Architekturen genutzt und für den vorliegenden Anwendungsfall adaptiert beziehungsweise auf deren Basis KI-Modelle trainiert werden. Die Datensammlung zum Training basiert auf eigens aufgenommen oder generierten Daten, da für den vorliegenden Fall keine zugänglichen Datensätze zur Verfügung standen, die in Format, Sensorauswahl und Güte für diese Arbeit geeignet waren (vgl. Kapitel 3.2.1).

Auf dieser Grundlage werden Konzepte für die Handlungserkennung von Menschen basierend auf Objektinteraktionen entwickelt, wobei die Teilschritterkennung detailliert genug sein soll, um zur Laufzeit Unterstützung beim Kochen bieten zu können.

Eine Bewertung der technischen Fähigkeiten findet in einer Reihe von Vorabtests für die einzelnen Teilsysteme statt, die Gesamtbewertung erfolgt im Sinne einer nutzendenzentrierten Entwicklung mittels *Usability Testing* und standardisierten Vorgehen.

1.3 Umfeld

Ausgangspunkt für die vorliegende Dissertation war das vom Bundesministerium für Bildung und Forschung geförderte Projekt *AuRorA*, in welchem ein einarmiger Roboter an einer Küchenzeile und eine intelligente Küchenumgebung erforscht wurden. In diesem Zusammenhang entstanden verschiedene *Smart Devices*, wie eine Küchenwaage, Salz- und Pfefferstreuer und eine Gestenerkennung mittels Tiefenbildkamera. Letztere sollte eingesetzt werden, um den aktuellen Kochschritt zu erkennen, für eine kontextsensitive Rückmeldung und Unterstützungen der Nutzenden. Die Gestenerkennung allein auf Basis der Tiefenbildkamera gestaltete sich schwierig, da sie aufgrund von Verdeckungen verschiedener Körperteile nicht hinreichend genau war, oder durch Reflexionen der Time of Flight-Messung die Messgenauigkeit der Tiefeninformation sank. Hinzu kamen Vorbehalte gegenüber dem Kamerasystem aus Gründen der Privatsphäre. So entstanden erste Ideen, die Kochhandlungen unabhängig von einer Kamera mittels mobiler Funksensorik zu erfassen und zu analysieren. Diese Ideen mündeten letztlich in der Entwicklung des hier vorgestellten, implementierten und evaluierten Systems.

1.4 Gliederung

Die vorliegende Dissertation ist in insgesamt neun Kapitel gegliedert. Nach der hiermit abgeschlossenen Einleitung schließt sich mit Kapitel 2, *Grundlagen und domänenspezifische Rahmenbedingungen*, sowohl eine Definition der notwendigen Begriffe als auch Erläuterungen zu den benötigten Grundlagen. Hierzu zählen Begriffe aus der Domäne sowohl in technologischer Hinsicht, als auch im Hinblick auf die spätere Evaluation. Weiterhin werden (elektro-)technische Grundlagen erläutert, um Verständnis für Funkkommunikation, Sensorik und künstliche Intelligenz zu schaffen.

In Kapitel 3, *Einordnung in den Stand der Wissenschaft und Technik*, erfolgt die Darlegung des aktuellen technologischen Status der Entwicklungen im Hinblick

auf Sensorik und Sensornetzwerken in Wissenschaft und Industrie, von künstlicher Intelligenz für die Anwendungsdomäne und der Aktivitätserkennung im Allgemeinen. Abschließend findet eine Abgrenzung des hier in dieser Arbeit entwickelten Systems vom aktuellen Stand der Wissenschaft und Technik statt.

Darauf aufbauend werden in Kapitel 4, *Forschungsfrage*, mehrere Fragen formuliert, davon eine Hauptfrage und drei Nebenfragen, die im Rahmen dieser Arbeit erforscht und beantwortet werden sollen.

Die grundlegenden Ideen und Gedanken zum *Entwurf eines KI-gestützten Sensornetzwerks zur Prozessoptimierung* werden in Kapitel 5 dargestellt. Der Abschnitt beinhaltet neben der Konzeptionierung auch eine Formalisierung in Form einer Anforderungsanalyse an die Gesamt- und Teilsysteme.

Es folgt in Kapitel 6, *Entwicklung des Sensornetzwerks*, die Beschreibung der durchgeführten Arbeiten beginnend bei der Technologieauswahl hin zu einem funktionsfähigen Sensornetzwerk. Dies beinhaltet die einzelnen Schritte von Spezifizierung und Design der Hardware bis hin zur Implementierung und Bestückung.

Daran schließt sich Kapitel 7, *Entwicklung des KI-Systems*, an. Angelehnt an das vorherige Kapitel werden hier für die Komponenten der künstlichen Intelligenz die Auswahl und die Implementierung der KI in das Gesamtsystem beschrieben.

Nach Abschluss der Umsetzungsarbeiten folgt in Kapitel 8, *Evaluation und Ergebnisdiskussion*, die Versuchsdurchführung. Das Sensornetzwerk wird in einer realen Küchenumgebung zum Kochen von Rezepten mit verschiedenen Proband*innen eingesetzt und mittels Usability Testing Verfahren evaluiert. Der Messaufbau und die Durchführung werden in diesem Kapitel dargelegt, die Ergebnisse dokumentiert und diskutiert.

Abschließend wird in Kapitel 9, *Zusammenfassung und Ausblick*, die Arbeit zusammenfassend dargestellt und Weiterentwicklungsmöglichkeiten aufgezeigt.

2 Grundlagen und domänenspezifische Rahmenbedingungen

Dieses Kapitel beschreibt die notwendigen Grundlagen, die zum Verständnis dieser Dissertationsschrift notwendig sind. Hierfür werden zunächst grundlegende Begriffe erläutert, anschließend Methoden des Usability Testing eingeführt und darauffolgend (elektro-)technische Grundlagen gelegt. Das Kapitel wird mit einer Erläuterung zu den in dieser Arbeit angewandten Algorithmen der künstlichen Intelligenz abgeschlossen.

2.1 Begriffsgrundlagen

Nachfolgend werden alle grundlegenden Begriffe, die zum Verständnis dieser Dissertation notwendig sind, beschrieben und definiert. Hierzu wird zunächst die grundlegende Begrifflichkeit, auf der die restlichen Arbeiten fußen, der *Prozess* definiert und seine Bedeutung für die Systementwicklung herausgestellt. Zur Einordnung werden anschließend die Begriffe *Ambient Assisted Living* und *Activities of Daily Living* erläutert, die essentiell für die Umgebung sind, in der das System eingesetzt werden soll. Abschließend werden noch die Begriffe *Human Activity Recognition* und *Ambient Assisted Living* eingeführt, die das Einsatzgebiet des Gesamtsystems weiter konkretisieren. Das entwickelte System ist keineswegs auf eine bestimmte Domäne beschränkt, sondern kann im Gegenteil in einer Reihe von Anwendungsbereichen eingesetzt werden. Weiterhin werden in diesem Kapitel

die Grundlagen des *Usability Testing* gelegt, deren Methoden in der Evaluation zum Einsatz kommen.

2.1.1 Die Handlung als Prozess

Nach DIN IEC 60050-351 ist ein Prozess (lat. procedere, "vorwärts gehen") definiert als die "*Gesamtheit von aufeinander einwirkenden Vorgängen in einem System, durch die Materie, Energie oder Information umgeformt, transportiert oder gespeichert wird*" [1].

Diese allgemeine Beschreibung zugrundelegend, wird im Rahmen dieser Arbeit ein Prozess als eine Abfolge von Handlungen definiert, die in ihrer Gesamtheit den Prozess als solchen bestimmen. Ein Teilschritt für sich kann dabei potentiell selbst ein komplexer Prozess sein. Im Sinne des in dieser Arbeit beschriebenen Systems bilden sie jedoch die kleinste Einheit, die als Teilschritt erkennbar ist.

2.1.2 Aktivitäten messbar machen: Activities of Daily Living

Eine konkrete Handlung, wie in Kapitel 2.1.1 definiert, kann beispielsweise das *Kochen* sein: Ein Prozess, der aus einer Vielzahl von Teilschritten bestehen kann, je nach Rezept und individuellen Unterschieden. Im Kontext der Pflegewissenschaften spielen eben jene Handlungen, oder auch *Aktivitäten*, eine wichtige Rolle zur Beurteilung des psychischen und physischen Zustands einer Person.

Die sogenannten *Aktivitäten des täglichen Lebens* (engl. *Activities of Daily Living*, kurz *ADL*) bezeichnen eine Möglichkeit, um zu beschreiben, wie selbstständig eine Person hinsichtlich verschiedener Kategorien ist. In den Pflegewissenschaften ist der Begriff etabliert und wird herangezogen, um beispielsweise bei Erkrankungen oder im Alter systematisch verschiedene Kategorien von menschlichen Bedürfnissen auf ihren Grad der selbstständigen Ausführung durch die Person zu überprüfen [38]. Mit der Kategorisierung ist im weiteren Verlauf eine Pflegeplanung möglich, um die Selbstständigkeit der Person in den verschiedenen

Bereichen zu erhalten oder wiederherzustellen.

Prinzipiell kann zwischen grundlegenden (engl. *basic*) und instrumentellen (engl. *instrumental*) ADLs unterschieden werden, je nach Anwendungsgebiet und evaluierender Domäne können einzelne Kategorien entfallen oder hinzukommen. Typischerweise zählen zu den grundlegenden Aktivitäten [61]:

- Essen
- Ankleiden
- Waschen
- Transferieren
- Toilettengang
- Kontinenz

Unter *Transferieren* wird die Mobilität einer Person zusammengefasst, z.B. ohne fremde Hilfe aus dem Bett zu kommen.

Die instrumentellen ADLs nach der Lawton-Skala¹ enthalten die nachfolgenden, komplexeren Aktivitäten und beschreiben laut Bookmann und Williams die Fähigkeit einer Person, selbstständig in einer Gemeinschaft zu leben [49] [15] [143]:

- Benutzen des Telefons
- Einkaufen
- Tätigkeiten im Haushalt erledigen
- Wäsche waschen

¹ Die Lawton-Skala bewertet die Unabhängigkeit von Personen anhand der genannten Kategorien mittels einer Einschätzung der funktionalen Fähigkeiten und einer entsprechenden Punktevergabe, wobei höchstens 8 Punkte erreicht werden können. 0 Punkte entsprechen einer hohen Abhängigkeit, 8 Punkten einer hohen Unabhängigkeit. Eine Handreichung zur Anwendung in der Praxis findet sich unter <https://www.msdmanuals.com/de/profi/multimedia/table/lawton-skala-der-instrumentellen-aktivitaeten-des-taeglichen-lebens>

- Fortbewegung mit Ausnahme des Gehens
- Einnahme von verschriebenen Medikamenten nach Anweisung
- Umgang mit Geld
- Zubereitung von Essen

2.1.3 Human Activity Recognition zur Erkennung von Handlungen

Um Handlungen technisch erfassbar zu machen, darunter fallen auch die zuvor erwähnten ADLs (Kap. 2.1.2), spielen Techniken zur Aktivitätserkennung beim Menschen (engl. *Human Activity Recognition* oder auch *Human Action Recognition*, kurz *HAR*) eine entscheidende Rolle.

Es gibt verschiedene Ansätze menschliche Aktivität sensorisch zu erfassen. Zu den gebräuchlichsten Kategorien zählen *am Körper getragene* (engl. *wearable*), *an Gegenständen angebrachte* (engl. *object tagged*) und *gerätefreie* (engl. *device-free*) Systeme [54]. Am Körper getragene Sensoren werden hierfür oftmals an Extremitäten angebracht, um beispielsweise Hand- und Fußbewegungen mittels Accelerometern zu erfassen. An Gebrauchsgegenständen angebrachte Sensoren dienen der Interaktionsanalyse zwischen den benutzen Objekten, um beispielsweise ein oftmals als lästig empfundenes An-, Ausziehen und Tragen von Sensoren im *wearable* Fall zu vermeiden. Letzteres kann auch in der Kategorie *device-free* vermieden werden, zu dieser zählen unter anderem auch kamerabasierte Systeme [125], die üblicherweise eine Installation und Kalibrierung benötigen, um mittels Bildanalysetechniken die Handlungen der Personen in Videoaufnahmen zu erfassen.

2.1.4 Verknüpfung von Technologie und Umfeld über Ambient Assisted Living

Unter *Ambient Assisted Living (AAL)* (deutsch Altersgerechte Assistenzsysteme) werden technische Assistenzsysteme verstanden, um Alltagsgegenstände mit dem Umfeld von hilfsbedürftigen Menschen zu verknüpfen. Ziel ist dabei Förderung und Erhalt der Selbstständigkeit von oftmals älteren, unterstützungsbedürftigen Personen durch technische Systeme im häuslichen Umfeld [46].

Ambient Assisted Living ist dabei eng verknüpft mit dem Begriff der *Ambient Intelligence (AmI)*, bei der eine „*unaufdringliche, vernetzte, anpassungsfähige, dynamische, eingebettete und intelligente Umgebung*“ geschaffen werden soll [117]. Hierzu wird in der Ambient Intelligence eine Umgebung mit Technologie angereichert [34], wodurch beispielsweise das Zuhause zu einem *Smart Home* wird [65]. Über das Einbringen von Technologie werden kontextsensitive Lösungen geschaffen, die es ermöglichen, individuelle Hilfsangebote anzubieten [40].

2.2 Usability Testing

Usability Testing beschreibt einen strukturierten Ansatz beim nutzendenzentrierten Entwurf eines Produkts, bei dem eben jenes anhand der Zielgruppe getestet wird. Die zukünftigen Nutzenden können auf diese Weise durch ihre Rückmeldungen direkten Einfluss auf die Gestaltung nehmen, wodurch letztlich die Marktakzeptanz erhöht werden soll [98]. Im Fokus stehen dabei Untersuchungen, die messbar machen sollen, in wie weit das Produkt den Wünschen und Anforderungen der Nutzendengruppe entspricht. Diese Herangehensweise ist weit verbreitet und wird beispielsweise bei der Entwicklung von Nahrungsmitteln angewandt, aber auch in technischen Domänen wie bei der Entwicklung von Webseiten oder Gerätschaften [37].

In den folgenden Unterkapiteln werden die wichtigsten Grundlagen des Usability Testings, die zum Verständnis der Evaluation dieser Arbeit benötigt werden, beschrieben.

2.2.1 Quantitative und qualitative Befragungen

Die quantitative Forschung hat zum Ziel, Zusammenhänge zu beschreiben, indem messbare Aussagen getroffen werden. Dabei kommen standardisierte Fragebögen zum Einsatz, die einer Probandengruppe vorgelegt werden, die möglichst einer Stichprobe der eigentlichen Zielgruppe entspricht. Grundsätzlich soll also vom Allgemeinen auf das Spezielle geschlossen werden. Demgegenüber steht die qualitative Forschung, bei der die subjektive Wahrnehmung einzelner Personen im Vordergrund steht, woraus sich dann vom Speziellen auf das Allgemeine schließen lassen soll [121].

Die qualitative Forschung bedient sich somit der *Induktion* als Vorgehen um neue Hypothesen aufstellen zu können, wohingegen die quantitative Forschung einen *deduktiven* Ansatz verfolgt, um eine zuvor aufgestellte Hypothese verifizieren oder falsifizieren zu können [23].

Beim Einsatz von Fragebögen als Untersuchungsmittel wird dabei häufig auf eine **Likert-Skala** zurückgegriffen.

Die Likert-Skala ist eine psychometrische Ratingskala, die oft in Befragung eingesetzt wird und nach Rensis Likert benannt wurde [78]. Typisch für diese Skala ist, dass den Versuchspersonen eine oder mehrere Aussagen, häufig schriftlich, vorgelegt werden, zu denen dann die Person ihren Grad der Zustimmung mitteilen soll. Zur Auswahl stehen üblicherweise Intervalle von „trifft überhaupt nicht zu“ bis „trifft voll und ganz zu“.

Die Größe dieser Intervalle ist variabel, oftmals werden ungerade Skalen, z.B. mit 5 oder 7 Stufen, eingesetzt, bei denen die Versuchsperson auch einen neutrale Position einnehmen kann. Soll die Versuchsperson hin zu einer positiven oder negativen Zustimmung gezwungen werden, so können auch gerade Intervalle, beispielsweise mit 4 oder 6 Stufen gewählt werden. Hierdurch wird vermieden, dass die Ergebnisse weniger aussagekräftig sind, wenn befürchtet wird, dass sich die Probandengruppe ungern positionieren möchte [107].

2.2.2 Die System Usability Scale zur Beurteilung der Bedienbarkeit

Die System Usability Scale (SUS) wurde Mitte der 1980er Jahre von John Brooke entwickelt, um möglichst einfach und schnell die Benutzerfreundlichkeit eines Systems zu bewerten. Das SUS Scoring vermeidet langwieriges Beantworten von Fragebögen, die mit dem Risiko behaftet sind, dass Testpersonen die Befragung aus Frustration oder Überforderungen nicht abschließen und so zu wenige Daten für die Bewertung der Benutzerfreundlichkeit zur Verfügung stehen [22]. Bei einer Metastudie über Usability-Studien, die zwischen den Jahren 1983 und 2008 durchgeführt wurden, kam in 17 von 40 Studien (42,5%) eine Nachbefragung mittels SUS zum Tragen [119]. Das Verfahren ist damit eine der am häufigsten verwendeten Methoden zur Usability-Befragung [76] [149].

Beim Scoring werden von den Versuchspersonen zehn standardisierte Fragen beantwortet [118], die im Wechsel positiv und negativ formuliert werden, so dass die beantwortende Person jede Frage lesen muss. Der Fragebogen sollte möglichst direkt nach Ende der Versuchsdurchführung von den Testpersonen durch Ankreuzen beantwortet werden, dabei sollte die Person eher schnell und instinktiv antworten [22].

Der Fragebogen zielt darauf ab, die Anwendbarkeit des Systems zu bewerten, es werden daher Fragen nach der Komplexität, Konsistenz der Bedienung, dem Lernaufwand und Akzeptanz gestellt. Die genauen Fragen sind im Anhang A.2 *SUS Fragebogen* zu finden.

Die Auswertung der Fragen findet mittels Skalierung statt, jede Frage ergibt 10 Punkte. Bei einer 5-Punkte Likert Skala werden die positiv formulierten Fragen (d.h. Fragen, die an ungerader Stelle in der Liste stehen) von 0 (Stimme überhaupt nicht zu) bis 4 (Stimme voll und ganz zu) kodiert und mit 2,5 multipliziert. Bei negativ formulierten Fragen (ungerade), findet die Kodierung umgekehrt statt. Letztlich können so Punktzahlen zwischen 0 und 100 Punkten für den gesamten Fragebogen erreicht werden, der SUS-Score.

Die Abbildung 2.1 zeigt, wie das Ergebnis typischerweise interpretiert werden

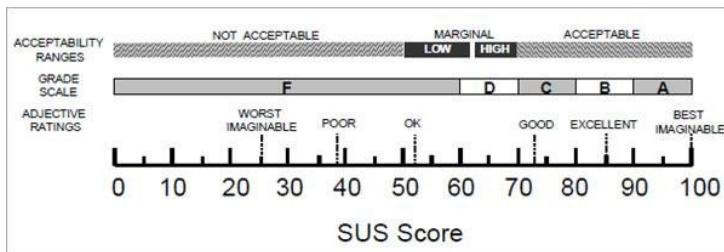


Abbildung 2.1: Übersetzung des SUS-Scores in gebräuchliche, adjektive Bewertungen und Schulnoten [11]

kann [11]. Werte von 0 - 60 weisen auf eine mangelhafte Usability hin, alles darüber ist akzeptabel. Werte ab 85 werden als exzellent betrachtet, wobei 100 die bestmögliche Punktzahl darstellt.

2.2.3 Messung der Belastung mittels Task Load Index

Der NASA Task Load Index (TLX) wurde bereits 1986 am NASA Ames Research Center in der Human Performance Arbeitsgruppe von Hart, Lowell et al. entwickelt [51]. Der TLX ist ein Beurteilungsinstrument, um eine *subjektive* Selbstauskunft von Personen zu erfragen, es stehen dabei Belastungen bzw. Anforderungen in verschiedenen Dimensionen im Vordergrund. Der TLX wird in unterschiedlichen Domänen eingesetzt; Untersuchungsgegenstand ist dabei immer eine Mensch-Maschine-Schnittstelle, bei über dieses Instrument die Belastung der Nutzenden *objektiv* erfasst werden soll.

Konkret werden, wie auch im Anhang A.3 *RTLX Fragebogen* dokumentiert, sechs verschiedene Kategorien von Anforderungen abgefragt:

- geistige Anforderung
- körperliche Anforderung
- zeitliche Anforderung
- Leistung
- Anstrengung
- Frustration

Jede Kategorie wird dabei mit einer Skala abgefragt, bei der die Testperson ein Kreuz setzen soll, je nachdem, ob sie die entsprechende Anforderung als *gering* oder *hoch* bzw. *gut* oder *schlecht* einschätzt. Zur Berechnung des Scores werden die Linien von links gezählt, bis zu eben jener, die die Person gesetzt hat, eins subtrahiert und das Ergebnis mit fünf multipliziert. Da es 21 Linien gibt (vgl. Anhang A.3), kann auf diese Weise je Frage ein Score von 0 - 100 erreicht werden [51].

In der ursprünglichen Fassung wird in einem ersten Schritt zuvor die Versuchsperson aufgefordert, die sechs Dimensionen paarweise zu priorisieren, nämlich danach, in der die Person die Relevanz der entsprechenden Anforderung für die

Erfüllung der Aufgabe höher einschätzt. Auf diese Weise entstehen 15 Vergleichspaare zwischen je zwei Anforderungen. Zur Errechnung der Gewichte wird nun die Gesamtzahl der Nennungen einer bestimmten Anforderung addiert und durch 15 geteilt, diese Gewichte werden dann mit dem jeweiligen Teil-Score multipliziert [114].

Häufig wird keine Gewichtung eingesetzt [50], dabei sind die Auswirkungen je nach Betrachtung unterschiedlich: Bei Hendy et al [52] war das Instrument auf diese Weise sensitiver, Liu & Wickens [81] konnten dagegen zeigen, dass die Sensitivität verringert wird und letztlich konnten Byers et al. [27] auch zeigen, dass sich die Sensitivität nicht verändert. Ein Verzicht auf die Gewichtung verringert aber in jedem Falle den Zeitbedarf für die Proband*innen, was in der Praxis bei Versuchsdurchführungen Vorteile mit sich bringt.

Arbeitslast	Wert
Niedrig	0 - 9
Mittel	10 - 29
Erhöht	30 - 49
Hoch	50 - 79
Sehr hoch	80 - 100

Tabelle 2.1: Die Einteilung der Arbeitslast gemäß TLX-Score, basierend auf [51]

In Tabelle 2.1 ist zu sehen, wie ein bestimmter Score zu interpretieren ist, diese Einteilung gilt sowohl für das Gesamtergebnis, als auch für die einzelnen Teilergebnisse der sechs Dimensionen.

2.3 Elektrotechnische Grundlagen und Messtechnik

Es folgen nun die weiteren elektrotechnischen Grundlagen, die für eine Signalstärkemessung benötigt werden. Hierzu zählen die Beschreibung einer elektromagnetischen Welle, als auch die Definition der Signalstärke und der darauf basierenden Received Signal Strength Indicator-Messung.

2.3.1 Elektromagnetische Wellen

Eine elektromagnetische Welle basiert auf der Kopplung von elektrischen und magnetischen Feldern und benötigt, anders als beispielsweise Schallwellen, kein Medium [35]. Die Entstehung von elektromagnetischen Wellen lässt sich unmittelbar aus den Maxwell-Gleichungen herleiten, diese wurden wie folgt von James Clerk Maxwell bereits im Jahr 1865 postuliert [87]:

$$\operatorname{div} \vec{E} = \frac{\rho}{\epsilon_0} \quad (2.1a)$$

$$\operatorname{div} \vec{B} = 0 \quad (2.1b)$$

$$\operatorname{rot} \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (2.1c)$$

$$\operatorname{rot} \vec{B} = \mu_0 \vec{j} + \mu_0 \epsilon \frac{\partial \vec{E}}{\partial t} \quad (2.1d)$$

Dabei bezeichnet \vec{E} die elektrische Feldstärke, \vec{B} die magnetische Flussdichte, ρ die Ladungsdichte, ϵ_0 die elektrische Feldkonstante, μ_0 die magnetische Feldkonstante, \vec{j} die elektrische Stromdichte und t die Zeit. *Div* und *rot* sind Differentialoperatoren: Die Divergenz (div) kann als Strömungsfeld verstanden werden, wobei Quellen eine positive Divergenz besitzen, Senken eine negative. Die Rotation (rot) beschreibt wiederum die Stärke und Drehachse eines Wirbels im Vektorfeld.

Gleichung 2.1a, auch als Gaußsches Gesetz bezeichnet, beschreibt die Ladung

als Quelle des elektrischen Feldes und Gleichung 2.1b, das Gaußsche Gesetz für Magnetfelder zeigt, dass das Feld der magnetischen Flussdichte quellenfrei ist. Das Induktionsgesetz, Gleichung 2.1c, beschreibt den Umstand, dass Änderungen in der magnetischen Flussdichte zu einem elektrischen Wirbelfeld führen. Das erweiterte Durchflutungsgesetz, Gleichung 2.1d, zeigt wiederum, dass elektrische Ströme zu einem magnetischen Wirbelfeld führen.

Daraus folgt, dass ein zeitvariantes Magnetfeld ein zeitvariantes elektrisches Feld induziert und umgekehrt. Dieser Vorgang benötigt hierzu kein Medium. Mittels einer Antenne und hochfrequentem Wechselstrom können solche elektromagnetischen Wellen in den freien Raum übertragen werden, wo sie sich selbstständig fortsetzen. Befindet sich eine weitere Antenne ausreichend nahe im Wechselfeld der Welle, induziert dieses wiederum Wechselströme in der empfangenden Antenne.

Informationen können in den elektromagnetischen Wellen beispielsweise durch Veränderung der Amplitude, Frequenz oder Phase kodiert werden, welche damit die Grundlage für eine drahtlose Kommunikation darstellen [136].

2.3.2 Signalstärke und RSSI

Bei der Ausbreitung von elektromagnetischen Wellen in einer realen Umgebung spielen weitere physikalische Effekte eine Rolle. Hierzu zählen Absorption, Transmission, Reflexion und Kombinationen [116], so dass es am Empfangspunkt einer solchen Welle zu Signalstörungen unterschiedlichster Art kommt [74]. Es existieren verschiedene Modelle, um die Empfangsstärke an einer bestimmten Stelle im Raum mathematisch zu berechnen. Die Modelle unterscheiden sich dabei hauptsächlich in den einfließenden Parametern, die je nach Umgebung oder gewünschter Genauigkeit der Vorhersage unterschiedlich stark gewichtet werden, bzw. überhaupt erst berücksichtigt werden.

Ein häufig eingesetztes Modell ist das *log-distance path loss model* [122], welches die Ausbreitung der Wellen innerhalb eines Gebäudes grundlegend abbilden kann. Anhand dieses Modells ist die Abhängigkeit zwischen Abstand und Signalstärke

gut erkennbar, es ist in Gleichung 2.2 wiedergegeben.

$$P_L = P_L(d_0) + 10\gamma \log_{10} \frac{d}{d_0} \quad (2.2)$$

P_L gibt dabei die zu erwartende Dämpfung in Dezibel (dB) an, γ ist der Pfadverlustexponent, der üblicherweise empirisch bestimmt oder über Referenzquellen angenähert werden kann [128] und d_0 die Referenzentfernung in Metern. Die aktuelle Entfernung, zur der die Dämpfung berechnet werden soll, fließt durch d in die Gleichung ein.

Unmittelbare Verwendung findet die Signalstärkemessung in Form des *Received Signal Strength Indicator* oder *Received Signal Strength Indication* (RSSI), ein Indikator für die enthaltene Leistung in einem empfangenen Funksignal. Der Wert ist abhängig vom jeweils empfangenden Gerät, allgemein steht jedoch ein höherer Wert für ein stärkeres Signal [120]. Mittels einem geräteabhängigen Skalierungsfaktor kann der Wert beispielsweise in Dezibel Milliwatt (dBm) angegeben werden und rangiert dann typischerweise im Bereich von -100 bis 0 dBm. Da der RSSI gemäß der Wellenausbreitung entfernungsabhängig ist, kann er für eine Lokalisierung in einem Sensornetzwerk herangezogen werden [99].

2.4 Sensoren und Sensornetzwerke

Sensoren ermöglichen die Aufnahme von Messwerten zur informationstechnischen Weiterverarbeitung. Sie stellen einen integralen Bestandteil für die Überwachung von realen Prozessen dar. Häufig werden hierfür nicht nur einzelne Sensoren eingesetzt, sondern ganze Netzwerke, um Messungen räumlich verteilt aufnehmen zu können. Beides wird im nachfolgenden Kapitel näher erläutert.

2.4.1 Sensorik

In diesem Abschnitt wird speziell der Aufbau eines Beschleunigungssensors beschrieben, da dieser häufig für eine Messung von Bewegungen herangezogen

wird. Auch im Rahmen dieser Arbeit findet er Anwendung, daher folgt nun eine detailliertere Beschreibung.

2.4.1.1 Beschleunigungssensoren zur Messung von Bewegung

Zur Bewegungserkennung von Objekten oder Personen werden häufig Beschleunigungssensoren, auch *Accelerometer* genannt, eingesetzt. Frühe Varianten nutzten zur Erfassung eine seismische Masse, um einen Schiebewiderstand zu verändern, welcher dann gemessen werden kann [8].

Moderne Varianten des Beschleunigungssensors verwenden ebenfalls seismische Massen, die einen dynamischen Druck auf einen piezoelektrischen Sensor ausüben, dessen elektrische Signale dann gemessen werden können [73].

Durch eine einfache Massenproduktion mittels Verfahren aus der Halbleitertechnik, sind mikroelektromechanisches Systeme (MEMS) [97] heute am weitesten verbreitet, vor allem bei integrierten Schaltungen mit mikroelektrischen Komponenten wie beispielsweise Mobiltelefonen. Diese Systeme können unter anderem eine seismische Masse nutzen, um eine integrierte elektrische Kapazität bei Beschleunigung zu verändern.

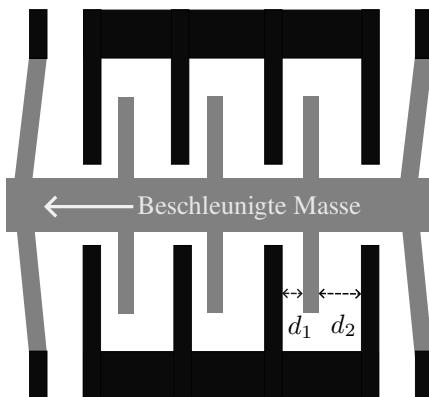


Abbildung 2.2: Schematischer Aufbau eines kapazitiven MEMS-Beschleunigungssensor basierend auf [97]

Schematisch lässt sich der innere Aufbau eines solchen Sensors wie in Abbildung 2.2 dargestellt beschreiben. Hier sind die Kondensatorflächen als graue und schwarze Kammstrukturen dargestellt: Wird der Sensor bewegt, verändern sich aufgrund der Trägheit der beschleunigten Masse die Abstände d_1 und d_2 der Kondensatorflächen, wodurch sich gemäß Formel 2.3 auch die Kapazität ändert. Ein einzelner Fächer dieses Aufbaus entspricht einem Plattenkondensator. Die Vielzahl solcher Plattenkondensatoren in einem MEMS-Beschleunigungssensor ist demnach analog zu einer Zusammenschaltung vieler Kondensatoren. Diese Abstandsänderung ist somit proportional zur Beschleunigung, die somit indirekt über die Kapazität als Messgröße zur weiteren Analyse beispielsweise in einem Mikrocontrollersystem mittels Analog-Digital-Wandlung zugänglich gemacht werden.

$$C = \varepsilon_0 \cdot \varepsilon_r \cdot \frac{A}{d} \quad (2.3)$$

In Gleichung 2.3 steht C für die Kapazität des Kondensators (in Farad, F), ε_0 für die elektrische Feldkonstante ($\varepsilon_0 \approx 8,85 \cdot 10^{-12}$ F/m), ε_r für die relative Permittivität des Dielektrikums zwischen den Platten (in Farad pro Meter, F/m), A für die Fläche der Platten (in Quadratmetern, m²) und d für den Abstand zwischen den Platten (in Metern, m).

Die Abstandsänderung selbst beruht auf der Kraft, die auf die Masse durch die Geschwindigkeitsänderung wirkt. Das zweite Newtonschen Gesetz [96], stellt die Beziehung von Impuls, Masse und Geschwindigkeit wie in Gleichung 2.4 dargelegt her, diese ist auch als *Grundgleichung der Mechanik* bekannt:

$$\vec{F} = m \cdot \vec{a} \quad (2.4)$$

Hierbei ist \vec{F} die resultierende Kraft auf einen Körper (in Newton, N), m die Masse des Körpers (in Kilogramm, kg), und \vec{a} die Beschleunigung des Körpers (in Metern pro Sekunde-Quadrat, m/s²).

Beschleunigungen werden im Kontext von Accelerometern häufig in Vielfachen

der mittleren Erdbeschleunigung g angegeben. Diese beträgt entsprechend Formel 2.5 ca. $9,81 \text{ m/s}^2$.

$$g \approx 9,81 \text{ m/s}^2 \quad (2.5)$$

2.4.2 Sensornetzwerke

Netzwerke aus Sensoren werden eingesetzt, um beispielsweise große Räume oder Flächen zu überwachen. Die Kommunikation zwischen den einzelnen Sensorknoten gestaltet sich je nach Einsatzzweck unterschiedlich. Die Spanne reicht von kabelgebundenen und kabellosen Netzen, über verschiedene Topologien und Anlageorte. Dieses Kapitel gibt hierzu einen Überblick.

2.4.2.1 Objektgebundene, tragbare & gerätefreie Sensornetzwerke

Ein wichtiges Kriterium bei der Einordnung von Sensornetzwerken ist der Installationsort. In der Literatur wird dazu häufig zwischen *objektgebundenen* (englisch: Object-tagged), *tragbaren* (englisch: Wearable) und sogenannten *gerätefreien* (englisch: Device-free) Sensorsystemen unterschieden [141].

Gerade in medizinischen Kontexten sind häufig am Körper getragene Sensoren vorzufinden, speziell auch im Zusammenhang der Aktivitätserkennung [103]. Diese können dann beispielsweise über Hautkontakt Vitalparameter messen. Am Körper getragene Sensoren haben höhere Anforderungen an die Hygiene und können auf die Nutzenden störend wirken [113]. Die Energieversorgung findet über Batterien oder Akkumulatoren statt.

Bei objektgebundenen Netzwerken geschieht die Energieversorgung auf demselben Weg. Allerdings haben diese Sensoren den Vorteil, dass sie als weniger störend empfunden werden, da sie nicht an- und ausgezogen werden müssen und außerdem kein unangenehmes Tragegefühl verursachen [42]. Nach aktuellem Stand häufig eingesetzt werden die sogenannten gerätefreien Systeme, hierunter zählen auch kamerabasierte Methoden. Diese Systeme bieten eine maximale Freiheit

bei der Interaktion, da sie weder das Objekt noch die Person einschränken. Eine Verwendung benötigt jedoch eine dauerhafte Installation der Hardware. Der Interaktionsraum ist dementsprechend auf den von den Kameras ausgeleuchteten Bereich beschränkt, durch die stationäre Installation stellt die Energieversorgung hingegen keine Einschränkung dar, da diese kabelgebunden erfolgen kann [54].

2.4.2.2 Netzwerktopologien

Die Topologie eines Netzwerks beschreibt die Anordnung der Geräte im Verbund. Dies trifft sowohl auf kabelgebundene als auch kabellose Netzwerke zu. Es ist zwischen physikalischer Anordnung, d.h. wie die Geräte durch Kabel oder Funkstrecken verbunden sind, und der logischen Anordnung, d.h. wie der Datenfluss zwischen den Geräten erfolgt, zu unterscheiden. Eine Übersicht üblicher Verbindungsarten ist in Abbildung 2.3 dargestellt [26].

In der Sterntopologie (Abb. 2.3 a) werden die Daten an einen zentralen Punkt geleitet, um dort beispielsweise weiterverarbeitet zu werden, alle Knoten haben den selben Adressaten. Im Gegensatz dazu hat bei der Ringtopologie (Abb. 2.3 b) jeder Knoten genau einen (Datenfluss nur in eine Richtung) oder zwei Empfänger (Datenfluss in beide Richtungen), die aber für jeden Knoten unterschiedlich sind und nicht doppelt existieren, so dass insgesamt ein geschlossener Ring entsteht. Diese Variante ist in kabelgebunden Netzen anzutreffen, selten jedoch in Funknetzwerken. Die Baumtopologie (Abb. 2.3 c) ähnelt der Sterntopologie, besitzt aber eine streng hierarchische Ordnung: Daten werden von den Sensoren in Knoten gebündelt, die ihrerseits die gebündelten Informationen an zentrale Stellen weitergeben. In der vermaschten Topologie (Abb. 2.3 d) kann jeder Knoten mit jedem anderen Knoten direkt kommunizieren. Dies kann die Laufzeit einzelner Datenpakete verringern und die Robustheit erhöhen, führt aber oftmals zu höherem Energieverbrauch [26].

Gerade im Hinblick auf die Energieeffizienz bietet jede Topologie Vor- und Nachteile, die aber auch in Abhängigkeit der Umgebung zu betrachten sind [104]. In der Praxis entstehen so häufig Mischformen, bei denen die grundlegende Struktur beispielweise einer Baumtopologie entspricht, in einzelnen Etagen eines Gebäudes

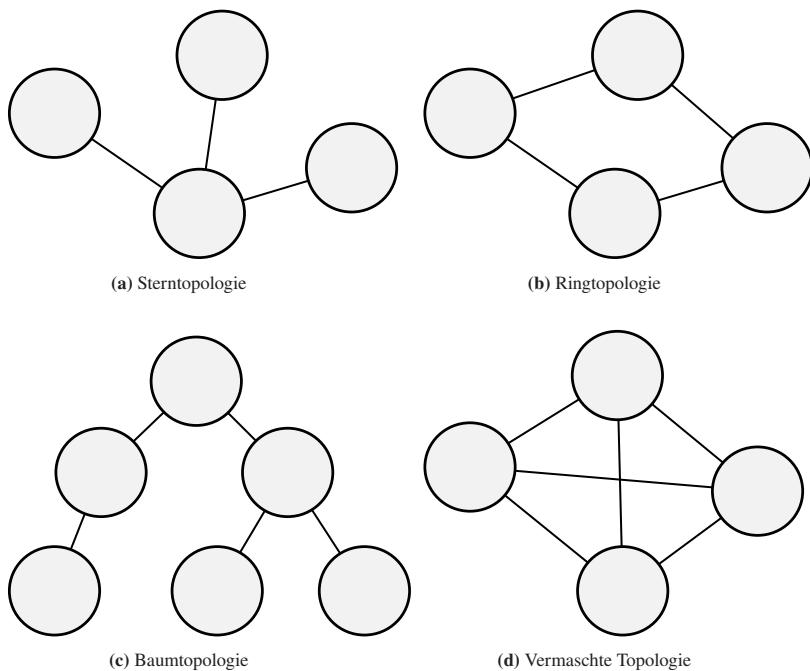


Abbildung 2.3: Netzwerk topologien in Sensor netzwerken nach [70]

dann aber ein vermaschtes Netz eingesetzt wird, um die Robustheit der Übertragung zu garantieren.

2.4.2.3 Lokalisierungsmethoden

Eine häufige Anwendung von Sensor netzwerken ist das Lokalisieren und Nachverfolgen von Objekten oder Personen. Hierzu bedarf es einer technischen Möglichkeit, um die einzelnen Sensoren zu orten. Nach Cheng et al. ergibt sich die in Abbildung 2.4 dargestellte Taxonomie [31].

Zunächst kann hierbei zwischen *Fremd-* und *Eigenlokalisierung* unterschieden werden. Bei der Fremdlokalisierung wird das Sensor netzwerk verwendet, um

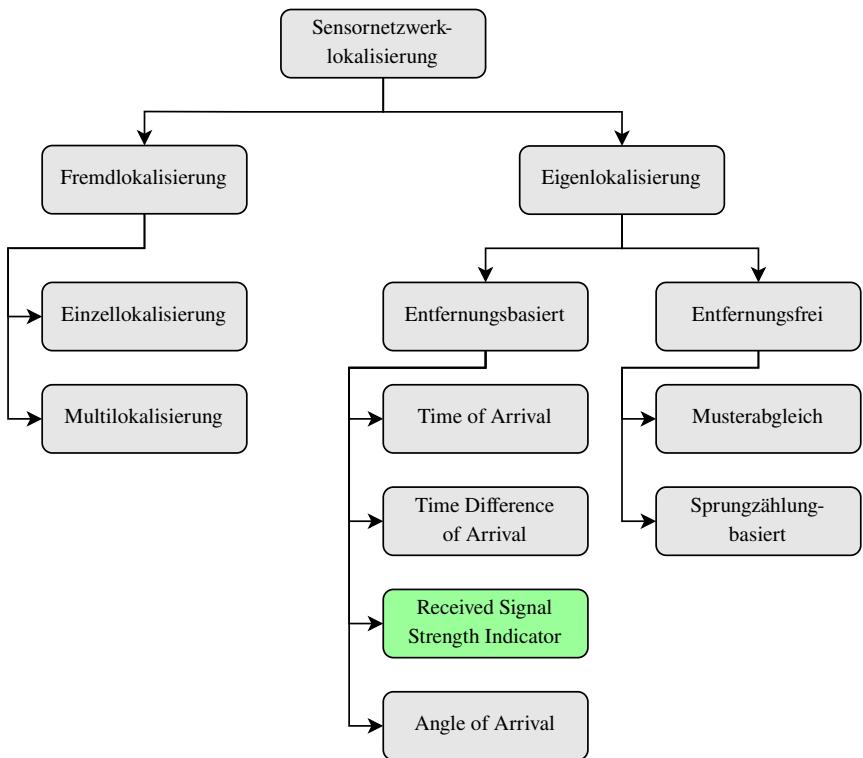


Abbildung 2.4: Lokalisierungsmethoden in Sensornetzwerken nach [31]

beispielsweise über die Messung verschiedener Konzentrationen eines Gemisches im Raum durch die Sensorknoten auf die Quelle eines Lecks zu schließen. Der Gegenstand der Lokalisierung ist also nicht Teil des Sensornetzwerks selbst. Im Gegensatz dazu ermittelt bei der Eigenlokalisierung jeder Sensorknoten seine eigene Position in Relation zu den anderen Knoten. Hierzu kann abermals zwischen *entfernungs frei* und *entfernungs basiert* unterschieden werden. Entfernungs frei meint in diesem Fall, dass keine Distanzen für die Lokalisierung gemessen werden, sondern eine Lokalisierung indirekt erfolgt. Beispielsweise indem beim *sprungzählungsbasierten Verfahren* die Anzahl der Sprünge eines Datenpakets von einem Knoten zu einem anderen mitgezählt werden, um so eine Maß für die

Entfernung zu erhalten. Beim *Musterabgleich* wird hingegen in einem Raum zum Beispiel die Signalstärke Punkt für Punkt vermessen (das sogenannten *Fingerprinting*), ein anschließend eingebrachter Sensorknoten kann sich dann mittels dieser Signalstärkekartierung lokalisieren. Ein sehr großes Feld ist die *entfernungs basierte Lokalisierung*, bei der Methoden wie *Time of Arrival* oder *Time Difference of Arrival* angewendet werden, um über die Signallaufzeit, bei bekannter Ausbreitungsgeschwindigkeit, die Entfernung zu bestimmen. *Angle of Arrival* basiert auf geometrischen Grundlagen, um über die unterschiedliche Ankunftszeit eines Signals in einem Antennenarray den Winkel zum Ursprung zu bestimmen [31]. Ein kostengünstiges Verfahren sind *Received Signal Strength Indicator* (RSSI)-Methoden, bei denen Aufgrund einer Abschwächung des Signals durch die Distanz zum Sender auf die Entfernung geschlossen wird [58]. Eine genauere Beschreibung dieser Methodik findet sich in Kapitel 2.3.2.

2.5 Künstliche Intelligenz

Der Begriff *künstliche Intelligenz* wurde bereits 1955 von John McCarthy geprägt, in den 2010er Jahren hat sich die Entwicklung stark beschleunigt [88]. *Künstliche Intelligenz* bezeichnet übergreifend Verfahren, mit denen Computer menschliche Intelligenz nachahmen können. Es ist möglich, dass diese Verfahren hierzu auch vorprogrammiert sind, in jedem Falle werden die Aufgaben regelbasiert gelöst. Eine Unterkategorie bildet das *maschinelle Lernen*, bei dem Computer durch Erfahrungslernen besser beim Lösen von Aufgaben werden und die Regeln zur Lösung selbst lernen. Eine Teilmenge innerhalb des maschinellen Lernens bildet wiederum *Deep Learning*, welches auf künstlichen neuronalen Netzen beruht und die biologische Struktur eines Gehirns in Teilen imitiert. *Deep* bezieht sich hierbei auf die *Tiefe* des neuronalen Netzes, d.h. auf die höhere Anzahl der verborgenen Schichten im Netz [28].

Die Anzahl der verfügbaren Modelle, Architekturen und Algorithmen ist sehr umfangreich. Im Zuge dessen werden im Folgenden nur diejenigen Architekturen

und Algorithmen näher beschrieben, die auch im weiteren Verlauf der Arbeit Anwendung finden.

2.5.1 Feature Engineering

Feature Engineering beschreibt einen Vorverarbeitungsschritt, welcher vor dem Training einer KI die relevanten Merkmale (engl. *feature*) eines Signals durch deterministische Methoden auswählt, um bei gleichzeitigem Erhalt der Vorhersagegenauigkeit die benötigte Datenmenge für Training und Inferenz zu minimieren [146]. Diese Technik setzt ein hohes Verständnis für die vorliegenden Daten voraus, um die optimalen Features auszuwählen. In klassischen Verfahren des maschinellen Lernens (vgl. Kapitel 2.5.3) ist das Feature Engineering häufig Teil der Vorverarbeitungskette und findet üblicherweise nach der Rohdatenaufbereitung, d.h. der Bereinigung der Daten um unerwünschte Artefakte, statt.

Eine häufig eingesetzte Methoden beim Feature Engineering stützt sich beispielsweise auf die Pearson Korrelation [80], um zu untersuchen, welches Feature eine größere Relevanz für die Klassifikation besitzt. Der Korrelationskoeffizient kann Werte von -1 bis +1 annehmen, sind zwei Features sehr stark korreliert, kann eines der beiden entfernt werden, da der Informationsgehalt dann redundant ist. Auf diese Weise kann die Dimension des Datensatzes bei Beibehaltung des Informationsgehalts reduziert werden, was einen Geschwindigkeitsgewinn bei Training und Inferenz zur Folge hat, ohne die Vorhersagegenauigkeit zu verringern.

Eine weitere häufig eingesetzte Methode ist die *Recursive Feature Elimination with Cross Validation* (RFECV) zur Auswahl geeigneter Features [91]. Hierbei werden automatisiert Modelle mit einem Subset der möglichen Features trainiert und im Anschluss miteinander verglichen, bis eine bestimmte, vorgegebene Anzahl an Features bei bestmöglicher Performanz erreicht ist. Dies stellt ein nützliches Vorgehen dar, vor allem bei stark in ihrer Rechenleistung oder Speichergröße beschränkten Systemen; hiermit kann das Modell beispielsweise auf eine bestimmte maximale Speichergröße optimiert werden. Um dabei eine Überanpassung (engl. *Overfitting*) zu vermeiden wird eine Kreuzvalidierung (engl. *Cross Validation*) eingesetzt, so dass sich Trainings- und Testdatensets auch bei

beschränkter Datensatzgröße stets unterscheiden und damit eine Generalisierbarkeit der Modelle gegeben ist [92].

2.5.2 Performanzmetriken

Performanzmetriken werden im Bereich der KI-Entwicklung herangezogen, um zum Beispiel die Leistung eines Algorithmus im Bezug auf die Korrektheit der Inferenz zu beurteilen. Es folgt daher eine Reihe von häufig eingesetzten Metriken, die auch in dieser Arbeit Anwendung finden.

Konfusionsmatrix

Eine Konfusionsmatrix wird in der Statistik und im maschinellen Lernen verwendet, um die Leistung eines Klassifikationsmodells zu bewerten. Hierzu wird üblicherweise in tabellarischer Form dargestellt (vgl. Tabelle 2.2), wie gut ein Modell bestimmte Klassen vorhersagen kann. Sie zeigt darüber hinaus die Art der Fehler an, die das Modell gemacht hat. Die Matrix wird wie in der Tabelle dargestellt für binäre Klassifikationsprobleme genutzt, kann aber auch für mehr als zwei Klassen erweitert werden [93].

	Vorhergesagt Positiv	Vorhergesagt Negativ
Tatsächlich Positiv	TP (Richtig Positive)	FN (Falsch Negative)
Tatsächlich Negativ	FP (Falsch Positive)	TN (Richtig Negative)

Tabelle 2.2: Darstellung einer Konfusionsmatrix

Hierbei sind *Richtig Positive* (engl. True Positives, TP) diejenigen Fälle, die tatsächlich positiv sind und auch vom Modell als positiv vorhergesagt wurden. Hingegen sind *Falsch Positive* (engl. False Positives, FP) die Fälle, die im Gegensatz zwar tatsächlich negativ sind, aber vom Modell fälschlicherweise als positiv vorhergesagt wurden. Weitere Kategorien sind *Richtig Negative* (True Negatives, TN), d.h. vom Modell als negativ vorhergesagte Fälle, die auch tatsächlich negativ

sind und abschließend die Gruppe *Falsch Negative* (False Negatives, FN), die solche Fälle enthält, die tatsächlich positiv sind, aber vom Modell fälschlicherweise als negativ vorhergesagt wurden [142]. Aus diesen Kategorien lassen sich weitere Metriken ableiten, darunter die nun nachfolgend näher erläuterten.

Genauigkeit

Die Genauigkeit (engl. Accuracy) berechnet sich entsprechend Formel 2.6 aus den korrekt vorgesagten Fällen der wahren Positiven (TP) und wahren Negativen (TN), bezogen auf die Gesamtheit aller Vorhersagen (TP + TN + FP + FN) [43].

$$\text{Genauigkeit} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.6)$$

Die Genauigkeit ist ein weit verbreitete Metrik um die Qualität eines Modells zu bestimmen. Bei unbalancierten Datensätzen, d.h. Sets, bei denen Klassen unterschiedlich häufig auftreten, kann diese Metrik jedoch ihre Aussagekraft verlieren, hierfür können dann andere Performanzmetriken, wie der F_1 -Score herangezogen werden.

Präzision

Wie in Gleichung 2.7 dargestellt, beschreibt die Präzision (engl. Precision), den Anteil der tatsächlichen wahren Fälle (TP), bezogen auf alle vom Modell als wahr vorhergesagten Fälle (TP + FP) [108].

$$P = \frac{TP}{TP + FP} \quad (2.7)$$

Sensitivität

Die Sensitivität, im Englischen häufig auch *Recall* genannt, bezieht im Gegensatz zur Präzision die richtig Positiven (TP) auf die Gesamtheit der tatsächlich positiven Fälle (TP + FN), wie in Gleichung 2.8 zu entnehmen ist[112].

$$R = \frac{TP}{TP + FN} \quad (2.8)$$

F_1 -Score

Der F_1 -Score eignet sich wie bereits erwähnt besonders für unbalancierte Datensets. Er bildet aus Precision und Recall das harmonische Mittel wie in Gleichung 2.9 zu sehen [83].

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \quad (2.9)$$

Ein F_1 -Score von 1 ist nur dann erreichbar, wenn sowohl Präzision als auch Sensitivität ihren Maximalwert erreichen - somit ist der F_1 -Score auch besonders dann wertvoll, wenn beide Metriken wichtig sind, wie beispielsweise bei einem medizinischen Test, bei dem sowohl möglichst alle kranken Personen gefunden werden sollen, gleichzeitig aber auch möglichst wenige falsche Diagnosen gestellt werden sollen. Durch die Verrechnung von Precision und Recall leidet im Gegenzug jedoch auch die Transparenz, da bei einem schlechten Ergebnis nicht ersichtlich ist, in welchem Maße die beiden Metriken zum Ergebnis beigetragen haben.

2.5.3 Klassische Methoden des maschinellen Lernens

Decision Trees

Entscheidungsbäume (engl. *Decision Trees*, DT) bilden einen Lernansatz für ML-Algorithmen, bei dem ein hierarchisches Modell anhand sortierter Merkmalswerte gebildet wird [68]. Damit können, wie mit anderen ML-Methoden ebenfalls, zwei Arten von Problemen gelöst werden, nämlich Klassifikations- und Regressionsprobleme [133]. Beide unterscheiden sich in der Art ihrer Ausgabe: Während Klassifikatoren diskrete Ausgangswerte ausgeben, um Eingangswerte einer Kategorie zuzuordnen, beispielsweise ob eine Email *kein Spam* oder *Spam* ist, sind die Ausgangswerte von Regressionsalgorithmen kontinuierlich. Dies kann unter anderem genutzt werden, um von den Eingangswerten auf eine Werteverteilung zu schließen, wie beispielsweise die erwarteten *Grundstückspreise* in Abhängigkeit von Einwohnerdaten [67].

Um einen Entscheidungsbaum auszuwerten (vgl. Abbildung 2.5) wird angefangen beim Wurzelknoten W der komplette Baum bis hin zu einem Blatt b durchlaufen. An jeder Verzweigung, einem *Knoten* t , wird eine Entscheidung zu einem bestimmten Merkmal getroffen und basierend auf dieser Entscheidung der weitere Baum durchlaufen. Am Schluss gibt das Blatt b dann die Klasse bzw. die Regression an [44].

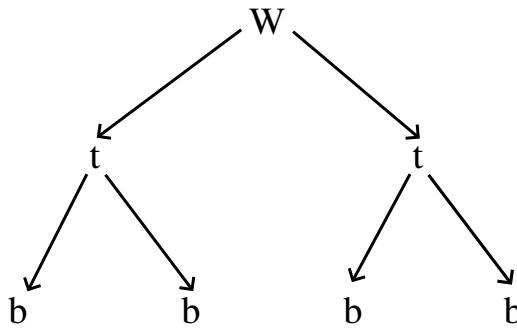


Abbildung 2.5: Schematische Darstellung eines Entscheidungsbaumes nach [44]

Die Generierung des Entscheidungsbaum geschieht durch die sogenannte Induktion [6], welche typischerweise durch ein Partitionierungsverfahren nach dem Teile-und-Herrsche-Prinzip durchgeführt wird [94]. Bekannt für ihre performante Partitionierung sind *CART* [21], *C4.5* [110] und *ID3* [109].

DTs sind anfällig für eine Überanpassung, aber dafür performant in ihrer Inferenz. Ihre innere Struktur ist gut nachvollziehbar, solange die Bäume nicht zu groß werden [92].

Random Forest

Ein Random Forest (RF) Modell besteht in seinem inneren Aufbau aus einer Reihe von einzelnen Entscheidungsbäumen und kann somit ebenfalls für Regression und Klassifikation eingesetzt werden [92]. Die DTs sind voneinander unabhängig und erzeugen individuelle Vorhersagen [133], die dann beispielsweise mittels Mehrheitsvotum über die endgültige Klassifikation entscheidet, das sogenannte

Aggregating. Das RF-Modell ist daher eine *Ensemble*-Methode, da hier Basismodelle, wie DTs, über ein Entscheidungs- und Gewichtungverfahren verknüpft werden, um so beispielsweise Overfitting zu vermeiden und ein besseres Ergebnis auf unbekannten Daten zu erhalten [68].

Zur Erzeugung des RF-Modells wird üblicherweise das *Bagging*-Verfahren angewandt, welches aus den beiden Teilschritten *Bootstrapping* und *Aggregating* (siehe oben) besteht [20]. Beim Bootstrapping werden mittels Ziehen und Zurücklegen von einzelnen Datenpunkten aus einem Datensatz neue Datensätze erzeugt, nämlich so viele, wie Entscheidungsbäume generiert werden. Jeder DT wird dann auf Basis eines generierten Datensatzes induziert. Die Datensätze können dabei Daten aus dem Originaldatensatz mehrfach oder gar nicht enthalten. Auf diese Weise entstehen zwischen den DTs leichte Abweichungen und eine Überanpassung wird vermieden.

RF-Modelle sind ebenfalls sehr performant, die Nachvollziehbarkeit der Entscheidung kann allerdings schneller leiden, da bei der Erzeugung eines Waldes an Entscheidungsbäumen die Interpretation schwieriger wird. Durch das Bagging-Verfahren ist dieses Modell jedoch deutlich robuster gegenüber Überanpassungen [68].

Support Vector Machine

Der Support Vector Machine (SVM) Algorithmus wurde bereits 1964 durch Vapnik und Chervonenkis erfunden [32] und hat bis heute zahlreiche Erweiterungen erfahren, beispielsweise 1992 durch Boser et. al für nicht-lineare Probleme [18]. Eine SVM kann sowohl zur Klassifizierung als auch zur Regressionsanalyse verwendet werden und finden dabei, im Gegensatz zu anderen Machine Learning Methoden, stets das globale Optimum [86]. Ziel ist hierfür die Separierung von Datenpunkten mittels einer Trenneraden, der sogenannten Hyperebene.

In Abbildung 2.6 ist die Hyperebene als rote Gerade eingezeichnet, die Datenpunkte, in der Abbildung als Fünfecke bzw. Sterne repräsentiert, werden innerhalb des Modells als Vektoren einbezogen. Zur Generierung der Hyperebene ist es notwendig, eine ausreichende Menge an Datenpunkten zur Verfügung zu haben, von denen bekannt ist, zu welcher Klasse sie gehören. Der Abstand derjenigen Datenpunkte,

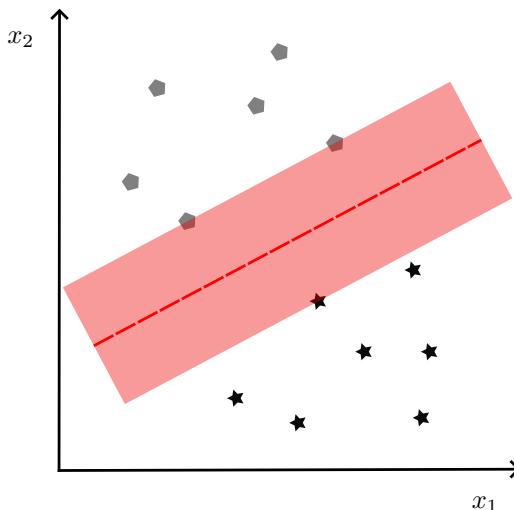


Abbildung 2.6: Schematische Darstellung einer SVM-Klassifizierung

die am nächsten zur Hyperebene liegen, wird zudem maximiert, damit die sogenannte *Margin*, in der Abbildung als rote Fläche dargestellt, möglichst groß wird [30]. Die namensgebenden Stützvektoren sind genau diese am nächsten liegenden Datenpunkte, da weiter entfernte Punkte keine Relevanz für die Konstruktion der Hyperebene besitzen. Ein neuer, unbekannter Datenpunkt kann somit in eine der Klassen zugeordnet werden.

Nicht-lineare Probleme werden mittels dem sogenannten *Kernel-Trick* klassifiziert, obwohl die Hyperebene zunächst nur als Gerade konstruiert werden kann. Die Trainingsvektoren werden hierzu in einen höherdimensionalen Raum überführt, wodurch die Vektormenge wieder linear trennbar wird. Durch die Rücktransformation in den ursprünglichen Datenraum wird so aus der linearen Hyperebene eine nichtlineare Hyperfläche, die auch stark überlappende Klassen trennen kann. Durch die Nutzung geeigneter Kernelfunktionen ist es möglich, die rechenintensiven Transformationen operativ nicht durchführen zu müssen [86].

2.5.4 Künstliche neuronale Netze

Mehrlagiges Perzeptron

Das mehrlagige Perzeptron (engl. *Multilayer Perceptron*, MLP), ist die ursprüngliche Form der mehrlagigen neuronalen Netze, die sich später weiterentwickelt und differenziert haben [95]. Das MLP besteht aus einer Eingangsschicht, ein oder mehreren verborgenen Schichten und einer Ausgangsschicht (vgl. Abbildung 2.7) [139]. In diesem Zusammenhang geläufig sind heutzutage sogenannten *Deep Neural Networks* (DNN), zu deutsch *tiefe neuronale Netze*. Je nach Definition genügen bereits zwei, zumindest aber mehrere verborgene Schichten, um als solches zu gelten. Durch eine Vielzahl an Schichten steigt die Fähigkeit des Netzes komplexe Zusammenhänge, beispielsweise in Bildern, zu erlernen. Im Gegensatz dazu steht aber die immer größer werdende Menge an benötigten Trainingsdaten und der wachsende Bedarf an Rechenleistung und damit auch der steigende Energiebedarf. Die verborgenen Schichten transformieren letztlich die Eingangswerte auf nicht-lineare Weise, so dass die Daten für die Ausgangsschicht linear trennbar werden [72].

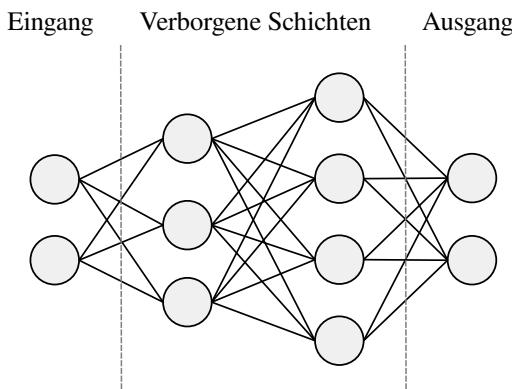


Abbildung 2.7: Schematische Darstellung eines MLP nach [139]

Jede Schicht enthält eine Reihe von sogenannten Neuronen, wobei alle mit allen weiteren Neuronen der vorherigen und nächsten Schicht verbunden sind. Dies

unterscheidet diese Architektur von anderen, ähnlichen Architekturen, wie beispielsweise Faltungsnetzen (engl. *Convolutional Neural Network*, CNN). Der Wert eines einzelnen Neurons setzt sich aus einem Gewicht und einem Bias zusammen, welche durch den Trainingsprozess ermittelt werden. Ein Neuron wird aktiviert, wenn der Ausgang entsprechend einer sogenannten Aktivierungsfunktion einen bestimmten Schwellenwert überschreitet. Diese Aktivierungsfunktionen sind zudem nicht-linear, damit das Netzwerk auch nicht-lineare Beziehungen modellieren kann, andernfalls könnten durch die linearen Operationen innerhalb der Netze (typischerweise Matrixmultiplikationen und Vektoradditionen) beliebig viele Schichten zu einer einzigen Schicht transformiert werden.

Die Gewichtung verändert den Beitrag eines einzelnen Neurons bzw. erhöht die Wahrscheinlichkeit, dass es aktiviert wird. Der Bias erlaubt auf globaler Ebene die Verschiebung des Ausgangs und ist typischerweise als Neuron implementiert, welches stets den Wert 1 hat, da es keine eigenen Eingänge besitzt. Sein Ausgang wird aber selbst gewichtet, so dass insgesamt der Ausgangswert des Neurons verschoben werden kann, um komplexere Funktionen abzubilden [13].

MLPs wurden erfolgreich für Spracherkennung, Bildkomprimierung und viele weitere Anwendungsfelder eingesetzt [124]

Long Short-Term Memory

Long Short-Term Memory (LSTM) Architekturen gehören zur Klasse der rekurrenten neuronalen Netze (RNN) und wurden 1997 von Hochreiter et al. vorgeschlagen [53].

Sie eignen sich besonders zur Erkennung von Mustern in Datensequenzen und Prozessen, beispielsweise in Zeitreihendaten von Sensordaten, da es mittels LSTMs möglich ist, zeitliche Zusammenhänge zu lernen, auch wenn die Zeitspanne eines Prozesses in verschiedenen Samples des selben Datensatzes unterschiedlich lang ist. Bei RNNs wird der Eingang nicht nur in Richtung des Ausgangs propageert, es ist zusätzlich auch möglich, dass Neuronen zu sich selbst, oder einer früheren Schicht rückgekoppelt sind [47].

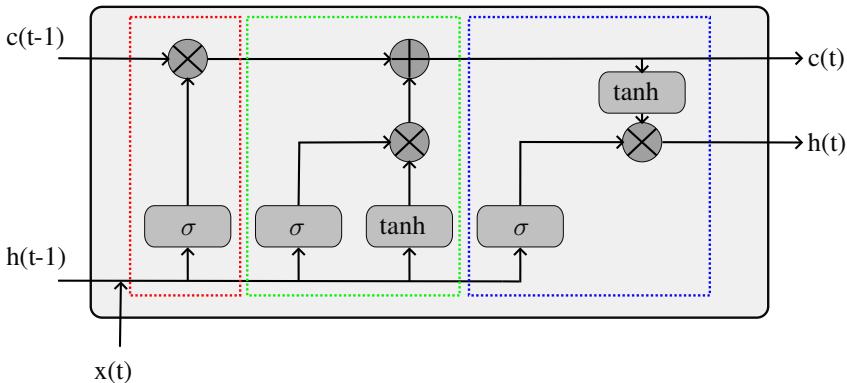


Abbildung 2.8: Schematische Darstellung einer LSTM-Zelle nach [10]

Für LSTMs charakteristisch ist das Vorhandensein von drei Toren (engl. *Gates*), diese sind in den jeweiligen Farben in Abbildung 2.8 dargestellt: Dem sogenannten *Input Gate* (Grün), welches entscheidet, wie wichtig der aktuelle Input ($x(t)$) ist, das *Output Gate* (Blau), welches den nächsten *Hidden State* ($h(t)$) bestimmt und schließlich das *Forget Gate* (Rot), wo entschieden wird, was aus dem vorangegangenen Schritt relevant ist [10]. Der Hidden State ist eine Repräsentation des vergangenen Inputs ($h(t - 1)$), über diesen Mechanismus wird also Information von einem Zeitschritt zum nächsten persistiert. Der Zellenzustand (engl. *Cell State*) $c(t)$ als Ganzes bildet wiederum das Langzeitgedächtnis des Netzes ab. Die grafischen X und + Symbole stellen punktweise Multiplikationen bzw. Additionen dar, σ und \tanh die jeweilige Aktivierungsfunktion.

3 Einordnung in den Stand der Wissenschaft und Technik

Diese Arbeit hat zum Ziel, ein System zu entwickeln, welches es erlaubt, Prozesse zu überwachen, insbesondere im Kontext des Kochens zuhause. Das Gesamtsystem gliedert sich dabei in die Teilbereiche der Datenaufnahme mittels Sensorik und in die algorithmische Auswertung der aufgenommenen Messdaten. Für Letzteres werden innerhalb dieser Arbeit verschiedene Methoden des maschinellen Lernens eingesetzt. Hier sei angemerkt, dass eine grundlegende Entwicklung von neurartigen KI-Architekturen nicht Teil dieser Arbeit ist, sondern viel mehr eine Auswahl und Anwendung derer.

Entsprechend des dargelegten Fokus gliedert sich der nachfolgend vorgestellte Stand der Wissenschaft und Technik in verschiedene Unterkapitel. Zunächst werden allgemein Sensorsysteme im Hinblick auf Aktivitätserkennung in Wissenschaft und Technik erörtert, welche in diesem Zuge auch angrenzende Gebiete, wie beispielsweise kamerabasierte Systeme, einschließt. Der Fokus liegt dabei auf Systemen, die in den Domänen des Sports, Pflege, Geriatrie und der Industrie im Allgemeinen eingesetzt werden.

Im Anschluss fokussiert sich die Betrachtung auf Methoden der künstlichen Intelligenz, die zur Handlungserkennung beim Menschen eingesetzt werden. Hier sind ebenfalls auch kamerabasierte Systeme in die Darstellung eingeschlossen.

Das Kapitel schließt mit einem Fazit und der Abgrenzung dieser Arbeit zum aktuellen Stand der Wissenschaft und Technik.

3.1 Sensorsysteme zur Handlungserkennung

Einleitend folgt nun ein Überblick über einzelne Sensorsysteme zur Handlungserkennung, die teilweise bereits erwerbar sind, oder nach aktuellem Stand in der Wissenschaft erforscht werden. Dieser Überblick beinhaltet die Domänen des Sports, der pflegerisch-geriatrischen- und Industriesensorik. In Sport und Industrie werden Sensoren bereits sehr häufig zur Handlungserkennung eingesetzt, für pflegerische Anwendungsfälle sind nur wenige Sensorsysteme im Einsatz.

3.1.1 Sensoren und Systeme

Die wahrscheinlich populärste Domäne für den Einsatz von Systemen zur automatisierten Handlungserkennung ist der Sport. Dieses Einsatzfeld deckt ein breites Spektrum an Anwendungen ab, begonnen beim Freizeitsport, über professionelle Einsatzzwecke bis hin zu physiotherapeutischen Maßnahmen. Forschungsfelder sind hierbei beispielsweise die Optimierung von Bewegungsabläufen, wie unter anderem von Peng et al. im Rahmen von Volleyballübungen gezeigt [105]. Grundlage ist hierbei eine inertiale Messeinheit (IMU), die die Bewegungen am Handgelenk misst und auf einer Speicherplatte ablegt. Andere Systeme, wie der *Blue Trident* von Vicon, basieren ebenfalls auf IMUs und können je nach Sportart flexibel angebracht werden.

In Abbildung 3.1 ist der Sensor zwischen den Schulterblättern angebracht, um die Bewegungen einer schwimmenden Person zu erfassen und automatisiert auszuwerten, um so den Bewegungsablauf zu optimieren und die Geschwindigkeit der Person für Wettkämpfe zu erhöhen.

Ein großer Markt stellen Fitnesstracker dar, die entweder in dedizierten Geräten oder als Teil einer Smartwatch angeboten werden. Namhafte Hersteller wie *Fitbit*, *Apple* oder *Garmin* bieten verschiedene Lösungen an, um damit sowohl Vitaldaten, aber auch Handlungen zu messen. Diese Aktivitäten sind hierbei üblicherweise sportbezogen und werden über die integrierte Beschleunigungssensorik erfasst. Fitnesstracker können so automatisch erkennen, ob eine Person in



Abbildung 3.1: Vicon Blue Trident zur Schwimmbewegungsoptimierung [82]

Bewegung ist oder nicht. Darauf hinaus kann typischerweise auch die Art der Bewegung unterschieden werden, so dass der Tracker automatisiert erkennt, ob beispielsweise ein Radfahrtraining oder Lauftraining begonnen wurde [48].

Ähnlich den üblichen Fitnesstrackern, ist das *Tempo Armband* der Firma CarePredict (vgl. Abbildung 3.2) ein am Handgelenk tragbares Gerät, welches zur Aktivitätserkennung bei Menschen eingesetzt werden kann.



Abbildung 3.2: Das Tempo Armband der Firma CarePredict [29]

Die Zielgruppe des Systems sind Senior*innen und ihre Zugehörigen, die mittels des Armbands stets den aktuellen Zustand der Person erfahren können. Im Armband werden verschiedene Sensoren, unter anderem zur Vitalwertbestimmung, eingesetzt, die Aktivitätserkennung wird aber auch hier über Beschleunigungssensoren umgesetzt. Das System soll dafür automatisiert Aktivitäten wie Essen, Schlafen, Baden und Kochen unterscheiden können. Bei Abweichungen vom Normalverhalten wird selbstständig Feedback an die zugehörigen Personen mittels App gegeben.

Ähnlich funktioniert *Moio Care* des Unternehmens Ahead Care. Über einen am Rücken getragenen Beschleunigungssensor können hier ebenfalls verschiedene Zustände von pflegebedürftigen Personen erfasst werden, wie beispielsweise Umlagerungen oder Stürze. Der *SensFloor* von Future-Shape verlagert die Zustandserfassung pflegebedürftiger Personen in eine drucksensitive Matte, worüber Aktivitäten aufgezeichnet und analysiert werden können, so dass auch hier bei Abweichungen vom Normalzustand eine Alarmierung ausgegeben werden kann.

Weitere Systeme werden im Umfeld von Chemieanlagen eingesetzt, wie beispielsweise jenes von Sun et al. [132]. Dieses System basiert auf unterschiedlichen Sensoren, die über eine gemeinsame Schnittstelle drahtlos Daten an eine zentrale Steuereinheit schicken. Zu den Sensoren zählen Feuer- und Gassensoren, aber auch solche für abnormale Geräusche. Eine beobachtende Person kann mittels Sensordatenfusion bei Überschreiten automatisiert ausgewerteter Grenzwerte alarmiert werden und gegebenenfalls einschreiten. Auch eine automatische Steuerung von Sicherheitsgeräten ist im Systemkonzept vorgesehen.

Kommerziell verfügbar ist *eKanban* von Steute. Das Sensornetzwerk kann um eine Reihe von heterogenen Sensoren, wie beispielsweise dem in Abbildung 3.3 dargestellten Universaltransmitter, erweitert werden, um verschiedene Aufgaben in einer Industrieanlage zu überwachen.



Abbildung 3.3: Universaltransmitter von Steute [129]

Unter anderem ist es damit möglich den Materialfluss in einer Fabrik auszuwerten, um so Einblicke in den laufenden Prozess zu erhalten. Hierzu zählen auch automatisierte Informationen an die Materialverwaltung, um Nachschub für Materialien zu bestellen.

3.1.2 Sensorsysteme mit direktem Nutzendenfeedback

In der Wissenschaft beschäftigen sich verschiedene Arbeitsgruppen mit Sensorsystemen mit direktem Feedback in der Domäne der Pflege und Geriatrie. So stellten Bouchard et al. 2020 ein intelligentes System vor, das Menschen mit leichten kognitiven Einschränkungen beim Kochen unterstützen soll [19]. Das System besteht aus Temperatursensoren, Infrarotsensoren, elektromagnetischen Sensoren und Kraftmessern, um den Zustand der Person zu erfassen. Diese Sensoren sind um den Herd angebracht und können mittels künstlicher Intelligenz den Nutzenden durch das Rezept führen und gefährliche Situationen erkennen. Über Anzeigen kann die Person dann Informationen erhalten und gegebenenfalls gewarnt werden. Zur Testung werden 24 Kochdurchgänge durchgeführt, wobei jedes Rezept doppelt ausgeführt wird, einmal fehlerfrei und ein weiteres Mal mit zwei bis vier von der Testperson aus einem Katalog ausgewählten Fehlern. Das System kann die Fehler mit einer Genauigkeit von 92 % erkennen.

Basharudin et al. untersuchen mit ihrem System hauptsächlich den Feedback-kanal, um eine möglichst effektive Unterstützung beim Kochen für Alzheimer-patient*innen zu bieten [12]. Hierzu werden jeweils vier grüne und vier rote LEDs in der Nähe von Küchenutensilien angebracht (vgl. Abbildung 3.4). Der Zweck der LEDs ist es, den Testpersonen Hinweise zu geben. Hierzu leuchten die grünen LEDs, wenn die Person mit der Benutzung des Gegenstands richtig liegt und rot, wenn das Vorgehen nicht korrekt ist.

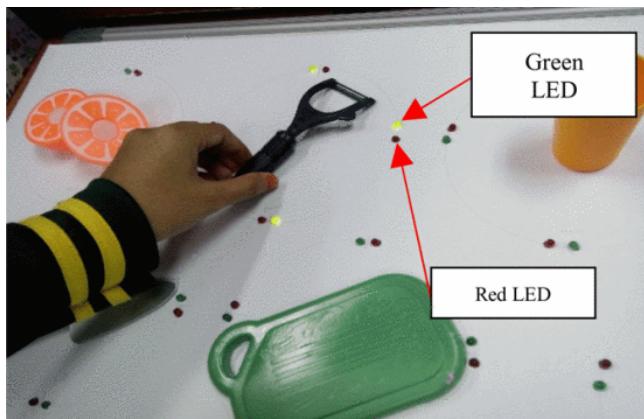


Abbildung 3.4: Feedbacksystem mittels LEDs von Basharudin et al. [12]

Eine Kamera erkennt, welcher Gegenstand gegriffen wird, so dass das System mit dem Wissen, welcher Kochprozessschritt aktuell durchgeführt wird, die Testperson informiert, ob der Gegenstand im aktuellen Schritt richtig oder falsch ist. Die Arbeitsgruppe kommt zum Schluss, dass die Testpersonen die Hinweise verstehen und mit dieser Unterstützung den Kochprozess vollständig durchführen können. Kosch et al. betrachten mit ihrem System ebenfalls hauptsächlich die Möglichkeiten des Feedbacks beim Kochen [66]. Mit ihrem System ist es möglich, Informationen als Videoprojektion und akustisch mittels Lautsprechern an die Nutzenden zu leiten. Hierbei werden kurze Videoclips eingespielt, die den Vorgang vormachen und so lange wiederholt werden, bis der entsprechende Teilschritt von der Testperson erfolgreich durchgeführt wurde. Zusätzlich gibt es über den Projektor

die Möglichkeit, direkt ein grünes Licht auf den korrekten Gegenstand als Hinweis zu projizieren (vgl. Abbildung 3.5).



Abbildung 3.5: Versuchsaufbau von Kosch et al. [66]

Ihre Testung mit zehn Testpersonen in einem Pflegeheim konnte zeigen, dass die Proband*innen mittels akustischem Feedback die Rezepte erfolgreich kochen können, die Videos werden allerdings nur betrachtet, wenn das akustische Feedback nicht verstanden wird. Dennoch zeigt sich, dass eine Kombination beider Informationskanäle gut für das Verständnis der Proband*innen für den Prozess ist.

Feedbacksysteme werden häufig auch in industriellen Anlagen erforscht. Sie ermöglichen die kontinuierliche Überwachung von Prozessen, um so Zeit- und Ressourcen zu optimieren oder gefährliche Situationen rechtzeitig zu erkennen und zu vermeiden. Li et al. haben hierzu einen intelligenten Schutzhelm entwickelt, der basierend auf Beschleunigungsdaten und einem eingebauten EEG die Erschöpfung der Person erkennt [77]. Das System beinhaltet die Möglichkeit eines Feedbacks mittels Vibrationsmotoren zur Sensibilisierung in einer erkannten gefährlichen Situation und kann bei weiterer Überschreitung bestimmter Erschöpfungsschwellen auch verknüpfte Geräte zur Sicherheit abschalten.

3.2 Künstliche Intelligenz für smarte Sensorsysteme

Künstliche Intelligenz wird zunehmend häufig in vielen Bereichen des Lebens eingesetzt, so auch bei der Aktivitäts- und Handlungserkennung. Der nachfolgende Abschnitt beschreibt daher die für diese Arbeit relevanten Datensätze und Algorithmen.

3.2.1 Datensets

Im Rahmen der Untersuchung des Stands der Wissenschaft und Technik der künstlichen Intelligenz für smarte Sensorsysteme werden im ersten Schritt relevante Datensets beschrieben und mögliche Grenzen aufgezeigt. Bei der Entwicklung von KI spielen Datensätze stets eine grundlegende Rolle, daher ist die Recherche nach vorhanden Daten eine gute Ausgangsbasis. Davon unberührt ist allerdings der Zugang zu den Daten, der z.B. lizenziert rechtlich eingeschränkt sein kann, oder die tatsächliche Passung der Daten zum Entwicklungsziel der KI.

Datensets basierend auf Beschleunigungsmesswerten

Der *Human Activity Recognition Using Smartphones* (HARUS) Datensatz [59] enthält Daten von 30 Personen im Alter zwischen 19 und 48 Jahren. Jede Person führte sechs Aktivitäten des täglichen Lebens (ADL, vgl. Kapitel 2.1.2) durch, nämlich Gehen, Treppen auf- und absteigen, Sitzen, Stehen und Liegen. Die Daten wurden erhoben, indem jeder dieser Testpersonen ein Smartphone an der Hüfte getragen hat (vgl. Abbildung 3.6).

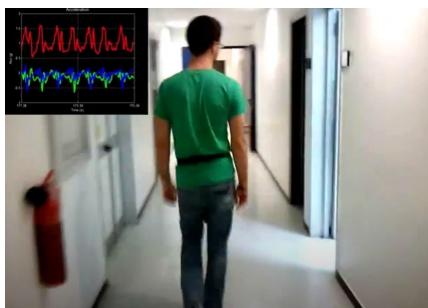


Abbildung 3.6: Ausschnitt aus der Generierung des HARUS-Datensets [59]

Dabei werden mit jeweils 50 Hz die Beschleunigungen in den drei Raumachsen, als auch die Winkelgeschwindigkeiten um die Raumachsen aufgezeichnet. Eine parallele Videoaufzeichnung wird genutzt, um die Daten anschließend zu annotieren. Auf Basis derselben Daten wurde von dieser Arbeitsgruppe noch der *Smartphone-Based Recognition of Human Activities and Postural Transitions* Datensatz erzeugt [60]. Dieses Set erweitert den HARUS-Datensatz um Übergänge zwischen den Bewegungen, namentlich Stehen-Sitzen, Sitzen-Stehen, Sitzen-Liegen, Liegen-Sitzen, Stehen-Liegen und Liegen-Stehen.

Beide Datensätze fallen also in den Bereich der *am Körper getragenen Sensorik* (vgl. Kapitel 2.1.3). Die aufgezeichneten Aktivitäten sind dabei als Ganzes erfasst. Eine Betrachtung als Prozess, d.h. eine Verkettung von Teilschritten die zum Ganzen führt, wird nicht betrachtet.

Der *Cooking Activity Dataset with Macro and Micro Activities* (CADMMA) Datensatz umfasst 60 Kochdurchgänge von vier Personen, die jeweils drei verschiedenen Rezepte fünf mal kochten [71]. Die Daten werden dabei durch jeweils zwei Smartphones, die am rechten Arm und der linken Hüfte angebracht sind und einem Motion Capture System mit 29 Markern aufgezeichnet, vergleiche hierzu Abbildung 3.7.

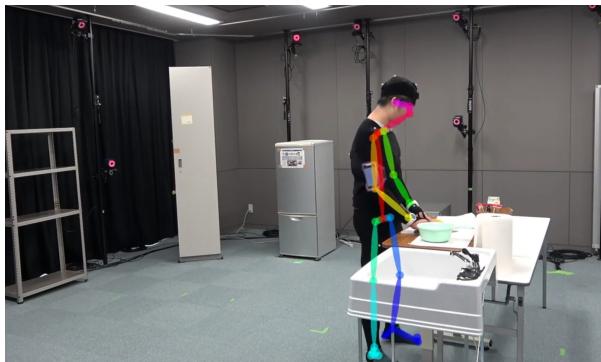


Abbildung 3.7: Einzelbild aus dem CADMMA-Datensatz [71]

Dieser Datensatz beinhaltet somit Aufzeichnungen von Beschleunigungsdaten aus am Körper getragener Sensorik, wird aber um Bildinformationen ergänzt. Die Proband*innen bereiten darin Sandwiches, Fruchtsalat und Müsli zu, welche im Rahmen dieses Datensatzes *Macro Activities* genannt werden. Die Personen taten dies auf Basis eines Skripts, sollten dabei aber so natürlich wie möglich agieren. Zu den annotierten, hier sogenannten *Micro Activities*, gehören Dinge aus Schubladen, Regalen und einem Kühlschrank nehmen, Früchte und Gemüse schneiden und Äpfel, Bananen und Mandarinen schälen.

Dieser Datensatz enthält damit Daten und Informationen zu Teilschritten eines Prozesses, die gemeinsam einen Gesamtprozess bilden. Allerdings stützen sich diese Informationen auch hier auf Bildauswertung und Beschleunigungsdaten, eine Lokalisierung von Gegenständen war nicht vorgesehen.

Datenset basierend auf Signalstärkemessungen

Der *RSSI Dataset for Indoor Localization Fingerprinting* Datensatz umfasst 105 sogenannte *Fingerprints* (deutsch: Fingerabdrücke), die in drei verschiedenen Szenarien generiert wurden [127]. Fingerprinting bezeichnet eine Technik, bei der in einer Umgebung mittels eines mobilen Empfängers die Signalstärke in verschiedenen Abständen zu einer Reihe von Sendern gemessen wird, so dass eine Signalstärkekarte der Umgebung entsteht.

In Szenario 1 wird ein 6 x 5,5 m großer Raum vermessen, bei dem alle möglichen

Interferenzquellen entfernt werden. Der Abstand der Fingerabdrücke beträgt dabei 0,5 m zueinander. Szenario 2 besteht aus Daten aus einem 5,8 x 5,3 m großen Raum, bei dem mit Absicht zusätzliche Transmitter als Störquellen hinzugefügt werden, um möglichst viele Interferenzen zu erzeugen. Szenario 3 ist abschließend in einer 10,8 x 7,3 m großen Arbeitsumgebung aufgenommen, in welcher typische Interferenzen durch WLAN- und Bluetooth-Geräte zu erwarten sind.

Diese Datensets schließen Beeinträchtigungen durch den menschlichen Körper selbst aus, da die Messungen ohne anwesende Personen durchgeführt werden. Durch das Fingerprinting ist eine Lokalisierung in diesen Räumen durch mobile Transmitter mittels Messungen zu fix installierten Sensoren erwartbar. Eine Lokalisierung durch Messungen zwischen den mobilen Sensorknoten ist jedoch nicht vorgesehen.

Datenset basierend auf Videos

Der Depth-included Human Action (DHA) Datensatz umfasst 357 Videos, welche sowohl RGB- als auch Tiefeninformationen beinhalten [79]. In diesen Videos sind 17 verschiedene körperliche Aktivitäten aufgezeichnet, die von 12 Männern und 9 Frauen durchgeführt wurden. Abbildung 3.8 zeigt ein einzelnes Frame aus einem dieser Videos.



Abbildung 3.8: Einzelbild mit Farb-, Tiefen- und Silhouetteninformationen (links, mittig, rechts) bei verschiedenen Lichtbedingungen (oben, unten) aus dem DHA-Datensatz [79]

Die Bewegungen umfassen dabei Bücken, Springen (Hampelmann), das Springen auf der Stelle, das Vorwärtspringen, einhändiges und zweihändiges Winken, Rennen, seitwärts Bewegen, Hüpfen, Laufen, Gehen, Klatschen, Fußtritte, Armschwingen, Pitch, Schwingen, Boxen und Tai-Chi. Diese Aktivitäten stellen eine große Bandbreite an Bewegungen dar, stationäre Aktivitäten wie Sitzen, Stehen oder Liegen sind jedoch nicht enthalten, es werden auch keine Gegenstände in Betracht gezogen, um den Kontext der Bewegung näher eingrenzen zu können.

Die Aufstellung dieser Datensätze ist nicht abschließend, stellt aber dennoch die zum aktuellen Zeitpunkt relevanten, potentiell zugänglichen, Datensätze für diese Arbeit dar. Es wird deutlich, dass gerade im Umfeld des Kochens eine unzureichende Datenlage vorherrscht, vor allem in Bezug auf *objektgebundene* Sensornetzwerke (vgl. 2.4.2).

3.2.2 Architekturen und Methoden

Fu et al. führten eine Reihe von Experimenten basierend auf dem DHA Datensatz durch [45]. Sie entwickeln hierzu eine Architektur, die auf einem LSTM-Netz mit Autoencoder basiert. Der Autoencoder wird hierbei genutzt, um eine unüberwachte Featureextraktion durchzuführen, so dass letztlich ein sogenanntes *Wörterbuch* (engl. Dictionary) basierend auf den Videodaten entsteht. Um die Performanz zu erhöhen wird zudem ein LSTM-Netz trainiert, um Redundanzen in den einzelnen Videobildern zu eliminieren und so nur relevante Keyframes zu erhalten.

Abbildung 3.9 zeigt die 25 automatisch generierten Dictionaries aus den Bewegtbildern. Die helleren Bereiche repräsentieren die Stellen mit mehr Bewegung in den Videosequenzen, die zu einer bestimmten Aktivität im DHA-Datensatz gehört. Durch Kombination der Keyframes mit den Dictionaries ist es mittels einer Support Vector Machine möglich, Aktivitäten in Videosequenzen zu klassifizieren. Die Gruppe erreichte mit dieser Methode eine Genauigkeit von 89,8 %.

Aranda und Vinagre schlagen ebenfalls ein kamerabasiertes System vor, welches auf Farbbildsequenzen einzelne Gegenstände und ihre Positionsänderungen



Abbildung 3.9: Darstellung der automatisch generierten Dictionaries [45]

bestimmen kann [7]. Für die Objektdetektion setzen sie auf ein Chrominanzhistogramm, die Interaktion mit den Objekten wird mit vier verschiedenen Aktivitäten verknüpft: Hinzufügen, Entfernen, Bewegen und unverändert. Über den Zeitverlauf soll so aus der Abfolge aus Objekten und ihren zugewiesenen Aktivitäten auf das gekochte Rezept geschlossen werden. Die vier unterscheidbaren Rezepte sind Kaffee mit Milch, heiße Schokolade, Müsli und ein Saftgetränk zubereiten.



Abbildung 3.10: Die farblich unterscheidbaren Utensilien zur Zubereitung der Rezepte [7]

Zur Durchführung der Rezepte stehen die in Abbildung 3.10 dargestellten Utensilien bereit. Von jedem Objekt wurden dazu 170 Fotos aus fünf verschiedenen Ansichten in unterschiedlichen Umgebungen aufgenommen. Mit diesem Vorgehen konnten 20 Videosequenzen perfekt zugeordnet werden. Auf diese Weise

entspricht das Vorgehen dieser Arbeitsgruppe dem sogenannten *Bag-of-Words*-Modell, bei dem die Reihenfolge einer Sequenz nicht relevant ist. Über eine Wahrscheinlichkeitsrechnung lässt sich dann aus einzelnen Zuständen, bestehend aus Aktivitäten und Utensilien, eine bestimmte Handlung zuordnen.

Andere Gruppen, wie Zou et al. setzen nicht auf kamerabasierte Systeme, sondern in diesem Falle auf zwei gewöhnliche WLAN-Router, die im Rahmen ihrer Untersuchungen so eingestellt werden, dass einer der beiden stets sendet und der andere empfängt [148]. Die Router werden drei Meter voneinander, in 1,5 Meter Höhe, platziert, in jeweils einem Konferenzraum (Abbildung 3.11, links) und einem Apartment (Abbildung 3.11, rechts).

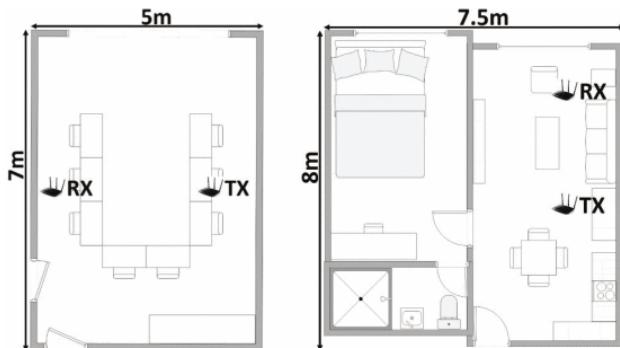


Abbildung 3.11: Die in Zou et al. untersuchten Räume mit Positionen der WLAN-Router [148]

Das Team macht sich dabei die Veränderungen in der WLAN-Signalausbreitung, hervorgerufen durch eine Versuchsperson im selben Raum, zunutze. Diese Signalstärkeänderungen sind über die Kanalzustandsinformationen (engl. Channel State Information, CSI) in gebräuchlichen Netzwerkarten ohne zusätzliche Messinstrumente zugänglich. Auf den CSI aufsetzend, entwickelt die Gruppe eine Maschine-Learning-Pipeline, die zunächst mittels Autoencoder nur relevante Informationen aus den CSI-Frames extrahiert, das Rauschen wird dabei unterdrückt. Im Anschluss interpretiert ein CNN die einzelnen Frames, um lokale Features zu lernen. Um zeitliche Abhängigkeiten zu erkennen, werden diese Informationen

dann durch ein LSTM-Netz gespeist, um schließlich mittels einer vollvernetzten Ausgangsschicht den Zustand der Person anzuzeigen. Auf diese Weise konnte sitzen, liegen, stehen, gehen und laufen einer einzelnen Testperson mit einer Genauigkeit von 97,6 % klassifiziert werden. Auch hier werden keine Teilprozesse untersucht.

Su et al. installieren sechs Bluetooth-Sender und messen anschließend den RSSI zu Bluetooth-fähigen Smartwatches, die von Proband*innen am Handgelenk getragen werden [130]. Zusätzlich werden Beschleunigungen und Winkelgeschwindigkeiten mit der Smartwatch selbst aufgezeichnet. Die RSSI Messungen werden dann mittels Random Forest klassifiziert, die Bewegungsdaten mittels einem Clusteringalgorithmus. Beide Informationen werden dann fusioniert, um zwischen kochen, Medikamenteneinnahme, essen, sitzen und gehen zu unterscheiden. Die Gruppe erreicht so eine Präzision von 92,1 % für *essen* und bis zu 99,1 % für *Medikamenteneinnahme*.

Pham und Olivier entwickeln ein System, welches beim Kochen Unterstützung bieten soll [106]. Hierzu installieren sie Beschleunigungssensoren an drei verschiedenen Messern und einem Löffel, um damit elf verschiedene Bewegungsaktionen erkennen zu können. Hierzu sollten 20 Testpersonen jeweils einen gemischten Salat zubereiten (vgl. Abbildung 3.12).

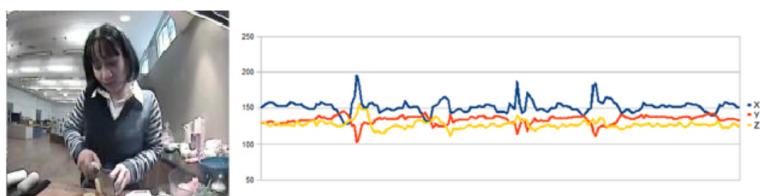


Abbildung 3.12: Beschleunigungen beim Schneiden [106]

Die erkannten Bewegungsaktionen waren Hacken, Schälen, Schneiden, Würfeln, Entkernen, Streichen, Essen, Umrühren, Löffeln, Kratzen und Schaben. Zur Klassifizierung untersucht das Team Naive Bayes-Klassifikatoren, Entscheidungsbäume und Bayes'sche Netze. Das entscheidungsbaumbasierte System erreicht eine

Gesamtgenauigkeit von 82,9 %, das Bayes'sche Netz 75,5 % und schließlich der Bayes-Klassifikator 72,7 %. Dieses Team betrachtet zwar die Teilprozesse, die zusammen den Gesamtprozess ergeben, dieser Gesamtprozess dient aber nur als Rahmenhandlung zur Erzeugung der Daten und wird nicht weiter in Betracht gezogen.

3.3 Usability Methoden zur Testung der Gebrauchstauglichkeit

Das Ziel dieser Arbeit ist nicht nur die Entwicklung eines technischen Systems, sondern auch dessen Überprüfung auf Nützlichkeit und Anwendbarkeit. Hierfür sollen sogenannten Usability-Testing-Methoden zur Anwendung kommen (vgl. Kapitel 2.2), daher folgt nun ein Überblick über gängige Testmethoden, die in Forschung und Praxis genutzt werden.

Maramba et al. machen in ihrem Überblicksartikel zu Usability-Testing im Rahmen von Gesundheitsapplikationen deutlich, dass zwar die Anzahl der entwickelten Apps exponentiell zunimmt, hingegen die Häufigkeit von Usability-Testing-Berichten zu eben jenen Apps nicht in gleichem Maße wächst [84]. Hieraus ergibt sich eine Diskrepanz, die vermuten lässt, dass kein konsequentes Usability-Testing für alle Applikationen in diesem Kontext eingesetzt wird.

Die Forschungsgruppe kam zum Schluss, dass Fragebögen die am meisten genutzte Methode sind und für eine allgemeine Messung der Usability geeignet sind, Nachteile aber darin liegen, dass konkrete Probleme die bei der Nutzung eines Systems auftreten und bei der Entwicklung adressiert werden müssen, nicht exakt bestimmt werden können. Hierfür könnten qualitative Methoden geeigneter sein. Weiterhin haben sie beobachtet, dass immer häufiger mehrere Methoden für die selbe Untersuchung gleichzeitig angewandt werden.

Abbas et al. stellten 2023 zwei Methoden gegenüber: Einerseits den UMUX Fragebogen, der den quantitativen Methoden zuzuordnen ist und andererseits eine

teilnehmende Beobachtung, die eine qualitative Methode ist. UMX steht für *Usability Metric for User Experience* und besteht aus einer Abfrage mit vier Fragen, wobei zwei davon positiv und zwei davon negativ formuliert sind. Die Fragen werden von den Testpersonen mit einer 5-Punkte Likert Skala schriftlich beantwortet. UMX wurde als zeitsparende Alternative zur System Usability Scale von Kraig Finstad bei Intel entwickelt, mit dem Ziel einer vergleichbaren Aussagekraft [41].

Die *teilnehmende Beobachtung* beschreibt eine Methode der Feldforschung, bei der Forschende bei der Durchführung der Versuche durch die Testpersonen anwesend sind und unter Umständen sogar an der Interaktion teilnehmen. Ziel ist die unmittelbare Erfahrung von Verhalten, Handlungen und Denkmustern der Proband*innen [14].

Abbasi et al. führten 309 Tests mit einer Reihe von Probanden durch, wobei jede Testperson entweder eine, zwei oder drei Smartphoneapplikationen testete: YouTube, Gmail und Google Maps. Die Proband*innen erhielten jeweils fünf Aufgaben zu einer App, z.B. *Sende eine Email* und wurden dabei beobachtet und füllten im Anschluss den UMX Fragebogen aus. Beobachtet wurde die Zeitdauer für die Ausführung der Aufgabe, die Anzahl der Fehler, die Vollständigkeit und ob die Testpersonen die Aufgabe abbrachen.

Alle Applikationen wurden von den Testpersonen im Schnitt mit über 85 bewertet, d.h. als *sehr gut*. Bei der Beobachtung verzeichneten die Forschenden hingegen unterschiedliche Fehlerraten und Zeitbedarfe, die vermuten lassen, dass die erhobene objektive Usability schlechter ausfallen müsste. Die Forschungsgruppe schließt daraus auf eine Diskrepanz zwischen Fragebogen und Beobachtung und stellt die Hypothese auf, dass Testpersonen in Fragebögen dazu neigen, sozial akzeptierte Antworten anzukreuzen, wenn sie beobachtet werden.

Weegen et al. begleiteten die Entwicklung ihres Systems mit verschiedenen Usability-Methoden. Das System dient der Aktivierung von gesundheitlich beeinträchtigen Personen und soll dazu beitragen, einen aktiveren Lebensstil zu führen. Das System besteht aus einer Smartphoneapplikation, einem Aktivitätssensor (vgl. Abbildung 3.13) und einer Webseite [138].



Abbildung 3.13: *It's LiFe!* Smartphone und Aktivitätssensor [140]

Das Team unterteilt die Usability-Testung in vier Phasen, die die Entwicklung des Systems begleiteten. In der ersten Phase wurden sechs Testpersonen beobachtet, wie sie das System nutzen und anschließend aufgefordert, Eindrücke aufzuschreiben und einen nicht-standadisierten Fragebogen mit zehn Fragen auf einer 7-Punkte Likert Skala zu beantworten. Die Ergebnisse flossen in die Weiterentwicklung des Prototypen ein.

In der zweiten Phase fand eine qualitative Erhebung mit einem *Thinking Aloud* Ansatz statt, bei der die elf Proband*innen aufgefordert wurden, ihre Gedanken zum System zu verbalisieren während sie sieben vorgegebene Aufgaben ausführten [16]. Anschließend sollten die Proband*innen den *Post-Study System Usability Questionnaire* Fragebogen (PSSUQ) ausfüllen, der aus 19 Fragen besteht, die jeweils mit einer 7-Punkte Likert Skala beantwortet werden [75].

Die dritte Phase besteht aus einem Pilottest mit 20 Testpersonen, die das System in ihrem Alltag einzusetzen. Bei regelmäßigen Treffen wurden die Teilnehmenden nach ihren Erfahrungen gefragt, die Antworten aufgezeichnet und transkribiert. Zusätzlich beantworteten die Testpersonen den *Software Usability Measurement Inventory* Fragebogen (SUMI), bei dem 50 Fragen auf einer 3-Punkte Likert Skala beantwortet werden [64].

Die vierte Phase fand abermals in einer Laborumgebung mit fünf Proband*innen statt. Die Testpersonen sollten nach der Testung das *Microsoft's desirability toolkit* verwenden, bei dem den Personen eine Auswahl von 118 Wörtern vorliegt,

wobei 60% der Wörter positiv behaftet sind, und 40% negativ [144]. Die Aufgabe besteht dann darin, zunächst alle Wörter einzukreisen, die die Testperson als zutreffend auf das System erachtet. Im zweiten Schritt sollen davon fünf ausgesucht werden, die am besten zur persönlichen Reaktion der Testperson passen und ihre Auswahl mündlich erklären.

Es zeigt sich eine Fülle an Testmethoden, die je nach Entwicklungsstand eingesetzt werden können, wobei die Zuordnung einer Methode zu einer bestimmten Phase nicht zwingend durch die Methode selbst vorgegeben ist, sie muss vielmehr in Abhängigkeit von äußeren Faktoren, wie beispielsweise Zielgruppe, Zeitbedarf und gewünschte Informationen gezielt ausgewählt werden.

3.4 Diskussion und Fazit

Zusammenfassend wird deutlich, dass der aktuelle Stand der Technik und Wissenschaft verschiedene Lücken im Bezug auf eine einfache und detailreiche Handlungserkennung aufweist. Typischerweise werden Handlungen aktuell als Ganzes klassifiziert, jedoch nicht einzelne Schritte innerhalb des Handlungsprozesses. Beispielsweise wird korrekt erkannt dass eine bestimmte Speise zubereitet wurde, nicht aber explizit die einzelnen Rezeptschritte, die zum fertigen Gericht führten. Infolgedessen wird eine Unterstützung der handelnden Personen während der Durchführung erschwert, da das System nicht in der Lage ist, einzelne Teilschritte sicher zu interpretieren und entsprechendes Feedback zu den Teilschritten zu geben, höchstens zum Gesamtprozess als Ganzes, sofern er unabhängig von den Teilschritten zu Laufzeit schon erkennbar ist.

Lücken zeigen sich auch in Form der Usability. Die vorgestellten Systeme setzen häufig auf eine Projektion von Videos oder anderen Informationen direkt auf die Arbeitsfläche. Dies sorgt zwar einerseits für eine einfache Lesbarkeit, andererseits muss der zugehörige Beamer gut platziert sein, damit es zu keinen Verdeckungen kommt, wodurch die Usability wieder leiden würde. Zudem bedeutet der Einsatz

von Projektionstechnik einen erhöhten Installationsaufwand, häufig auch gekopelt an eine Kalibrierung, um eine planare Projektion zu ermöglichen.

Weiterhin wird deutlich, dass viele aktuelle Systeme auf am Körper getragener Sensorik basieren oder sich auf visuellen Analysen mittels Kameras stützen. Eine solche Herangehensweise bietet verschiedene Vorteile: Kameras werden beispielsweise stationär im Raum installiert, eine Energieversorgung ist daher, bei entsprechendem Installationsaufwand, stets gegeben. Eben dieser Installationsaufwand, in Kombination mit einer noch immer geringen Akzeptanz von Kamerasystemen im privaten Umfeld schmälern jedoch die Attraktivität dieser Technologie für einen Praxiseinsatz in Form eines Produkts.

Am Körper getragene Sensorik umgeht diese Problematik, birgt aber andere Herausforderungen. Hier ist eine Stromversorgung mittels Batterie oder Akkumulator notwendig, die regelmäßig ausgetauscht oder aufgeladen werden müssen. Weiterhin werden, von Person zu Person unterschiedlich, aber nicht beliebig viele Sensoren am Körper akzeptiert, da diese an- und ausgezogen werden müssen. Zusätzlich muss gegebenenfalls darauf geachtet werden die Sensoren korrekt und an richtiger Position anzulegen. Außerdem führen am Körper getragene Geräte zu Einschränkungen bei der Hygiene, z.B. dem Duschen. In Hinblick auf den Stand der Wissenschaft und Technik bezüglich der Anwendung von künstlicher Intelligenz zeigt sich der üblicherweise hohe Energiebedarf: Die KI-Systeme werden für gewöhnlich auf einem leistungsstarken Hintergrundsystem ausgeführt, wodurch sich die Zeiten für Training und Inferenz der Modelle kurz halten; sie sind entsprechend groß und leistungsfähig. Nur sehr selten werden KI-Modelle trainiert und ausgerollt, die direkt auf den üblicherweise leistungstechnisch stark eingeschränkten Sensorknoten selbst funktionieren. Weiterhin zeigt sich, dass zwar eine Reihe von Datensätzen existieren, im Umfeld des Kochens und mit an Objekten angebrachten Sensoren ist zum Zeitpunkt der Recherche jedoch kein geeigneter Datensatz bekannt.

Insgesamt zeigt sich Potential für die Erforschung von Methoden und Algorithmen für die Prozessanalyse mittels Sensoren, die an Objekten angebracht sind mit dem Ziel, Subprozesse erkennen zu können. Es existiert zum aktuellen Zeitpunkt kein Sensornetzwerk zur Erfassung von Kochprozessen, welches einzelne Rezeptschritte unterscheiden kann, um damit auf das Rezept selbst schließen zu können. Außerdem ist kein eingebettetes System bekannt, dass dann währenddessen, durch die Kenntnis des Teilprozesses, ein Feedback an die nutzende Person gibt, um diese beim Kochen zu unterstützen. In der wissenschaftlichen Literatur finden sich Ansätze mittels am Körper getragener Sensorik oder auf Basis von Kamerasystemen, die jedoch ihrerseits Nachteile mit sich bringen.

Es besteht Forschungsbedarf hinsichtlich einer Gesamtintegration eines solchen Systems, besonders auch im Hinblick auf die Praxistauglichkeit und Nutzenfreundlichkeit. Hierbei muss unter anderem berücksichtigt werden:

- Eingeschränkte Akkukapazität
- Minimales Bauvolumen
- Robustheit der Inferenz gegenüber unterschiedlichen Personen
- Nutzenstiftendes Feedback

Außerdem gilt es zu untersuchen, inwieweit eine Inferenz auf dem Sensorknoten selbst Vorteile gegenüber klassischen Methoden bietet.

4 Forschungsfrage zur Überwindung des aktuellen Stands der Wissenschaft und Technik

Die Forschung in der vorliegenden Arbeit bezieht sich im Rahmen der bereits im Kapitel 1.1 aufgezeigten Rahmenbedingungen auf die Konzeptionierung, Entwicklung und Anwendung eines Systems zur Unterstützung von älteren Personen in ihrer gewohnten Umgebung. Erklärtes Ziel ist hierbei, dass eine Installation von Komponenten in diese Umgebung möglichst vermieden werden soll und sich ein entsprechendes System einfach und intuitiv bedienen lässt. Hieraus ergibt sich auch die nachfolgende Forschungsfrage, welche den Kern dieser Arbeit bildet:

Wie kann Technologie im Smart Home unterstützen, um die Autonomie von Senior*innen im Alltag zu verbessern?

Die Beantwortung dieser Leitfrage findet innerhalb dieser Arbeit primär im Küchenumfeld statt. Erklärtes Ziel ist es, dass die entwickelten Methoden und Prinzipien auch auf andere Domänen möglichst einfach übertragbar sind. Nachfolgende Teilfragen konkretisieren die Leitfrage weiter und geben einen ersten Ausblick auf die besonderen Herausforderungen in diesem Kontext:

FF1 Wie kann künstliche Intelligenz eingesetzt werden, um das Kochen zuhause zu unterstützen?

FF2 Welche Anforderungen stellen sich an die Hardware zur Erfassung eines Kochvorgangs?

FF3 Wie können die erforschten Methoden und Prinzipien gestaltet werden, um einen Einsatz auch in anderen Domänen zu ermöglichen?

Anhand dieser Fragen wird ein System entwickelt, welches aus verschiedenen Bausteinen besteht. Jeder Baustein für sich wurde mit dem Ziel entworfen möglichst universell einsetzbar zu sein, ein Zuschnitt auf das Küchenanwendungsbereich oder auch dem Smart Home wird vermieden. Hierdurch kann das System als Ganzes oder auch Teilespekte hieraus potentiell auf andere Domänen übertragen werden. Durch eine stetige Inbezugnahme von realweltlichen Herausforderungen, wie beispielsweise Kosten für die Produktion oder Anforderungen an die Betriebszeit für eine sinnhafte Anwendung, soll ein Transfer über die Grenzen einer wissenschaftlichen Arbeit gewährleistet sein.

5 Entwurf eines KI-gestützten Sensornetzwerks zur Prozessoptimierung

Bevor mit der Entwicklung eines Systems begonnen wird, welches eine Beantwortung der Forschungsfragen ermöglicht, soll in diesem Kapitel zunächst das Gesamtkonzept vorgestellt werden. Aus diesem Konzept und weiteren Randbedingungen ergeben sich wiederum Anforderungen, welche ihrerseits dann als Leitbild für die Entwicklung herangezogen werden können.

5.1 Konzeptionierung des Gesamtsystems

Für die Entwicklung des Gesamtsystems wird ein iteratives Vorgehen gewählt, wodurch sich eine Lösung schrittweise ergibt. Nach dem initialen Entwurf, welcher die Auswahl von Komponenten beinhaltet, werden bereits erste Tests durchgeführt. Das sich daraus ergebende Feedback fließt dann zurück in die weitere Entwicklung. Mit diesem Vorgehen ergeben sich verschiedene Entwicklungsabschnitte, an deren Ende jeweils eine Testung steht.

Im Folgenden sind die grundlegenden Komponenten des Gesamtsystems übersichtsartig dargestellt und es werden erste, grundlegende technologische Entscheidungen beschrieben. Die detaillierte Schilderung der Ausgestaltung der jeweiligen Komponenten folgt im Anschluss in den nachfolgenden Kapiteln.

Zur Beantwortung der Forschungsfrage (Kapitel 4), wie Technologie im Smart Home unterstützen kann, muss zunächst eine Entscheidung zur eingesetzten Technologie getroffen werden, um hieraus entsprechende Systeme entwickeln zu können. Wie im Abschnitt 1.3 *Umfeld* dargelegt, begründet sich die vorliegende Arbeit auf den Vorarbeiten eines Forschungsprojekts (*AuRorA*) im Küchenumfeld. Speziell in diesem Kontext zeigte sich schnell, dass der Einsatz von Kamerasystemen in der Häuslichkeit ungern akzeptiert wird, auch andere Untersuchungen zum Thema Akzeptanz von Kamerasystemen im privaten Umfeld, wie beispielsweise von Arning et al. [9] und Ziefler et al. [147] bestätigen dies.

Im Rahmen der vorliegenden Arbeit wird daher ein System entwickelt, welches keine Kamera benötigt. Neben den zuvor erwähnten Akzeptanzproblemen werden auf diese Weise auch weitere Nachteile der kamerabasierten Lösungen umgangen, beispielsweise Verdeckungen durch die Nutzenden selbst oder anderen Objekten, wie es bei Arbeiten in der Küche zu erwarten ist.

Auf Grundlage der Vorarbeiten im Projekt AuRorA und der zu erwartenden größeren Akzeptanz in der Nutzengruppe wird im Rahmen dieser Arbeit ein Sensornetzwerk erforscht und aufgebaut werden, bei dem die einzelnen Sensorknoten direkt an relevante Objekte, beispielsweise Küchenutensilien, angebracht werden. Entsprechend der Definition in Abschnitt 2.4.2 *Sensornetzwerke* wird also ein *Object-tagged*-Netzwerk entwickelt.

Darüber hinaus wird für die Umsetzung das geplante Netzwerk in seine Teilkomponenten *Sensorknoten*, *Hintergrundsystem*, *eingebettete künstliche Intelligenz* und *Feedbacksystem* gegliedert. Die geplante Entwicklung wird nachfolgend für die einzelnen Teilsysteme weiter spezifiziert.

1. Sensorknoten

Ausgangspunkt für die Gesamtsystementwicklung ist die Konzeptionierung und erste Implementierung eines Sensorknotens. Die Auswahl der dafür vorgesehenen Bauteile stellt im weiteren Verlauf die Weichen für nachfolgende Entwicklungen. Beispielsweise hat die Speichergröße des eingesetzten Mikrocontrollers direkte Auswirkungen auf die Umsetzbarkeit von eingebetteten Machine Learning Algorithmen.

Auch die Auswahl der durch die Hardwarearchitektur vorgegebene Funktechnologie hat maßgeblichen Einfluss auf verschiedene Parameter des Systems: Einerseits auf die Akkulaufzeit und damit indirekt auch auf die Benutzungsfreundlichkeit, andererseits auf die möglichen Übertragungsraten zwischen den Sensorknoten und zum Serversystem. Die Datenrate und damit die Möglichkeit zur Übertragung von unterschiedlich komplexen bzw. umfangreichen Daten hat ihrerseits direkt Auswirkungen auf die Ausgestaltung einer künstlichen Intelligenz zur Inferenz des Kochprozesses.

Die weitere Umsetzung der Sensorknoten und der Aufbau des Netzwerks wird in Kapitel 6 detailliert beleuchtet.

2. Hintergrundsystem

Nach erster Integration der Sensorknoten beginnt parallel die Entwicklung des Backendsystems. Dieses System dient einerseits dazu, mittels der aus den Sensorknoten stammenden Daten KI-Systeme zu trainieren, die den Prozess und dessen Teilschritte erkennen können. Andererseits dient das System im Betrieb dazu, zur Laufzeit die trainierten Modelle auszuführen, um so das Feedback generieren zu können, welches an die Anwendenden zurückgegeben wird. Im Rahmen dieser Arbeit findet Training und Inferenz auf demselben System statt. In der Praxis kann beides getrennt werden, da üblicherweise die reine Inferenz weniger Leistung benötigt als das Training, wodurch ein deutlich kostengünstigeres System zur Inferenz vor Ort bei den Anwendenden zum Einsatz kommen kann.

In Kapitel 7 werden die Schritte bis zum umgesetzten KI-Hintergrundsystem beschrieben.

3. Eingebettete künstliche Intelligenz

Im weiteren Verlauf wird der Einsatz von KI-Modellen auf den Sensorknoten selbst untersucht. Es gilt zu überprüfen, inwieweit eine Inferenz auf dem Sensorknoten selbst Vorteile für das Gesamtsystem bieten kann. Je nach eingesetzter Technologie zur Handlungserkennung mittels Sensorknoten werden unterschiedliche Datenraten zur Übertragung der Informationen benötigt. Ein häufiges Senden und Empfangen steigert wiederum den Energiebedarf und führt zu einer sinkenden Akkulaufzeit. Je nach Funktechnologie kann auch bereits die benötigte Datenrate

selbst zu hoch für die verfügbare Bandbreite des Funksystems sein.

Eine Inferenz auf dem eingebetteten Mikrocontroller des Sensorknotens kann gerade für Letzteres eine Lösung sein, da in einem solchen Falle nicht mehr die Rohdaten, beispielsweise die Beschleunigungswerte eines Accelerometers, übertragen werden müssten, sondern nur noch das Ergebnis der Inferenz selbst, beispielsweise die Erkennung einer bestimmten Bewegung.

Die Implementierung der eingebetteten KI wird in Kapitel 7 dargestellt.

4. Feedbacksystem

Das Ziel ist es, mit dem Feedbacksystem eine Rückkopplung zu den Nutzenden herzustellen, so dass diese während der Tätigkeit unmittelbar unterstützt werden können. Das Feedback muss demnach unmittelbar bei Bedarf erfolgen, beispielsweise wenn ein Teilschritt falsch durchgeführt oder ausgelassen wurde, konkret im Kontext des Kochens also wenn ein Kochschritt vergessen wurde oder die falsche Zutat hinzugefügt wird.

Das Feedback sollte daher möglichst eindeutig und auffällig sein, ohne jedoch zu stören oder abzulenken. Verschiedene Kommunikationskanäle stehen dafür prinzipiell zur Verfügung, wie zum Beispiel akustisch über Töne, haptisch über Vibration oder visuell über Licht. Diese Kanäle gilt es abzuwägen und die erfolgversprechendsten zu implementieren und zu testen. Die Hardwareumsetzung des Feedbacksystems ist in Kapitel 6 aufgeführt, die Evaluationsergebnisse folgen als Endergebnis in Kapitel 8.

Zusammengefasst lässt sich aus den vorherigen Überlegungen ein erstes Systemkonzept, wie in Abbildung 5.1 dargestellt, ableiten, auf der linken Seite ist das Sensornetzwerk zu sehen, welches untereinander Signalstärkemessungen durchführt. Diese Daten werden an das Hintergrundsystem weitergeleitet, welches mittels KI-Verfahren die Messungen analysiert und daraus Teilschritte und Kochrezepte inferiert. Diese sind in der Grafik mit den Klassen 1, 2, 3 usw. bezeichnet. Nach erfolgreicher Inferenz erhalten die Nutzenden ein Feedback, welches über den selben Kanal wie die Messdaten an das Sensornetzwerk zurückgegeben wird, wo es dann durch die Sensorknoten kommuniziert wird.

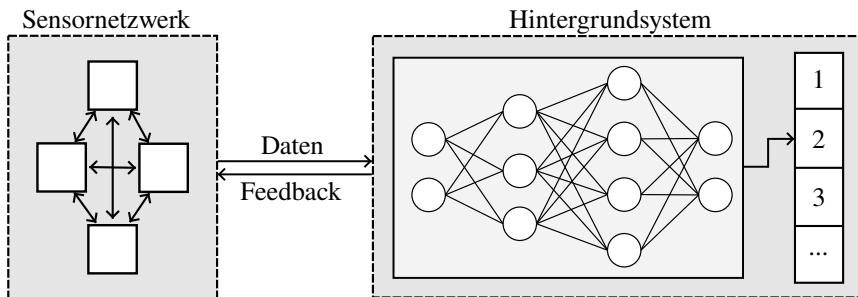


Abbildung 5.1: Gesamtsystemübersicht

Insgesamt sieht das Konzept vor, besonderes Augenmerk auf eine kostengünstige Umsetzung zu legen, mit dem Ziel, eine spätere wirtschaftliche Verwertung zu ermöglichen. Dazu müssen die Sensorknoten aus möglichst wenigen, gut verfügbaren und günstigen Komponenten bestehen. Daraus ergibt sich ebenfalls das Ziel so wenige Varianten wie möglich einzusetzen, um die Verwaltung dieser zu vereinfachen.

Aus diesem Gesamtkonzept werden nun nachfolgend die grundlegenden Anforderungen an das System und seine Subsysteme abgeleitet.

5.2 Anforderungen an das Gesamtsystem

Zur weiteren Entwicklung des Gesamtsystems werden zunächst eine Reihe von Anforderungen erfasst. Diese ergeben sich durch Vorarbeiten aus dem Umfeld der Arbeit (vgl. Kapitel 1.3), Befragungen und Gespräche mit Fachkräften und auch aus der Umgebung, in der das System eingesetzt werden soll. Sie dienen im weiteren Verlauf als Grundlage und Leitlinien für die Entwicklung und Evaluierung des Gesamtsystems.

Nachfolgend werden die Anforderungen sortiert nach Teilsystem aufgelistet.

Anforderungen an die Sensorknoten

- A01 Ihre Ausstattung weicht nicht grundlegend voneinander ab, so dass nicht mehrere unterschiedliche Versionen entwickelt und gepflegt werden müssen.
- A02 Jeder Sensorknoten verfügt über eine Akkulaufzeit von mindestens einer Woche.
- A03 Ein einzelner Sensorknoten ist höchstens 25 mm x 25 mm x 15 mm groß (Höhe/Breite/Tiefe).
- A04 Das Aufladen geschieht durch einen USB-Anschluss oder eine einfache Ladevorrichtung.
- A05 Bauteile werden so ausgewählt, dass eine lange Marktverfügbarkeit besteht.
- A06 Ein Anbringen an verschiedenen Küchenutensilien ist leicht und schadlos möglich.
- A07 Eine Anbringung an handelsüblichen Küchenutensilien ist möglich.
- A08 Die Materialkosten für einen einzelnen Sensorknoten betragen weniger als 10€.
- A09 Jeder Sensorknoten gibt direktes Feedback an die Nutzenden.
- A10 Der verwendete Mikrocontroller lässt eine Entwicklung und Einsatz von eingebetteten Machine Learning Algorithmen zu.

Anforderungen an das Sensornetzwerk

- A11 Die Kommunikation zwischen den Sensorknoten erfolgt drahtlos.
- A12 Die eingesetzte Funktechnologie ermöglicht eine Abstandsbestimmung zwischen den einzelnen Sensorknoten.
- A13 Das Kommunikationsprotokoll basiert auf Standardfunktionalitäten der Funktechnologie, so dass eine Kompatibilität mit Dritten möglich ist.

Anforderungen an das KI-System

- A14 Eine nutzendenunabhängige Klassifizierung ist möglich.
- A15 Teilschritte im Kochprozess, z.B. Umrühren im Topf, können vom System automatisch erkannt werden.
- A16 Das Gesamtrezept, bestehend aus Teilschritten, kann vom System automatisch erkannt werden.
- A17 Der nächste Teilschritt kann vom System vorhergesagt werden, so dass eine Unterstützung durch das Feedbacksystem ermöglicht wird.

Anforderungen an das Feedbacksystem

- A18 Das System gibt den Nutzenden während der Durchführung Rückmeldung, so dass eine Unterstützung zur Laufzeit ermöglicht wird.
- A19 Die Rückmeldung des Systems soll einfach wahrgenommen werden können, aber nicht störend sein.
- A20 Das Feedbacksystem erinnert die Person an Teilschritte im Kochprozess.
- A21 Das Feedbacksystem führt zu einer Aktivierung der Person.

Sonstige Anforderungen

- A22 Die Nutzenden erhalten Feedback zur Laufzeit, d.h. während eines Kochvorgangs.
- A23 Das System wird von den Nutzenden als hilfreich bewertet.
- A24 Das System verwirrt die Nutzenden nicht.

6 Entwicklung des Sensornetzwerks

Nach der Definition der grundlegenden Anforderungen und den daraus resultierenden Konsequenzen für das Design im vorangegangenen Kapitel beginnt in diesem Abschnitt die Umsetzung des Systems. Zunächst wird im vorliegenden Kapitel 6 die Entwicklung der Hardware beschrieben, inklusive der dafür notwendigen Software zur Inbetriebnahme des Sensornetzwerks. In Kapitel 7 folgt die Beschreibung der Entwicklung des Hintergrundsystems und der KI-Algorithmen. Dabei fließen jeweils die Anforderungen ein und werden weiter präzisiert, so dass am Ende ein Gesamtsystem entsteht, welches alle benötigten Funktionen abbildet. Die Entwicklung der Hardware fokussiert sich besonders auf die grundlegende Unterstützung von KI-Methoden, der Fähigkeit Nutzenden Feedback zu geben und einem minimalen Strombedarf, um eine lange Akkulaufzeit zu ermöglichen.

6.1 Auswahl einer geeigneten Lokalisierungstechnologie

Gemäß den grundlegenden Anforderungen A11 bis A13 (vgl. Kapitel 5.2) soll als Lokalisierungstechnik eine Funktechnologie zum Einsatz kommen, kamera-basierte Methoden werden entsprechend hier nicht näher betrachtet. Es ergeben sich wie in Kapitel 2.4.2.3 dargestellt verschiedene Möglichkeiten für die Lokalisierung der Küchenutensilien. Eine Fremdlokalisierung kann ausgeschlossen werden, da diese üblicherweise eine Installation von Referenzsensoren in die Umgebung bedeutet, dies soll in dieser Arbeit explizit nicht weiter betrachtet werden.

Die Musterabgleich-Methode (vgl. Kapitel 2.4.2.3) bedeutet ebenfalls einen erhöhten Installationsaufwand, da hierfür eine möglichst detaillierte Signalstärkekarte der Umgebung eingemessen werden muss, bevor die Sensorknoten dann über einen Abgleich mit der zur Laufzeit gemessenen Signalstärke ihre Position bestimmen können. Diese Methode wird daher ebenfalls nicht weiter betrachtet. Die sprungzählungsbasierte Technik zählt zunächst die Anzahl der Sprünge (*Hopping*), die ein Datenpaket vom eigentlichen Knoten zu anderen Knoten im Netzwerk benötigt. Darüber kann dann mittels Referenzknoten eine tatsächliche Entfernung abgeschätzt werden. Diese Technik ist interessant für Sensornetzwerke, die große Flächen abdecken. In dem in dieser Arbeit angestrebten Szenario in der Küche ist davon auszugehen, dass durch die räumliche Nähe der Knoten zueinander kein Hopping stattfinden würde und daher kein Vorteil aus dieser Methode zu erwarten ist.

Die weitere Auswahl der distanzbasierten Lösung wird auf Grundlage von Kumar et al., Jondhale et al. und Cheng et al. durchgeführt und ist in Tabelle 6.1 dargelegt [69] [58] [31].

Technologie	Messgröße	Genauigkeit	Aufwand	Kosten
Time of Arrival	Zeit	+	-	-
Time Difference of Arrival	Zeitdifferenz	+	-	-
Angle of Arrival	Winkel	++	--	--
Received Signal Strength	Signalstärke	-	++	++

Tabelle 6.1: Gegenüberstellung der verschiedenen distanzbasierten Lokalisierungsmethoden

Die Herausforderung von zeitbasierten Methoden wie Time of Arrival (ToA) und Time Difference of Arrival (TDoA) ist die Notwendigkeit einer möglichst perfekten Zeitsynchronisierung zwischen den Ankernoten, um Ungenauigkeiten bei der Lokalisierung zu vermeiden. Kann eine sehr gute Synchronisierung erreicht werden, zeigt sich das in einer hohen Genauigkeit. Die Kosten erhöhen sich durch diese Notwendigkeit entsprechend, da hier Synchronisierungsmethoden implementiert werden müssen. Beide Methoden sind anfällig für Fehler die durch sogenannten Non-Line of Sight (NLOS) Übertragungen entstehen, d.h. Übertragungen ohne

direkten Sichtkontakt wie Echos eines Signals, die mit Verzögerungen eintreffen. Winkelbasierte Techniken, wie Angle of Arrival (AoA) können eine sehr hohe Präzision der Lokalisierung erreichen, allerdings sind hierfür Antennenarrays notwendig, um den Winkel eines ankommenden Signals zu bestimmen. Dies erhöht den Aufwand bei einer Hardwareumsetzung und letztlich auch die Kosten deutlich.

Eine weitere Technik basiert auf der Messung der Signalstärke (*Received Signal Strength*, vgl. Kapitel 2.3.2), bei der die Dämpfung eines Funksignals proportional zur Entfernung der Quelle ist. Ein Nachteil dieser Methode ist die relativ hohe Ungenauigkeit, da die Stärke des Signals abhängig von Umgebungsbedingungen ist. Die Signalstärke fällt zudem exponentiell, so dass schon nach kurzer Entfernung das Signal von Messrauschen überlagert wird und daher eine sinnvolle Entfernungsmessung nur auf kurze Distanz möglich ist. Vorteile sind jedoch der geringe Aufwand und die kaum vorhandenen Kosten: Praktisch alle verfügbaren Funkmodule für unterschiedlichste Technologien (WiFi, Bluetooth usw.) verfügen bereits mittels der bereitgestellten Programmierschnittstellen über Möglichkeiten, die Signalstärke zu messen. Zudem ist keine zusätzliche Hardware notwendig, so dass es genügt, ein geeignetes Funkmodul auszuwählen und mit einer minimalen äußeren Beschaltung zur Energieversorgung zu versehen, um dieses Messprinzip anzuwenden.

Für den weiteren Verlauf dieser Arbeit wird daher auf die Verwendung der Signalstärkemessung als Lokalisierungsprinzip gesetzt. Für das angestrebte Szenario sind keine absoluten Abstände zwischen den Utensilien notwendig und es wird angenommen, dass selbst eine ungenaue Positionsbestimmung genügt, um beispielsweise mit Methoden der künstlichen Intelligenz dennoch die zugrunde liegende Handlung zu erkennen. Diese Hypothese gilt es im Weiteren zu überprüfen und damit auch in wie weit eine solche kostengünstige Lokalisierungstechnik genügt, um Aktivitäten von Menschen zu analysieren.

6.2 Auswahl des Feedbackkanals

Um Personen, die mit dem System kochen werden, Hilfestellung geben zu können, bedarf es eines Rückkanals vom System zum Nutzenden. Dieser Rückkanal sollte gemäß den Anforderungen A18 bis A21 (vgl. Kapitel 5.2) so gestaltet sein, dass die Anwendenden zur Laufzeit Informationen über den korrekten nächsten Schritt erhalten, dabei nicht stört und die Person aktiviert. Eine Reihe von technischen Lösungen sind hierzu denkbar, diese sind in Tabelle 6.2 zusammengefasst und bewertet.

Technologie	Vorteile	Nachteile
Sprachassistenz	<ul style="list-style-type: none">• eindeutige Anweisungen• leicht verständlich	<ul style="list-style-type: none">• zusätzlicher Hardwarebedarf• Übersetzung der Sprachanweisungen in verschiedene Sprachen notwendig
Signalton	<ul style="list-style-type: none">• einfach umzusetzen• kostengünstig	<ul style="list-style-type: none">• potentiell störend• Lokalisierung durch Nutzenden nicht eindeutig• keine eindeutigen Anweisungen
Projektion	<ul style="list-style-type: none">• eindeutige Anweisungen• leicht verständlich	<ul style="list-style-type: none">• zusätzlicher Hardwarebedarf• hoher Installationsaufwand• hohe Kosten
Signallicht	<ul style="list-style-type: none">• unaufdringlich• kostengünstig• einfach umzusetzen	<ul style="list-style-type: none">• keine eindeutigen Anweisungen• möglicherweise nicht auffällig genug
Vibration	<ul style="list-style-type: none">• kostengünstig• einfach umzusetzen• auch akustisch wahrnehmbar	<ul style="list-style-type: none">• Lokalisierung durch Nutzenden nicht eindeutig• keine eindeutigen Anweisungen• hoher Energieverbrauch

Tabelle 6.2: Vergleich der verschiedenen Feedbackmöglichkeiten

Technologien, wie Sprachassistenten oder ein Projektor, der die nächsten Schritte auf die Arbeitsplatte projiziert, könnten einerseits sehr detaillierte Informationen über den Rezeptablauf oder die nächsten Schritte geben, benötigen andererseits aber zusätzliche Hardware wie Lautsprecher oder Projektoren. Zudem sind die Technologien teuer in der Anschaffung und bei Einsatz von Text oder Sprache ist

zusätzliche eine Übersetzung in die Sprache der Nutzenden notwendig. Da das Ziel des in dieser Arbeit entwickelten Systems auch die *Aktivierung* der Anwendenden ist, ist zudem fraglich, ob solche Systeme zu viel Arbeit abnehmen, so dass eher das Gegenteil eintritt und beispielsweise Senior*innen zu wenig gefordert werden. Insgesamt werden diese Technologien daher nicht weiter betrachtet.

Varianten, die den letzten Aspekt berücksichtigen sind *Signaltöne*, *Vibration* und *Signallichter*. Alle diese Varianten lassen sich einfach und kostengünstig in Sensorknoten integrieren und benötigen damit keinen zusätzlichen Installationsaufwand für den Betrieb. Die Ansteuerung von Vibrationsmotoren benötigt aber vergleichsweise viel Energie, was im Widerspruch zu einer möglich langen Akkulaufzeit steht, diese Option wird daher nicht weiter betrachtet. Signaltöne lassen sich über Piezoelemente sehr stromsparend erzeugen, allerdings ist ein solcher Ton nicht zweifelsfrei einem bestimmten Objekt zuzuordnen, wenn mehrere solcher Küchenutensilien während eines Kochvorgangs nahe beieinander liegen.

Übrig bleibt damit ein Feedbacksystem mittels Signallichtern, konkret in Form einer LED, die auf jedem Sensorknoten angebracht wird. Diese können platz- und kostensparend implementiert werden, über verschiedene Farb- und Blinkmuster können auch zusätzliche Informationen, z.B. zum Status des Systems, mitgeteilt werden. Es gilt zu untersuchen, ob LEDs im Betrieb auffällig genug sind und von den Proband*innen ausreichend gut wahrgenommen werden. Auch im Bezug auf ein möglichst inklusives Design stellt dies nicht die optimale Möglichkeit dar, falls Personen z.B. Beeinträchtigungen beim Sehen besitzen. Möglicherweise ist daher zukünftig eine Kombination verschiedener Technologien am erfolgsversprechendsten. Damit in Zukunft und über diese Arbeit hinaus weitere Optionen getestet werden können, wird eine Möglichkeit zum Anschluss anderer Feedbacktechnologien, wie Piezoelemente oder Vibrationsmotoren, auf den Sensorknoten vorgesehen. Im Rahmen der Arbeit werden einzig LEDs auf ihre Tauglichkeit hin untersucht werden.

6.3 Spezifizierung der Sensorknoten

Für die Entwicklung und den Betrieb eines eingebetteten Systems stellt die Auswahl des Mikrocontrollers eine zentrale Frage dar, deren Beantwortung stark mit dem späteren Einsatzzweck zusammenhängt. Unterschiedliche *System on Chip (SoC)* mit ihren Mikrocontrollern bieten eine Vielzahl von Konfigurationsmöglichkeiten an Speicher, Rechenleistung, Energiebedarf, Größe, verfügbarer Peripherie und viele weitere.

Für das sich in dieser Arbeit in Entwicklung befindliche System ist vor allem ein geringer Energieverbrauch und die Möglichkeit einer Lokalisierung, basierend auf einer geeigneten Funktechnologie, zwischen den Knoten entscheidend. Nebenbedingungen sind aber auch die Kosten für ein einzelnes Modul, deren reine Materialkosten sich größtenteils aus dem SoC ergeben, und die Langzeitunterstützung, sowohl durch den Hersteller selbst, als auch durch eine Community, die es ermöglicht gegebenenfalls Informationen für eine Problemlösung zu finden.

Tabelle 6.3 vergleicht SoCs verschiedener Hersteller nach ihrem Energiebedarf, den unterstützten Funktechnologien, Kosten und Größe. Sie dient im weiteren Verlauf als Entscheidungsgrundlage für die Auswahl eines Moduls.

Hersteller	SoC	Energiebedarf	Funktechnologien	Kosten	Größe
Nordic Semiconductor	nRF52840 [101]	4,6 mA	Bluetooth Low Energy, Bluetooth Mesh, Thread, Zigbee, 802.15.4, ANT	3,45€	7 * 7 mm aQFN73
Espressif	ESP32-C6 [39]	315 mA	Wi-Fi, Bluetooth Low Energy, 802.15.4, Zigbee, Thread	1,92€	5 * 5 mm QFN40
Infineon	PSoC4 BLE [56]	18,7 mA	Bluetooth Low Energy	4,33€	7 * 7 mm QFN56
Texas Instruments	CC2650 [135]	9,1 mA	Bluetooth Low Energy, ZigBee, 6LoWPAN	3,77€	7 * 7 mm VQFN48
Renesas Technologies	DA14580 [111]	3,7 mA	Bluetooth Low Energy	2,15€	6 * 6 mm QFN48
Microchip Technology	BTLC1000 [89]	4,5 mA	Bluetooth Low Energy	2,24€	4 * 4 mm QFN32

Tabelle 6.3: Vergleich der verschiedenen SoCs

Für diese Vorauswahl wurde als Energiebedarf der Strombedarf bei einer Funkkommunikation gemäß der jeweiligen Datenblätter herangezogen. Die Kosten beziehen sich jeweils auf die Preise eines einzelnen SoCs bei Abnahme von 1000 Stück. Alle Angaben sind im Kontext des Zeitpunkts der Abfrage zu verstehen, durch globale Ereignisse wie Lieferengpässe oder technologische Entwicklungen ist diese Auswahl ständig veränderlich. Zudem ist diese Auswahl nicht erschöpfend, es sind unzählige Varianten der selben Chips und viele weitere Modelle anderer Hersteller erhältlich. Diese Vorauswahl stützt sich daher auf Vorarbeiten von Afaneh [5] und Teel [134].

Vor diesem Hintergrund wird der Chip von Espressif *ESP32-C6* wegen seines sehr hohen Strombedarf nicht weiter betrachtet, da damit nur eine ungenügende Laufzeit der einzelnen Sensorknoten zu erwarten ist. Auch der Infineon *PSoC4 BLE* sticht durch seinen höheren Strombedarf heraus und wird ebenfalls nicht weiter betrachtet, genauso wie der Texas Instrument *CC2650*.

Der Chip *DA14580* von Renesas Technologies verbleibt nicht in der weiteren Auswahl, da der Controller bereits als *NRND* (Not Recommended for New Designs) markiert ist, d.h. das Produkt nähert sich dem Ende seines Lebenszyklus und sollte daher für neue Projekte nicht mehr eingesetzt werden.

Um letztlich eine Entscheidung zwischen dem *nRF52840* von Nordic Semiconductor und dem Microchip Technology *BTLC1000* zu treffen, die im Strombedarf nahezu identisch sind, muss die eingesetzte Funktechnologie betrachtet werden. Wie im Unterkapitel 6.1 dargestellt, sollen die Sensorknoten so entworfen werden, dass auch zukünftige Weiterentwicklungen möglich sind. Aus diesem Grund ist die Einschränkung auf eine einzelne Funktechnologie, wie es beim *BTLC1000* der Fall ist, nicht wünschenswert. Der *nRF52840* bietet hier mehr Freiheiten, um verschiedene Technologien und Netzwerktopologien zukünftig testen und implementieren zu können. Dieser Umstand rechtfertigt auch den höheren Preis und Platzbedarf. Es wurde daher entschieden, die weitere Entwicklung mit dem *nRF52840* durchzuführen. Der Chip verfügt über 256 KB Arbeitsspeicher, 1 MB Flashspeicher, eine mit maximal 64 MHz taktende Cortex-M4 CPU, einen integrierten 2,4 GHz Bluetooth 5.4 Transceiver mit Protokollunterstützung für Bluetooth Low Energy (BLE), Bluetooth Mesh, Thread und Zigbee [101].

6.4 Design und Implementierung der Sensorknoten

Nachdem im vorherigen Kapitel der Mikrocontroller ausgewählt wurden, folgt nun das eigentliche Design der Platinen, die Bestückung und Implementierung der grundlegenden Software, die zum Betrieb der Sensorknoten und des Netzwerks notwendig ist.

Zunächst werden die weiteren notwendigen Bauteile zum Betrieb ausgewählt, dies beinhaltet einen Spannungsregler, ein Batteriemanagementsystem, einen Beschleunigungssensor und weitere Komponenten zum energiesparenden Betrieb, wie Antennen oder Quarzoszillatoren.

Softwareseitig wird in diesem Kapitel der grundlegende Systemablauf implementiert, dies umfasst die Kommunikation zwischen den Knoten und die Messung der Signalstärke, die Anbindung des Beschleunigungssensors und Feedbackkanals an den Mikrocontroller und den Datenaustausch zu einer Recheneinheit in Form eines Computers.

6.4.1 Energieversorgung und Ladeschaltung

Für die Energieversorgung und Ladeschaltung bedarf es einer Spannungsregelung und einer Laderegelung, die geeignet ist, die vorgesehenen Lithium-Polymer (LiPo) Akkus sicher zu laden. LiPos liefern eine Nennspannung von 3,7 V und besitzen eine vergleichsweise hohe Energiedichte, dürfen aber beim Laden nur mit vorgegebenen Strömen bzw. Spannungen belastet werden. Typischerweise wird der zulässige Ladestrom als C-Rate (C) ausgedrückt, handelsübliche Akkus werden mit 0,5 C bis 1 C geladen. Die C-Rate entspricht dem Strom der Akkukapazität in Milliamperestunden (mAh), bei einem Akku mit 1000 mAh sind 0,5 C folglich 500 mA, 1 C analog 1000 mA.

Für die Sensorknoten wird als Batteriemanagementsystem (BMS) der *MCP73832T-2ACI/OT* von Microchip Technology ausgewählt [90], der einen einstellbaren Ladestrom von 15 mA bis 500 mA unterstützt, so dass auch sehr kleine Akkus geladen werden können. Der Chip selbst besitzt als Laderegler zusätzlich eine automatische Abschaltung und ist mit dem USB-Standard kompatibel, so dass Geräte sicher über einen USB-Anschluss geladen werden können. Für den Ladevorgang einer Akkuzelle benötigt der Chip selbst nur ca. 500 μA (typ.), ist der Ladevorgang beendet beansprucht der BMS nur 1 μA (typ.), dadurch eignet er sich gut für stromsparende Geräte wie die hier vorgestellten Sensorknoten.

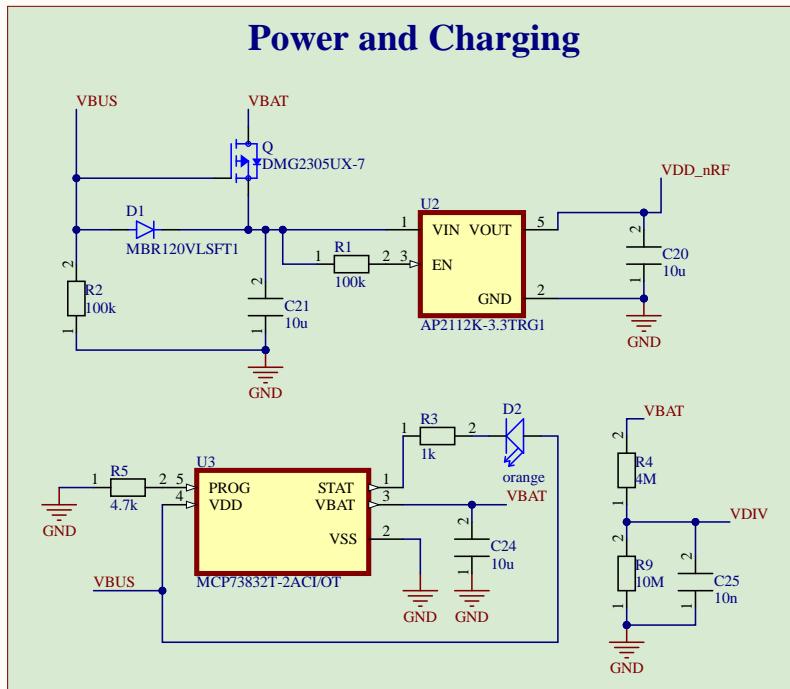


Abbildung 6.1: Schaltplan der Strom- und Laderegelung

Als Spannungsregler, der die Versorgungsspannung der Systemkomponenten auch bei wechselnder Eingangsspannung, wie bei einem sich entladenden Akku, konstant hält, wird der *AP2112K-3.3TRG1* von Diodes Incorporated ausgewählt [36]. Der Spannungsregler ist ein sogenannter *Low Dropout*-Regler (LDO), d.h. die Batteriespannung muss nur geringfügig größer sein, als die Ausgangsspannung des Reglers. Der Chip liefert konstant 3,3 V bei maximal 600 mA, die Dropoutspannung beträgt 250 mV, d.h. das System funktioniert bis zu einer Entladung des LiPo-Akkus auf 3,55 V. Der LDO benötigt als Versorgung nur $55 \mu A$ (typ.), im Standbymodus sogar nur $0,01 \mu A$, so dass auch diese Komponente gut geeignet ist für einen langanhaltenden Akkubetrieb.

Schaubild 6.1 zeigt die beiden Energieversorgungsbestandteile zusammen mit ergänzenden externen, passiven Bauelementen zum Betrieb und Konfiguration der Mikrochips.

6.4.2 Accelerometer und Feedbackkanal

Als Beschleunigungssensor, der die Handlungserkennung durch einen weiteren Informationskanal in Form von Beschleunigungsdaten unterstützen kann, wird der **BMA400** von Bosch Sensortec verwendet [17]. Für die Begründung zur Ergänzung eines Accelerometers sei an dieser Stelle auf das nachfolgende Unterkapitel 6.4.4 verwiesen.

Der BMA400 löst Beschleunigungen zwischen ± 2 g bis ± 16 g, je nach gewünschtem Modus. Im Energiesparmodus werden $3,5 \mu A$ bei 1,2 V bis 3,6 V benötigt, im Betrieb höchstens $14,5 \mu A$. Der Chip verfügt über eine Kommunikation über SPI oder I²C und zusätzlich über zwei Interrupt-Pins, die getriggert werden, wenn der Chip beschleunigt wird. Auf diese Weise ist es beispielsweise möglich, dass ein daran angeschlossener Mikrocontroller erst aufgeweckt wird, wenn das Gerät bewegt wird, wodurch sich der Energiebedarf des Gesamtsystems je nach Anwendung nochmals deutlich verringern lässt.

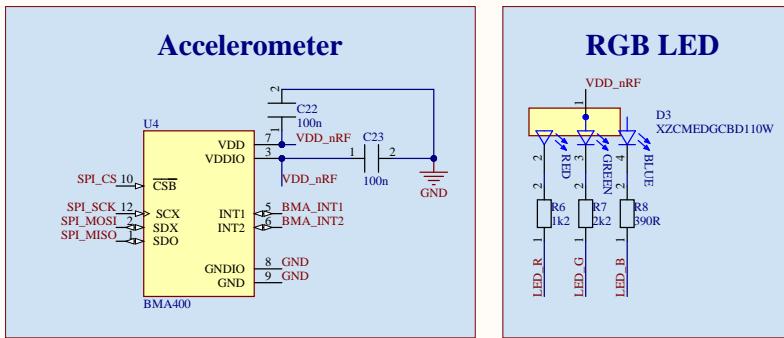


Abbildung 6.2: Schaltplan des Beschleunigungssensors und der RGB-LED

Als Leuchtmittel für das visuelle Feedback wurde eine RGB-LED von Kingbright ausgewählt [63]. Die LED emittiert bei einem Strom von 2 mA genug Licht, um im Arbeitsbereich deutlich sichtbar zu sein. Konkret bedeutet dies: 15 mcd (milicandela) rotes, 50 mcd grünes und 14 mcd blaues Licht. Durch den geringen Strombedarf bei 1,8 V (rot) bzw. 2,65 V (grün, blau) eignet sie sich gut

für energiesparende Anwendungen wie die hier entwickelten Sensorknoten, die Anbindung an die Gesamtschaltung ist aus Abbildung 6.2, rechts, zu entnehmen.

6.4.3 Antennen

Ein elementarer Bestandteil der Funkkommunikation ist die Wahl bzw. der Entwurf einer Antenne. Je nach Wahl des Designs kann die Performanz des Systems z.B. hinsichtlich Übertragungsreichweite und Energieeffizienz maßgeblich beeinflusst werden. Eine Antennenauslegung ist ein komplexer Prozess, daher ist es üblich, dass Hersteller von Funkkommunikationsmodulen ihre Produkte entweder bereits in Versionen mit Antenne ausliefern oder für Module ohne Antenne ein Referenzdesign mitzuliefern, welches nur noch geringfügig angepasst werden muss.

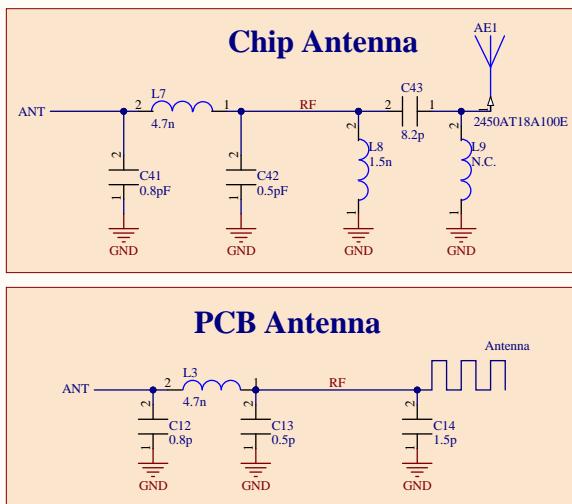


Abbildung 6.3: Schaltplan der Chipantenne (oben) und PCB-Antenne (unten)

Für das hier entwickelte Sensornetzwerk werden zwei Varianten eingesetzt. Eine PCB-Antenne (Abbildung 6.3, unten), die dem Referenzdesign für den eingesetzten nRF52840 SoC nachempfunden ist. Diese Antenne wird als Leiterbahn in Mäanderform direkt auf die Platine des Sensorknotens aufbracht, so dass keine zusätzlichen Materialkosten entstehen, der Platzbedarf erhöht sich jedoch um ca. 10 mm x 5 mm, da im Bereich der Antenne kein anderes Bauteil aufgebracht sein darf.

Eine weitere Variante wird mit einer Chip-Antenne realisiert, die mit den Maßen 1,6 mm x 0,8 mm x 0,4 mm sehr kompakt ist, dadurch aber einen Antennengewinn von typischerweise -2.0 dBi aufweist [57]. Zusätzlich wird eine äußere Beschaltung, wie in Abbildung 6.3, oben, als Anpassnetzwerk benötigt, welches vom Hersteller vorgegeben wird und gegebenenfalls optimiert werden muss, beispielsweise auf die Ausgangsimpedanz des Antennenanschlussspins des SoC.

6.4.4 Gesamtschaltung

Die Grundprämisse des hier entwickelten Sensornetzwerks ist die bestmögliche Kostenreduktion bei einer gleichzeitigen, ausreichend genauen Handlungsanalyse, die den übrigen Anforderungen hinsichtlich Genauigkeit und Echtzeitfähigkeit genügen kann. Da nicht ausgeschlossen ist, dass die eingesetzte Lokalisierungs-technik zu wenige Informationen für dieses Ziel liefert, werden im weiteren Verlauf zwei Varianten betrachtet: Eine minimale Variante (Smartpatch Lite), die nur über die absolut notwendige Beschaltung verfügt und eine reguläre Variante (Smartpatch), die noch zusätzlich um einen Beschleunigungssensor ergänzt wird. Dieses Vorgehen wurde gewählt, da die Entwicklung und Produktion von Hardware sehr aufwendig ist; um Verzögerungen während der Entwicklung des Systems zu vermeiden, wurde entsprechend zwei Varianten entworfen und umgesetzt. Um die Baugröße des Smartpatch Lite noch weiter zu minimieren verfügt dieser über das Chip-Antennendesign, während der etwas größere Smartpatch durch eine PCB-Antennen kommuniziert. Der Smartpatch Lite wird zudem mit passiven Bauelementen in der Größe 0201 bestückt (ca. 0,6 mm x 0,3 mm), die ein händisches Löten sehr erschweren. Der Smartpatch verwendet hingegen

Bauelemente der Größe 0402 (ca. 1,0 mm x 0,5 mm), ein manuelles Verändern der Bestückung ist dadurch bei entsprechenden Vorkenntnissen noch möglich.

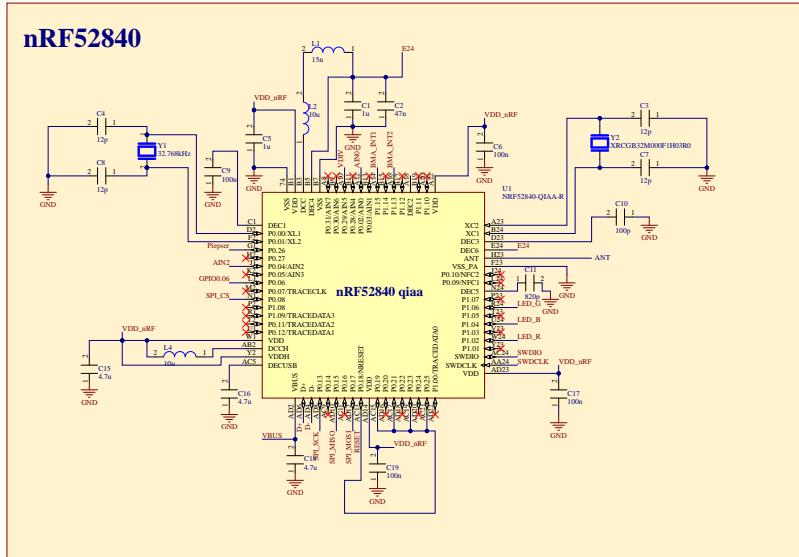


Abbildung 6.4: Beschaltung des Mikrocontrollers nRF52840

Abbildung 6.4 zeigt die Anbindung des Mikrocontrollers an die Peripherie. Zu bemerken ist an dieser Stelle auch die Verwendung des externen 32,768 kHz Quartz (Y1). Dies verringert den Strombedarf um $0,5 \mu A$ auf $I_{LFXO} = 0,2 \mu A$ gegenüber der Nutzung des internen Taktgebers ($I_{LFRC} = 0,7 \mu A$). Im Gegenzug bedarf es einer zusätzlichen Bestückung eines Taktgebers und zweier Kapazitäten (C4, C8) woraus ein zusätzlicher Platzbedarf und höhere Bauteilkosten resultieren.

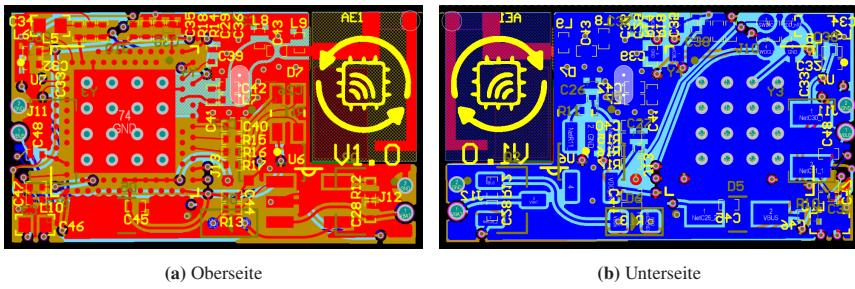


Abbildung 6.5: PCB-Design der minimalen Smartpatch Lite Variante

Abbildung 6.5 stellt die Leiterplatte des Smartpatch Lite dar, sowohl Oberseite als auch Unterseite, Abbildung 6.6 ergänzend dazu die beiden Seiten des Smartpatch PCB.

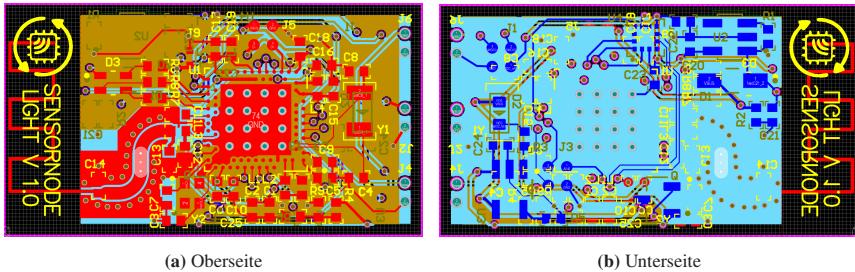


Abbildung 6.6: PCB-Design der regulären Smartpatch Variante

Beide Leiterplattenansichten sind zur besseren Lesbarkeit in der Größe aneinander angeglichen. Das Ergebnis der Sensorknotenentwicklung ist in den Abbildungen 6.7 (Vorderseite, Smartpatch oben, Smartpatch Lite unten) und 6.8 (Rückseite, ebenfalls Smartpatch oben, Smartpatch Lite unten) zu sehen, ebenfalls Smartpatch oben, Smartpatch Lite unten) zu sehen, mit einer maßstabsgetreuen 1-Eurocentmünze als Referenz.

Oben ist jeweils die vollausgestattete Smartpatch Platine mit PCB-Antenne und Accelerometer zu sehen, unten die reduzierte Smartpatch Lite Variante mit Keramikantenne und ohne Beschleunigungsmesser. Der Smartpatch misst 28 x 15,5 mm, die Lite Variante hingegen 20 x 11,5 mm.

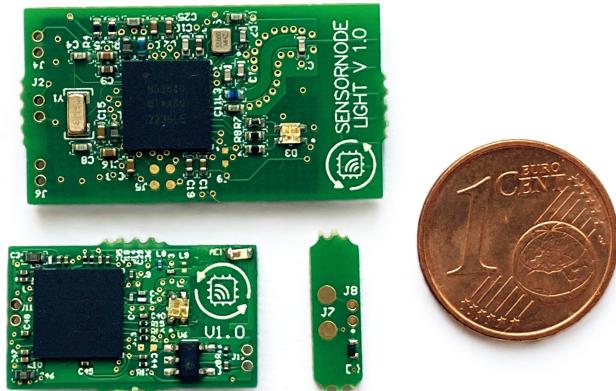


Abbildung 6.7: Vorderseite der beiden Smartpatch-Varianten

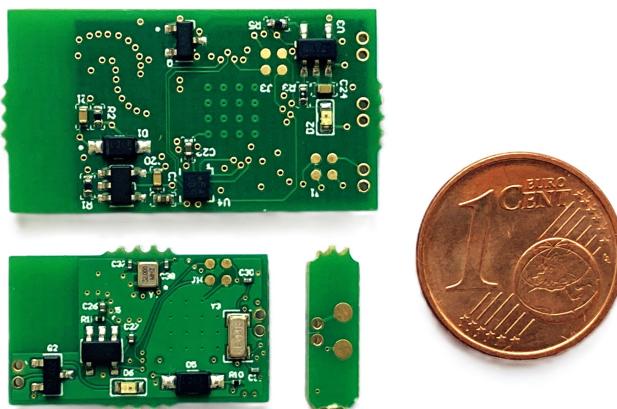


Abbildung 6.8: Rückseite der beiden Smartpatch-Varianten

Es folgt nun die Implementierung der Software zum Datenaustausch zwischen den Sensorknoten, zur Übertragung an eine zentrale Recheneinheit und zur Steuerung des Feedbackkanals.

6.4.5 Sensorknotensoftware

Dieser Abschnitt dokumentiert die Entwicklung der Sensorknotensoftware und beschreibt den Systemablauf bei der Datenaufnahme und Verarbeitung der Signalstärkemessungen, als auch die Implementierung der Feedbackfunktion.

Ziel ist es, die Steuerung der Sensorknoten so zu gestalten, dass einerseits eine hinreichend genaue Datenaufnahme möglich ist, andererseits aber auch die Batterielaufzeit maximiert wird.

Der Ablauf der Signalstärkemessung und Datenübertragung lässt sich wie in Abbildung 6.9 dargestellt abbilden und wird im Folgenden näher erläutert.

Ausgangspunkt ist eine Unterteilung des Systemzustands in *Scanning* und *Advertising*. Während dem Scanning überwacht jeder Knoten den Funkkanal und überprüft ihn auf neue Datenpakete. Wird während der Scanzeit kein Paket empfangen, wird der Scan neugestartet. Erreicht den Knoten hingegen ein Datenpaket, so wird zunächst überprüft, ob das Gerät bekannt ist und somit Teil des Netzwerks ist; dies geschieht anhand einer eindeutigen Geräteidentifikationsnummer (ID). Der nRF52840 verfügt über ein sogenanntes *Whitelisting*, das es ermöglicht, unbekannte Geräte noch vor einem weiteren Verbindungsauflaufbau abzuweisen, wodurch ein unnötiger Energiebedarf für einen Datenaustausch vermieden wird.

Ist das Gerät auf der Whitelist eingetragen und damit bekannt, wird im nächsten Schritt die Signalstärke des Datenpaketes und die Geräte-ID gespeichert. Für Sensorknoten, die an Küchenutensilien angebracht sind, ist somit das *Scanning* für diesen Zyklus abgeschlossen und es beginnt das *Advertising*. Sollte der Sensor allerdings das Gateway sein, d.h. der Knoten, welcher mit der Recheneinheit verbunden ist, so wird auch der Inhalt des Datenpaketes gespeichert. Jener besteht aus den Signalstärkemessungen des übertragenden Sensors zu allen anderen Sensorknoten.

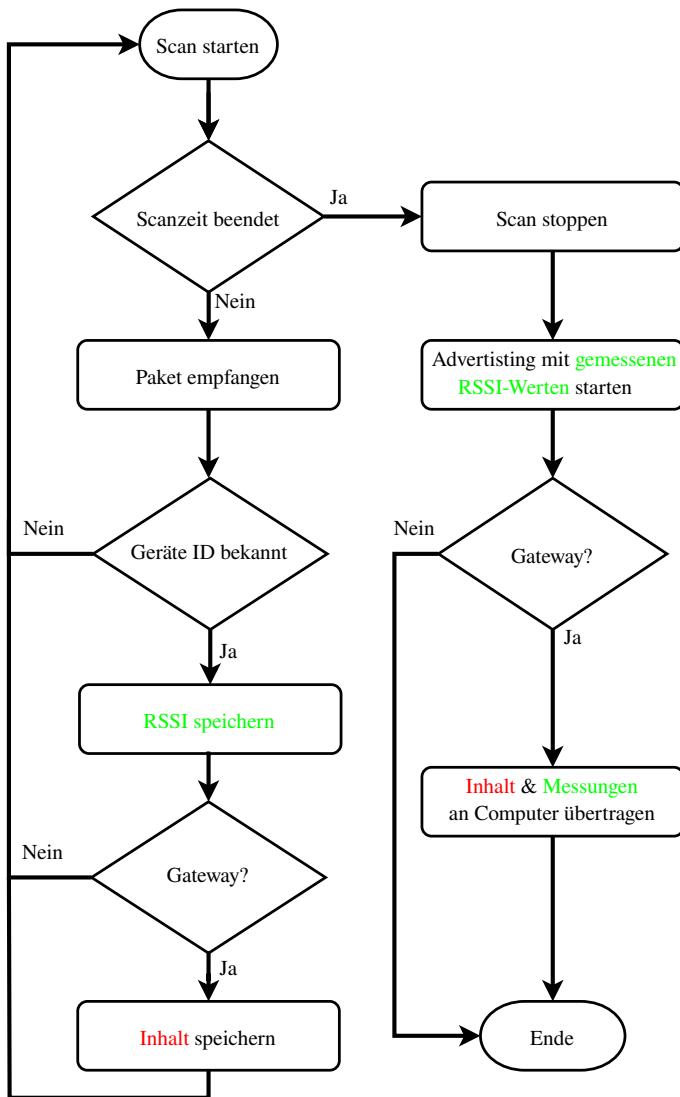


Abbildung 6.9: Ablaufdiagramm der Sensorknotensoftware

Beim Advertising werden alle Messungen eines jeden Knotens als Datenpakete ohne Verbindungsziel versendet, andere Knoten können diese Pakete empfangen und entsprechend wieder eine neue Signalstärkemessung durchführen. Das Gateway selbst überträgt die gesammelten Messungen an einen Computer. Auf diese Weise entsteht eine Sterntopologie aus einem Gateway und mehreren intelligenten Küchenutensilien.

Beim Akkumulieren der Messungen auf dem Gateway entsteht eine Matrix, die demnach mit beispielsweise 0,5 Hz (vorgegeben durch die Scanning- und Advertisingzeiträume) erzeugt wird, welche die Signalstärkemessungen zwischen allen Knoten untereinander enthält. Jede Zeile steht dabei für einen Sensor an einem Küchenutensil, der Wert in der zugehörigen Spalte ist dann der RSSI-Messwert zu einem anderen Sensor an einem weiteren Küchenutensil. Die Diagonale wird folglich auf Null gesetzt.

Die Aktivierung der LED eines Sensorknotens findet ebenfalls über ein Advertising und der eindeutigen Geräte-ID statt. Wird in der zentralen Recheneinheit der nächste Kochschritt inferiert, kann über eine Zuordnung der Schritte zu den Kochutensilien die ID zugeordnet werden. Die Recheneinheit überträgt die Information an den Gatewayknoten, welcher wiederum ein Advertisementspaket aussendet, welches die Geräte-ID derjenigen Knoten enthält, die aufleuchten und IDs der Knoten, die ihre LEDs ausschalten sollen. Die LED bleibt dann so lange an, bis der nächste Schritt inferiert ist und folglich neue Anweisungen zum Zustand der LED eintreffen. Abbildung 6.10 illustriert diesen Vorgang für das Anschalten der LED des Sensorknotens mit der ID 2.

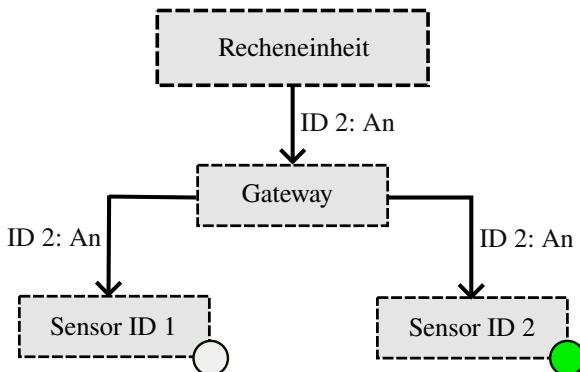


Abbildung 6.10: Befehlsweitergabe für Feedback

Nachfolgend werden die Sensorknoten ersten Tests unterzogen, ihr Energiebedarf gemessen und somit ihre Tauglichkeit für die weitere Verwendung gemäß den Anforderungen überprüft.

6.5 Vorabtestung der Sensorknoten

Die Vorabtestung der Sensorknoten soll den korrekten Betrieb der Sensorknoten sicherstellen. Dies umfasst sowohl die Kommunikation zwischen den Knoten und der zentralen Recheneinheit, RSSI-Messung und Akkumulation der Messung im Gatewayknoten, Ausgaben über den Feedbackkanal als auch eine Energieverbrauchsmessung zur Bestimmung der zu erwartenden Laufzeit.

Die prinzipielle Funktionsweise kann durch eine Übertragung über den Sensorknoten an die Recheneinheit bestätigt werden. Hierzu wurden alle Knoten verwendet und der Gatewayknoten mit dem PC per Kabel verbunden, der seinerseits wiederum eine Verbindung über UART mit dem Gateway aufbaute. Die empfangenen Daten wurden anschließend lokal gespeichert.

Tabelle 6.4 zeigt beispielhaft eine solche Wertematrix aus einer Testaufzeichnung zu einem einzelnen Zeitpunkt. In diesem Beispiel ist **Zeile 3, Quell-ID 2** der Pfannensensor, der eine Signalstärke von -88 dBm zum Sensor am Messer, **Spalte 4**,

Ziel-ID 3, gemessen hat. Alle Sensorknoten konnten erfolgreich die Signalstärke messen und versenden, der Gatewayknoten alle Daten sammeln und als Matrix an den PC übertragen. Es zeigt sich, dass die reziproken Messungen zwischen zwei Knoten üblicherweise nicht den selben RSSI-Wert ergeben, dies kann beispielsweise auf leicht unterschiedliche Antennenabstimmungen zurückzuführen sein, oder auf Multipfadausbreitungen, so dass an den jeweiligen Orten der Sensoren unterschiedliche Interferenzen auftreten.

		Ziel-ID								
		0	1	2	3	4	5	6	7	8
Quell-ID	0:	0	-88	-91	-90	-78	-88	-84	-79	-89
	1:	-90	0	-72	-88	-94	-86	-91	-97	-81
	2:	-83	-79	0	-88	-93	-86	-89	-87	-90
	3:	-90	-89	-92	0	-71	-71	-67	-68	-80
	4:	-77	-86	-92	-68	0	-67	-76	-68	-74
	5:	-88	-81	-88	-70	-67	0	-57	-72	-80
	6:	-79	-95	-87	-64	-96	-58	0	-71	-92
	7:	-80	-95	-92	-66	-67	-72	-71	0	-80
	8:	-86	-80	-84	-79	-76	-80	-85	-80	0

Tabelle 6.4: Beispiel einer RSSI Messungen, alle Werte in dBm

Mit einer weiteren Testung mittels RSSI-Messung zwischen zwei Sensorknoten kann zusätzlich auch die erwartete Entfernungsabhängigkeit der Signalstärke bestätigt werden, vergleiche hierzu Abbildung 6.11.

Die Grafik zeigt sieben verschiedene, in unterschiedlichen Farben markierte, RSSI-Messungen. Ein Sensorknoten blieb während der Messung stationär, der Abstand des anderen Knotens wurde in eine Raumrichtung von drei bis 60 Zentimetern in Dreizentimeterschritten variiert. Die Messungen wurden an verschiedenen Orten in der Küche wiederholt. Es zeigt sich eindeutig eine Entfernungsabhängigkeit, wobei die Empfangsstärke schon nach wenigen Zentimetern abnimmt und

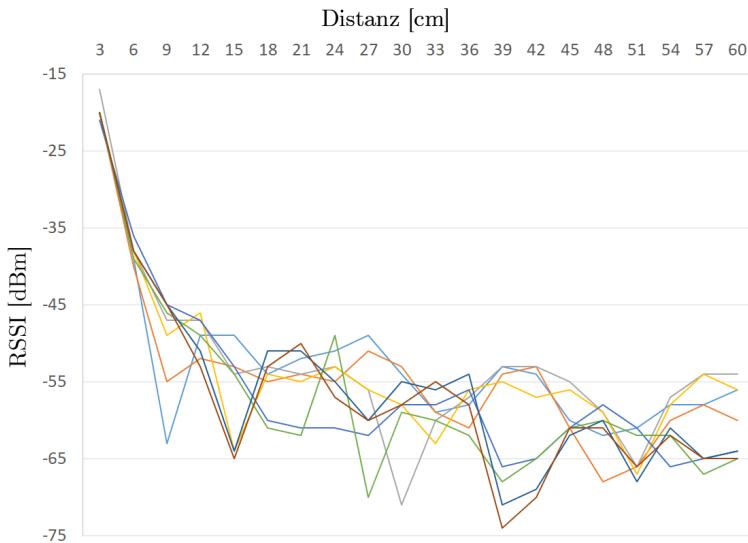


Abbildung 6.11: RSSI-Messung zwischen zwei Sensorknoten an verschiedenen Orten

bereits nach ungefähr 33 cm abflacht; eine exponentielle Dämpfung ist erkennbar (vgl. das log-distance path loss model, Kapitel 2.3.2). Zudem ist zu erkennen, dass die Messung stark rauschbehaftet ist und sich große Änderungen der Signalstärke durch die Umgebung ergeben. Mit dieser Ausgangslage gilt es im weiteren Verlauf zu überprüfen, in wie weit die Hypothese, dass alle notwendigen Informationen für eine Rekonstruktion eines Kochvorgangs durch Machine-Learning-Verfahren enthalten sind, zutreffend ist.

Zum Test des Feedbackkanals wurde von der Recheneinheit der Befehl an die Sensorknoten gesendet, der eine Aktivierung bzw. Deaktivierung der LEDs auslöst. Auf diese Weise konnte bestätigt werden, dass alle LEDs funktionieren und plangemäß an- und ausgeschaltet werden können.

Die technische Funktionalität des Sensornetzwerks kann somit bestätigt werden. Es gilt den Stromverbrauch eines Sensorknotens zu messen, um so auf die zu erwartende Laufzeit ohne Nachladen schließen zu können. Hierzu wird das *Power Profile Kit II* von Nordic Semiconductor eingesetzt [102].

Der Plot in Abbildung 6.12 zeigt in grau die gemessene momentane Leistung des Smartpatch im Verlauf von zehn Sekunden, während in rot die durchschnittliche Leistung über diesen Zeitraum eingezeichnet ist.

Im Graph sind Energieverbrauchsspitzen von bis zu 45 mW erkennbar, diese treten jede Sekunde auf und fallen mit dem eingestellten Scanning-Intervall zusammen. Weiterhin sind Spitzen von 30 mW sichtbar, die ihrerseits durch das konfigurierte Advertising-Intervall von 500 Millisekunden verursacht werden. Im Schnitt erreichte das System eine Leistungsaufnahme von 2,8 mW über 10 Sekunden.

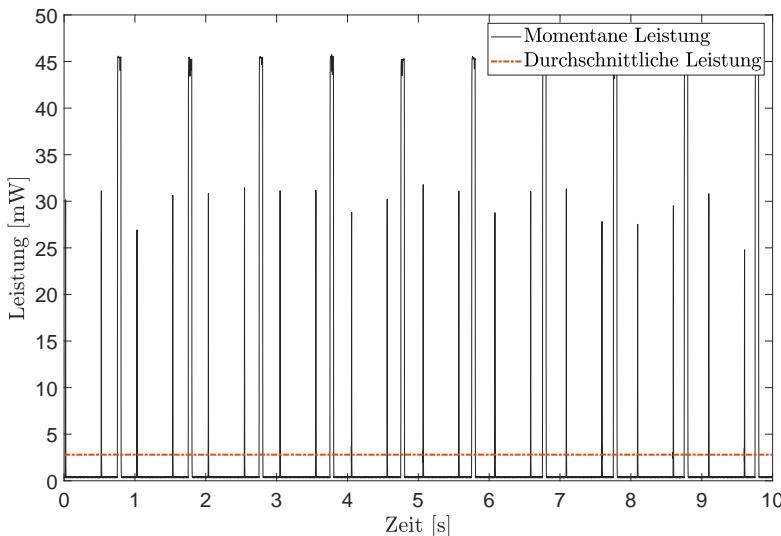


Abbildung 6.12: Energieverbrauchsmessung des Smartpatch

Ein 160 mAh Lithium-Polymer-Akku entspricht in seinen Dimensionen ungefähr der Größe des Smartpatch und ist somit gut für die Energieversorgung geeignet. Mit einer Akku-Nennspannung von 3,7 V wird damit eine Akkukapazität von ca. 592 mWh erreicht. Bei einer durchschnittlichen Leistungsaufnahme eines Sensorknotens von 2,8 mW kann somit bereits eine Laufzeit von 211 Stunden bzw.

knapp neun Tagen erreicht werden. Die Anforderungen an die Laufzeit (vgl. Kapitel 5.2) können damit bereits erreicht werden. Es besteht jedoch noch weiteres Potential zur Verlängerung der Laufzeit, da typischerweise eine Unterstützung beim Kochen nur zu bestimmten Zeiten benötigt wird, zu anderen Zeiten könnte das System beispielsweise in einen Ruhezustand versetzt werden. Auch eine Optimierung der Scanning- und Advertising-Intervalle kann zukünftig für eine Verlängerung der Systemlaufzeit in Betracht gezogen werden.

7 Entwicklung der Machine Learning Komponenten

Im vorangegangenen Kapitel 6 konnte gezeigt werden, dass die Funktionalität der Sensorknoten bei einem geringen Energiebedarf gegeben ist. Das Feedbacksystem wird zentral von der Recheneinheit bedient, indem die einzelnen LEDs der jeweiligen Sensorknoten geschaltet werden. Die Knoten können sowohl untereinander die Signalstärke in Form des RSSI messen, als auch diese Informationen an das Gateway weiterleiten, welches die Daten an die zentrale Recheneinheit überträgt. Diese RSSI-Messdaten gilt es im weiteren Verlauf zu analysieren, um daraus sowohl das gekochte Rezept, als auch einzelne Teilschritte identifizieren zu können, so dass letztlich eine kochende Person Unterstützung erhalten kann.

In diesem Kapitel wird der Entwicklungsprozess der Machine Learning Komponenten nachgezeichnet. Da im System an mehreren Stellen ML-Methoden eingesetzt werden, beginnt das Kapitel mit einer Übersicht der eingesetzten Subsysteme (7.1), im Anschluss erfolgt die Erläuterung der Konzeptionierungs- und Implementierungsarbeit, gefolgt von einem technischen Prätest. Die Unterkapitel zu Konzeptionierung, Implementierung und Test liegen dabei jeweils für das Subsystem *Hintergrundsystem* (7.2, 7.3 & 7.4) und *eingebettetes System* (7.6, 7.7 & 7.8) vor. Unterkapitel 7.5 begründet den Einsatz des eingebetteten ML-Systems aus Beobachtungen im Prätest des Hintergrundsystems.

7.1 Einsatzgebiete von Machine Learning Algorithmen im System

Diese Arbeit basiert auf der Hypothese, dass die Information über den aktuellen Kochprozesszustand im Abstand der Utensilien, zumindest teilweise, enthalten ist. Diese Entfernung soll möglichst kostengünstig gemessen werden, was zu der Frage führt, ob jene Information auch noch in einer rauschbehafteten Messgröße wie einer RSSI-Messung ausreichend deutlich enthalten ist, so dass mittels Machine-Learning Methoden diese Information extrahiert werden können.

Als Vorgriff auf die weiteren Ergebnisse wird an dieser Stelle bereits aus Gründen der Nachvollziehbarkeit und Übersichtlichkeit die Begriffe *Hintergrundsystem* und *eingebettetes ML-System* eingeführt und ihre Notwendigkeit kurz erläutert.

Die primäre Einbettung der Machine Learning Methoden findet im *Hintergrundsystem* statt, dies ist das System, welches in der zentralen Recheneinheit aus den RSSI-Messmatrizen das aktuell gekochte Rezept und den einzelnen Teilschritt inferiert. In ersten Tests zeigten sich gute Ergebnisse, als Ergänzung für Grenzfälle wurde zusätzlich ein eingebettetes KI-System entwickelt. Dieses System inferiert auf Basis von Beschleunigungsdaten auf dem Mikroprozessor des betreffenden Sensorknotens, das Ergebnis der Inferenz wird Teil des Datenpakets zusammen mit den RSSI-Messdaten. Eine nähere Begründung für dieses Vorgehensweise findet sich in Unterkapitel 7.5, es folgt nun unmittelbar die Beschreibung der Entwicklungen für das Hintergrundsystem.

7.2 Konzeptionierung des Machine Learning Hintergrundsystems

Nach aktuellem Stand werden für Aufgaben, wie es die Prozesserkennung am Beispiel des Kochens darstellt, häufig Ende-zu-Ende Architekturen eingesetzt. Diese Art von Architektur verringert den Aufwand beim Design und der Implementierung solcher Netze, da die Rohdaten direkt verarbeitet werden und es somit möglich ist, das KI-Modell ohne Umwege zu trainieren. Der Nachteil dieser Methode

ist jedoch, dass für eine robuste Inferenz sehr große Mengen an Trainingsdaten benötigt werden, damit alle relevanten Merkmale in den Daten enthalten sind. Zusätzlich erhöht sich die Komplexität der Fehlersuche und -behebung, da keine expliziten Zwischenschritte aus- oder eingeschlossen werden können.

Aus diesem Grund wird im Rahmen dieser Arbeit eine modulare Architektur erarbeitet und implementiert, die es ermöglicht, für einzelne Teilaufgaben angepasste KI-Modelle zu trainieren, um diese bezüglich ihrer Performanz zu optimieren. Die zu implementierenden Teilfunktionalitäten sind in Tabelle 7.1 dargestellt. Es gilt zunächst die bestmöglichen Machine Learning Algorithmen für jede Unteraufgabe zu finden und anschließend die einzelnen Modelle zu verknüpfen, so dass aus den übertragenen RSSI-Messungen die entsprechenden Inferenzen durchgeführt und Vorhersagen getroffen werden können.

Aufgabe	Modulname	Funktion
Teilschritterkennung	SubstepClassifier	Erkennung des aktuellen Teilschritts im Kochprozess, beispielsweise <i>Salzen</i> oder <i>Schneiden</i>
Teilschrittvorhersage	SubstepInferrer	Vorhersage des nächsten Teilschritts im Kochprozess, notwendig zur Unterstützung durch das Feedbacksystem
Prozesserkennung	ProcessClassifier	Erkennung des Gesamtprozesses, d.h. des gekochten Rezepts

Tabelle 7.1: Teilfunktionalitäten und ihre Bedeutung für die Gesamtarchitektur

Der Begriff *Teilschritte* muss zunächst weiter definiert werden, da das Wort selbst keine Aussagen über die Granularität der damit assoziierten Handlungen zulässt. Es ist zwar potentiell denkbar, auch sehr kurze Teilespekte als *Teilschritt* zu definieren, allerdings erhöht dies die benötigte Varianz an Trainingsdaten erheblich. Ebenso bedarf es einer größeren Menge an Trainingsdaten, wenn viele unterschiedliche, abstraktere Teilschritte erkannt werden sollen. Aus diesem Grund werden im Rahmen dieser Arbeit die in Tabelle 7.2 dargelegten Teilschritte als

solche definiert.

Ziel ist die Balance zwischen einer ausreichenden Granularität, die eine möglichst frühzeitige Unterscheidung des Gesamtprozesses anhand der Teilschritte zulässt, bei einer Minimierung der Teilschrittanzahl, damit nur eine handhabbare Menge an Trainingsdaten erzeugt werden muss. Die Spalte *Enkodierung* in der Tabelle zeigt die interne Repräsentation der Teilschritte und wird später ebenfalls für die Darstellung der Testergebnisse genutzt.

Teilschritt	Enkodierung
untätig	0
salzen	1
Käse reiben	2
Kartoffeln reiben	3
Pfannenwender benutzen	4
Topf auf den Herd stellen	5
schöpfen mit Löffel	6
umrühren mit Löffel	7
Pfanne auf den Herd stellen	8
Zwiebeln schneiden	9
Pilze schneiden	10
Kartoffeln schneiden	11

Tabelle 7.2: Definition der Teilschritte und ihre Enkodierung

Diese zwölf verschiedenen Teilschritte werden so gewählt, dass sie einerseits zwar eine gewisse Allgemeingültigkeit besitzen, andererseits aber auch zu den drei in der Evaluation angewandten Rezepten *Gedünstete Karotten*, *Gemüsesuppe* und *Nudeln mit Tomaten* passen. Diese Rezepte sind nicht zu umfangreich, um mit möglichst vielen Proband*innen in einem begrenzten Zeitrahmen Tests durchführen zu können, außerdem benötigen die Versuchspersonen keine besonderen

Fähigkeiten, um diese kochen zu können. Eine nähere Erläuterung zur Auswahl der Rezepte und eine genaue Beschreibung findet sich in Kapitel 8.1.

Offensichtlich ist diese Bibliothek an Teilschritten nicht abschließend und muss für andere Rezepte erweitert werden, sie stellt aber eine Grundlage dar, um einfache Basisrezepte durch das System erkennen zu können.

7.3 Implementierung des Hintergrundsystems

Zur Implementierung der Architektur müssen verschiedene Komponenten entwickelt werden. Im ersten Schritt wird Python als Programmiersprache ausgewählt, da diese eine hohe Lesbarkeit besitzt und von dem Anwendungsbeispiel keine speziellen Anforderungen an die Laufzeit oder Sicherheit gestellt werden. Zusätzlich sind für Python verschiedene Machine Learning Module verfügbar, die breiten Einsatz im wissenschaftlichen Umfeld finden und dadurch prädestiniert sind für die weitere Verwendung in dieser Arbeit, da viele Ressourcen zur Verfügung stehen, um auftretende Herausforderungen zu lösen. Konkret werden daher im weiteren Verlauf die scikit-learn¹ als auch die TensorFlow-Module² genutzt, um die Architekturen der Modelle zu implementieren.

Abbildung 7.1 zeigt den Datenfluss, ausgehend von den Rohdaten aus dem Sensornetzwerk, über die Teilschrittklassifizierung im Modul *SubstepClassifier*, welches als Ausgabe den aktuellen Kochschritt inferiert. Diese abstrahierte Information bildet die Grundlage für die Teilschrittvorhersage im Modul *SubstepInerrer*, welches den nächsten Teilschritt prädiziert, und das Prozesserkennungsmodul *ProcessClassifier*, welches letztlich das aktuell Rezept inferiert.

Tabelle 7.3 zeigt eine Vorauswahl an Algorithmen für die einzelnen Module. Diese Auswahl basiert auf Eigenschaften der Datensätze und auf dem aktuellen Stand

¹ <https://scikit-learn.org/stable/> abgerufen am 14.10.2024

² <https://www.tensorflow.org/> abgerufen am 14.10.2024

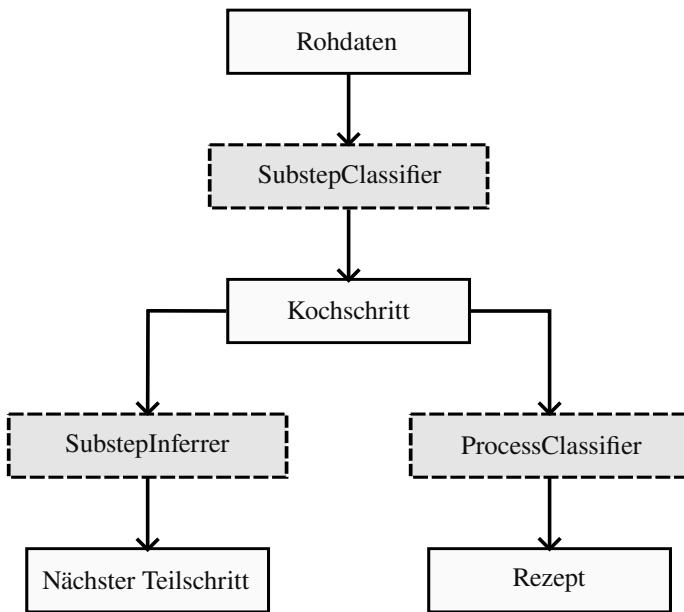


Abbildung 7.1: Datenfluss zwischen den KI-Modulen

der Forschung. Die hauptsächliche Herausforderung liegt in der begrenzten Datenmenge, die im Rahmen dieser Arbeit eigenständig erzeugt wird. Unter diesen Umständen führen klassische Machine Learning Modelle, wie Support Vector Maschinen (SVM), zu besseren Ergebnissen [62].

Neben einer SVM-Architektur soll für jedes Modul dennoch auch eine Architektur basierend auf einem neuronalen Netz untersucht werden, wie in Tabelle 7.3 dargestellt. Für zeitabhängige Eingangdaten, wie es die Zeitreihen der RSSI-Messungen darstellen, kommen hierfür rekurrente neuronale Netze (RNN) in Frage. Besonders Long Short-Term Memory (LSTM) Modelle, die durch ihre Architektur zeitliche Zusammenhänge kodieren können, sind hierfür interessant. Ein Sonderfall stellt das *SubstepClassifier*-Modul dar, da die zugrunde liegenden Daten nicht nur als zeitvariantes Problem postuliert werden können, sondern auch

Modul	Algorithmen
SubstepClassifier (zeitinvariant)	SVM, MLP
SubstepClassifier (zeitvariant)	SVM + Buffer, LSTM
SubstepInferrer (zeitvariant)	SVM + Buffer, LSTM
ProcessClassifier (zeitvariant)	SVM + Buffer, LSTM

Tabelle 7.3: Mogliche Algorithmen fur die Module

als zeitinvariantes: Es ist denkbar, dass es zur Klassifikation des aktuellen Teilschritts keine Abfolge von Interaktionen benotigt, sondern eine Art Schnappschuss des aktuellen Zustands, d.h. dass die Anordnung der momentanen Utensilien im Raum genugt. Diese Moglichkeit wird im weiteren Verlauf ebenfalls untersucht.

7.4 Erster technischer Pretest

Fur den ersten technischen Pretest wurden Daten von elf Personen im Alter von 22 bis 36 Jahren aufgezeichnet und ausgewertet. Durch diese Datenaufzeichnung entstand ein Datensatz mit 23 Sets, wobei jedes Set die sieben Testrezepte enthalt. Der Datensatz besteht aus insgesamt 33 Stunden Messdaten, oder durchschnittlich 12 Minuten je Rezept. Von den 23 Sets wurden wie zur Testung von ML-Algorithmen ublich ca. 25 % der Daten fur die Validierung zurckgehalten. Dies bedeutet, dass 17 Sets zum Training der Algorithmen verwendet wurden und sechs zur Validierung. Wie bereits im vorherigen Abschnitt erwahnt, ist dies in Aufwand und Relation gesehen eine umfangreiche Datenaufzeichnung, fur tiefe neuronale Netze aber noch immer eine geringe Datenmenge.

Die Tabelle 7.4 zeigt die sieben Rezepte, die fur den Pretest ausgewahlt werden. In der Tabelle ist neben dem Namen abermals ihre Kodierung dokumentiert.

Testrezept	Enkodierung
Käseomelette	0
Kartoffelrösti	1
gekochte Kartoffeln	2
Nudeln kochen	3
gebratene Pilze	4
gebratene Paprika mit Zwiebeln	5
Bratkartoffeln	6

Tabelle 7.4: Definition der Testrezepte und ihre Enkodierung

Diese Rezepte basieren ebenfalls auf den in Tabelle 7.2 festgehaltenen Teilschritte, sind aber im Vergleich zu den Rezepten für die Evaluation mit einer anderen Strategie gewählt. Tabelle 7.5 zeigt die Vorgehensweise.

Rezepte	Gleiche Utensilien?	Gleiche Reihenfolge?	Gleicher Zeitbedarf?
Käseomelette, Kartoffelrösti	Ja	Nein	Nein
Nudeln kochen, gekochte Kartoffeln	Ja	Ja	Nein
gebratene Pilze, gebratene Paprika mit Zwiebeln, Bratkartoffeln	Ja	Ja	Ja

Tabelle 7.5: Kriterien für die Auswahl der Prätestrezepte

Für den Prätest ist die Testung der technischen Funktionalität von zentraler Bedeutung. Es sind daher sieben verschiedene Rezepte gewählt worden, die in ihrer Ähnlichkeit variieren. Für *Käseomelette* und *Kartoffelrösti* werden, wie bei allen

anderen Rezepten, die gleichen Utensilien verwendet, die Reihenfolge der Benutzung der Gegenstände und die Kochzeiten sind aber unterschiedlich. Es ist zu erwarten, dass dies den einfachste Fall für die KI-Analyse darstellt, da sich durch die unterschiedlichen Reihenfolgen und Zeiten viele Merkmale zwischen den Kochvorgängen unterscheiden. Für *Nudeln kochen* und *gekochte Kartoffeln* sind die Kochzeiten zwar immer noch unterschiedlich, es werden aber die Küchenutensilien in der selben Reihenfolge verwendet. Dies erhöht die Analyseschwierigkeit. Der schwierigste Fall ergibt sich für *gebratene Pilze*, *gebratene Paprika mit Zwiebeln* und *Bratkartoffeln*, da hier die Kategorien *genutzte Küchenutensilien Reihenfolge der Utensilienverwendung* und *Zeitbedarf* nahezu identisch sind. Die Rezepte wurden den Testpersonen ohne spezielle Instruktionen gegeben, um die Varianz der Daten zu erhöhen. Die Testrezepte finden sich zur Dokumentation in Anhang A.1.

Da die aufgenommen Datenmengen je Rezept in den Datensets nahezu ausbalanciert sind, wird die *Genauigkeit* (engl. *Accuracy*) als Performanzkriterium herangezogen. Zudem wird die Genauigkeit häufig in anderen wissenschaftlichen Arbeiten verwendet, so dass mittels diesem Kriterium eine gute Vergleichbarkeit gegeben ist.

SubstepClassifier Algorithmus	Genauigkeit
SVM	0,81
SVM + Buffer	0,55
MLP	0,15
LSTM	0,13

Tabelle 7.6: Ergebnisse für die möglichen Algorithmen für das SubstepClassifier Modul

Tabelle 7.6 zeigt die Ergebnisse der verschiedenen Algorithmen für das Substep-Classifier Modul. Es zeigt sich deutlich, dass die Modelle, die auf neuronalen Netzen beruhen, wie MLP und LSTM, mit 15 % bzw. 13 % deutlich schlechter

klassifizieren. Bei 12 möglichen, gleichverteilten Teilschritten sind sie nur wenig besser als eine zufällige Auswahl der Klasse, die statistisch in 8,3 % der Fälle auftritt.

Die SVM ohne Buffer, der eine zeitunabhängige Betrachtung der Daten zugrunde liegt, erbringt ein um ungefähr 25 Prozentpunkte besseres Ergebnis (81 %), als die Variante mit Buffer, basierend auf der zeitabhängigen Betrachtung (55 %).

SubstepInferrer Algorithmus	Genauigkeit
SVM + Buffer	0,81
LSTM	0,74

Tabelle 7.7: Ergebnisse für die möglichen Algorithmen für das SubstepInferrer Modul

Für das SubstepInferrer Modul sind die Testergebnisse in Tabelle 7.7 dokumentiert. Auch hier liefert das SVM basierte Modell, in diesem Fall mit Buffer, mit einer Genauigkeit von 81 % ein besseres Ergebnis als das LSTM basierte Modell mit 74 %.

ProcessClassifier Algorithmus	Genauigkeit
SVM + Buffer	0,49
LSTM	0,51

Tabelle 7.8: Ergebnisse für die möglichen Algorithmen für das ProcessClassifier Modul

Letztlich wird die Gesamtprozessklassifikation mittels dem ProcessClassifier Modul bewertet, in Tabelle 7.8 finden sich die dazugehörigen Ergebnisse. Der LSTM-Algorithmus schneidet hier mit 51 % minimal besser als der SVM basierte Algorithmus mit 49 % ab. Beide Werte erscheinen im ersten Moment ungenügend, nachfolgende Erklärung bietet jedoch eine Einordnung.

Das ProcessClassifier Modul inferiert auf Basis der Ausgangsdaten des SubstepClassifier Moduls mit jedem Teilschritt den Gesamtprozess. Gleichen sich zwei Rezepte zu Beginn aus Sicht der Sensordaten, beispielsweise in dem bei zwei Rezepten zuerst die Pfanne auf den Herd gestellt und im Anschluss mit dem Messer eine Zutat geschnitten wird, kann in diesen beiden ersten Teilschritten keine sichere Klassifikation des Gesamtprozesses vorgenommen werden. In einem solchen Zustand ist das Ergebnis des ProcessClassifier Moduls also Zufall. Dieser Umstand ergibt sich aus dem Wunsch, die Teilschritte aber auch den Gesamtprozess zur Laufzeit zu erkennen, bei einer Klassifikation nach Ende des Kochvorgangs trüte dieses Problem nicht auf.

Für die Berechnung des Genauigkeit-Kriteriums fließen diese zufälligen Inferenzergebnisse jedoch mit ein, somit sinkt der Wert erheblich. Vor diesem Hintergrund ist also eine Genauigkeit von ca. 50 % ein Zeichen für die Funktionsfähigkeit gegenüber einer reinen Zufallsauswahl, die bei sieben Rezepten ein Ergebnis von ca. 14 % erwarten ließen.

Modul	Algorithmus	Genauigkeit
SubstepClassifier	SVM	0,81
ProcessClassifier	LSTM	0,51
SubstepInferrer	SVM + Buffer	0,81

Tabelle 7.9: Ausgewählte Algorithmen für die einzelnen Module

Zusammengefasst ergibt sich die in Tabelle 7.9 aufgelistete Auswahl für die einzelnen Module: Eine Support Vector Maschine für das SubstepClassifier Modul, ein rekurrentes Netz in Form eines Long Short-Term Memory Modells für das ProcessClassifier Modul und eine Support Vector Maschine mit Buffer für das SubstepInferrer Modul.

Um ein mögliches Overfitting zu erkennen und zu vermeiden, wurde die Architektur mit den ausgewählten Algorithmen der Teilmodule auf einen unabhängigen Validierungsdatensatz angewendet.

Modul	Genaugigkeit
SubstepClassifier	0,79
ProcessClassifier	0,51
SubstepInferrer	0,75

Tabelle 7.10: Ergebnisse der einzelnen Module mit optimalem Algorithmus auf einem Validierungsdatensatz

Wie in Tabelle 7.10 dargelegt, erreichen die einzelnen Module ähnliche Werte wie auf den vorherigen Datensätzen. Das SubstepClassifier Modul erreicht eine Genauigkeit von 79 % und damit 2 Prozentpunkte weniger, das ProcessClassifier Modul erreicht abermals 51 % und das SubstepInferrer Modul erreicht eine Genauigkeit von 75 %, also eine Verringerung um 6 Prozentpunkte. Alle Module erzielen somit geringere Werte, jedoch mit vernachlässigbaren Differenzen, so dass eine Overfitting ausgeschlossen werden kann.

Insgesamt zeigt sich an dieser Stelle, dass die grundsätzliche Idee, eine modulare Architektur zu implementieren, zielführend ist, da für jedes der Teilmodule ein anderer Algorithmus optimal ist. Bei einer Ende-zu-Ende Architektur wäre an dieser Stelle keine Differenzierung möglich gewesen. Es zeigt sich jedoch auch, dass Optimierungspotentiale bestehen, gerade im Hinblick auf Personen, die während des Kochens Unterstützung bekommen sollen und für die es daher wichtig ist, möglichst zutreffende Hilfestellung zu erhalten. Eine genauere Beleuchtung der Optimierungspotentiale finde sich im Kapitel 7.5.

Im Rahmen dieser Arbeit wurde daher exemplarisch ein eingebettetes KI-System implementiert, welches im darauffolgenden Abschnitt motiviert und im Anschluss konzeptioniert, implementiert und getestet wird.

7.5 Optimierung der Prozessanalyse

Mit einer Genauigkeit von 79% für die Teilschrittklassifizierung und 75% für die Teilschrittvorhersage (vgl. Tabelle 7.10) ist bereits eine Unterstützung für Nutzende denkbar, für eine möglichst hilfreiche Begleitung beim Kochen besteht hier jedoch weiteres Optimierungspotential, damit das Feedbacksystem bestens jederzeit die korrekte Information liefert und keine Verwirrung stiftet.

Speziell im vorliegenden Anwendungsfall besteht eine weitere Möglichkeit für den Einsatz von künstlicher Intelligenz an anderer Stelle, nämlich auf den Sensorknoten selbst. Zunächst wird daher untersucht, worin genau die Optimierungspotentiale bestehen. Dazu werden die Konfusionsmatrizen der Ergebnisse des SubstepClassifier und des ProcessClassifier Moduls aus dem letzten Kapitel näher betrachtet.

Abbildung 7.2 zeigt die Konfusionsmatrix des SubstepClassifier Moduls im Hintergrundsystem, die Enkodierung der Klassen erfolgt gemäß Tabelle 7.2. Auf den ersten Blick ist eine deutliche Diagonale zu erkennen, d.h. eine hohe Übereinstimmung zwischen den durch Inferenz ermittelten und den tatsächlichen Klassen liegt vor.

Bei näherer Betrachtung können vorrangig zwei Fehlercluster beobachtet werden:

1. Fehlercluster *Reiben*: Teilschritte 2, *Käse reiben*, und 3, *Kartoffeln reiben*, werden vom System häufig miteinander verwechselt.
2. Fehlercluster *Schneiden*: Teilschritte 9, *Zwiebeln schneiden*, und 10, *Pilze schneiden*, werden ebenfalls vom System häufig verwechselt.

Beide Cluster können darauf zurückgeführt werden, dass sich jeweils beide Handlungen in Dauer und Ausführung stark ähneln. Sowohl das Reiben von Käse als auch von Kartoffeln ist aus Sicht des Sensors, der an der Reibe befestigt ist, austauschbar. Gleiches gilt für den Sensor am Messer, im Bezug auf das Schneiden von Zwiebeln bzw. Pilzen, Teilschritte 9 und 10. Teilschritt 11, *Kartoffeln schneiden* ist gemäß der Konfusionsmatrix von dieser Verwechslung weniger betroffen.

	0	1	2	3	4	5	6	7	8	9	10	11	
0	452	3	0	0	0	0	0	0	0	0	0	0	1
1	6	1274	0	0	0	1	0	1	0	0	0	0	0
2	0	7	82	93	8	9	0	1	0	0	0	0	0
3	0	37	314	402	43	45	1	0	0	0	1	0	0
4	0	3	0	0	172	1	0	0	1	0	1	2	0
5	0	12	2	0	17	275	12	0	0	0	0	0	4
6	1	0	0	1	12	0	108	0	1	0	0	3	0
7	0	0	0	0	2	0	2	27	0	3	0	0	0
8	0	4	1	0	9	1	0	0	426	5	0	8	0
9	1	3	0	0	0	3	0	0	5	96	8	5	0
10	4	1	1	0	0	2	0	3	13	118	39	9	0
11	7	3	0	0	3	3	0	2	14	4	2	186	0
	0	1	2	3	4	5	6	7	8	9	10	11	
	Ermittelte Klasse												

Abbildung 7.2: Konfusionsmatrix für die Testergebnisse des SubstepClassifier Moduls

Die Art, wie Menschen üblicherweise Kartoffeln schneiden unterscheidet sich offensichtlich stärker gegenüber dem Schneiden von Pilzen und Zwiebeln, so dass der ML-Algorithmus die Unterschiede erlernen konnte.

In Abbildung 7.3 ist außerdem die Konfusionsmatrix für den Gesamtprozess, d.h. das Rezept zu sehen, welches vom ProcessClassifier auf Basis der Teilschritte prädiziert wird. Die Klassen sind gemäß Tabelle 7.4 encodiert. An dieser Stelle sei nochmals darauf hingewiesen, dass eine Inferenz für den Gesamtprozess zur Laufzeit bei jedem Durchlauf der KI-Pipeline wie in Abbildung 7.1 dargestellt stattfindet. Das hat zur Folge, dass deutlich mehr Gesamtprozessvorhersagen vorliegen, als tatsächlich Rezepte gekocht wurden, wodurch sich die Zahlenwerte in der Konfusionsmatrix erklären lassen.

	0	1	2	3	4	5	6
0	133	0	0	0	0	0	0
1	70	11	0	0	0	0	0
2	2	0	27	6	1	0	1
3	1	0	7	16	0	0	0
4	84	1	0	0	29	3	2
5	32	1	0	0	12	26	0
6	39	0	0	0	7	9	28

Abbildung 7.3: Konfusionsmatrix für die Testergebnisse des ProcessClassifier Moduls

Auch hier zeigen sich besonders zwei Auffälligkeiten: Die tatsächliche Klasse 1 *Kartoffelrösti* wird sehr häufig mit Klasse 0 *Käseomelette* verwechselt. Ebenso werden die Klassen 4, 5, 6, namentlich *gebratene Pilze*, *gebratene Paprika mit Zwiebeln* und *Bratkartoffeln*, häufig miteinander verwechselt, allerdings auch oft mit Klasse 0, *Käseomelette*

Einige dieser Phänomene ergeben sich als direkte Konsequenz aus den Beobachtungen zu den Fehlern bei der Teilschrittklassifizierung durch den SubstepClassifier, da die Inferenz im ProcessClassifier auf den Ausgangsdaten des Substep-Classifier geschieht. So basieren Kartoffelrösti und Käseomelette maßgeblich auf dem Teilschritt Kartoffeln bzw. Käse reiben, das restliche Rezept ähnelt sich stark. Ebenso basieren die Rezepte der Klassen 4, 5 und 6 vorrangig auf dem Schneiden

der jeweiligen Hauptzutat in Form von Pilzen, Paprika, Zwiebeln oder Kartoffeln, so dass sich hier die Falscherkennung fortsetzt. Einzig die häufige Erkennung der letztgenannten Rezepte als Käseomelette kann hierdurch nicht erklärt werden, es ist zu vermuten, dass hier die Verwendung der Pfanne in Kombination mit dem Pfannenwender zu einer Fehldetektion führt, bzw. sich der zeitliche Ablauf ähnelt.

Zur Optimierung der Erkennungsrate wäre es förderlich die Fehlercluster aufzulösen. Hierzu ist es denkbar, weitere Unterscheidungsmerkmale heranzuziehen, beispielsweise auftretende Beschleunigungen. Hierzu wurde wie schon bei der Sensorknotenkonzeptionierung (Kapitel 6.4) dargelegt, ein Beschleunigungssensor auf der Smartpatchplatine vorgesehen und bestückt.

Exemplarisch soll im weiteren Verlauf untersucht werden, ob sich durch eine Beschleunigungsmessung und einer lokalen Inferenz auf dem Sensor mittels eingebetteter KI der Fehlercluster *Schneiden* auflösen lässt. Die Konzeptionierung des Systems folgt in Kapitel 7.6.

7.6 Konzeptionierung des eingebetteten ML-Systems

Die Implementierung von Machine-Learning-Algorithmen auf einem eingebetteten System unterliegt einer Reihe von Einschränkungen, die sich aus der zugrundeliegenden Hardware ergeben. So muss einerseits der Speicherbedarf des KI-Modells berücksichtigt werden, da sowohl der Fest- als auch der Arbeitsspeicher auf Mikrocontrollern stark beschränkt ist. Andererseits steht nur eine begrenzte Inferenzzeit zur Verfügung, weil der Mikrocontroller auch weiterhin seinen anderen Aufgaben, wie der Messung des RSSI-Werts, periodisch nachkommen muss. Zudem ist die Taktgeschwindigkeit des Prozessors ebenfalls stark begrenzt, weshalb eine Inferenz keine komplexen Berechnung benötigen darf.

Aus diesem Grund wird für die Implementierung eine *Feature Engineering* Technik eingesetzt (vgl. Kapitel 2.5.1). Hierfür wird im ersten Schritt das auftretende Beschleunigungssignal beim Schneiden der vier Gemüsesorten *Pilze*, *Paprika*,

Zwiebeln und *Kartoffeln* genauer betrachtet. In Abbildung 7.4 ist hierfür exemplarisch das Amplitudenspektrum eines einzelnen Kartoffelschnitts dargestellt.

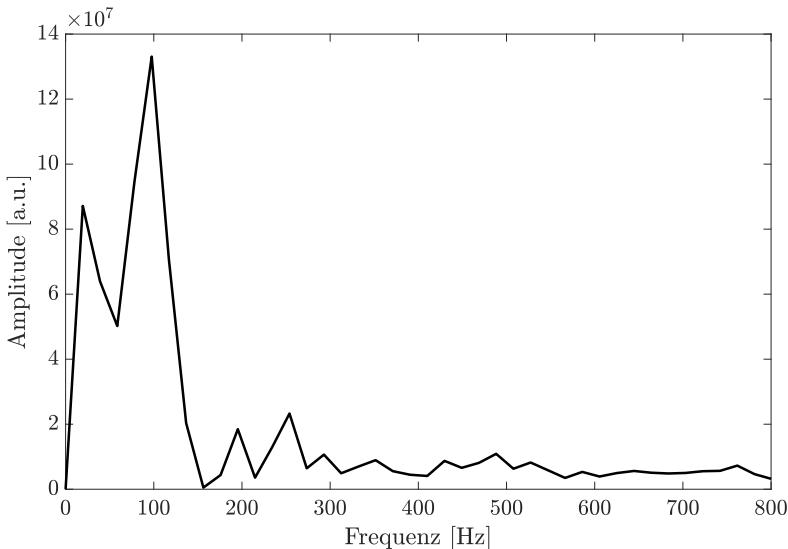


Abbildung 7.4: Spektrum des Beschleunigungssignals in x-Achse eines beispielhaften Kartoffelschnitts

Der Plot wurde mittels einer schnellen Fouriertransformation aus den Beschleunigungswerten des auf dem Smartpatch integrierten Sensors berechnet, der Sensorknoten war hierzu magnetisch an der Schneidefläche befestigt, wodurch die Orientierung des Sensors in seinem Gehäuse vorgegeben werden kann. Dargestellt ist die Amplitude in arbitrary units (a.u.) über der Frequenz in Hertz (Hz). Es wurde die Beschleunigung in x-Achse genutzt, d.h. bei einem Schnitt zum Tisch hin.

Es ist deutlich zu sehen, dass relevante Signale bis zu einer Frequenz von ca. 250 Hz auftreten, dies zeigt sich auch bei Betrachtungen anderer Schnitte und bei anderen Gemüsesorten. Gemäß dem Nyquist-Shannon-Abtasttheorem wird

also eine Abtastrate benötigt, die größer als 500 Hz ist, um Aliasingartefakte im digitalen Signal zu vermeiden. Allerdings gilt es zu untersuchen, in wie fern solche Artefakte für das KI-Modell relevant sind. Die Nutzung einer geringeren Abtastrate senkt den Energiebedarf des Beschleunigungssensors, es wäre also von Vorteil, wenn eine geringere Frequenz ausreichte.

Im ersten Schritt wird aus dem Verlauf der Beschleunigungsdaten einzeln extrahierter Schnitte die Magnitude basierend auf den x-, y- und z-Beschleunigungen berechnet. Die Magnitude stellt die euklidische Norm des Beschleunigungsvektors dar, demnach enthält sie die richtungsunabhängige Information über die Gesamtbeschleunigung.

Basierend auf diesen vier Dimensionen werden für jedes Zeitfenster 12 statistische Parameter berechnet, sowohl im Zeit- als auch im Frequenzbereich: Somit ergeben sich insgesamt 96 Features ($4 \times 12 \times 2$). Die 12 Parameter sind Mittelwert, Median, Standardabweichung, mittlere absolute Abweichung vom Median, Maximum, Minimum, Energie, Interquartilsabstand, Entropie, Schiefe, quadratisches Mittel und Varianz.

Diese Features und die Inferenz darauf müssen zur Laufzeit auf dem Sensorknoten berechnet werden, in einem weiteren Schritt gilt es daher zu untersuchen, in wie weit jedes dieser Merkmale zur korrekten Klassifizierung beiträgt. Um die Datensetze und den Berechnungsaufwand zu reduzieren, wird daher dann die Anzahl der Features auf die relevantesten begrenzt werden. Die genaue Vorgehensweise folgt nun in Kapitel 7.7

7.7 Implementierung des eingebetteten ML-Systems

Der verwendete Mikrocontroller stellt 256 KB Arbeitsspeicher, 1 MB Flashspeicher und eine mit maximal 64 MHz taktende Cortex-M4 CPU mit Floating Point Unit bereit, KI-Modelle basierend auf neuronalen Netzen sind hierfür daher nicht geeignet. Stattdessen wurden klassische Machine Learning Verfahren für erste Tests ausgewählt: K-Nearest Neighbour (KNN), Support Vector Machine

(SVM), Decision Trees (DT), Random Forest (RF) und Gaussian Naive Bayes (NB). Diese Verfahren zeichnen sich durch einen geringen Datenbedarf und eine schnelle Inferenzzeit aus. Erste Vorabtests zur Untersuchung der notwendigen Abtastrate werden zunächst auf einem gewöhnlichen Desktop-PC durchgeführt. Die Daten hierfür und für alle weiteren Tests in diesem Kapitel werden mit einem an einem Messer angebrachten Smartpatch Sensorknoten erzeugt, die Übertragung der Rohbeschleunigungsdaten findet über eine Kabelverbindung zum Computer statt.

In Tabelle 7.11 sind die Genauigkeiten für die Modelle bei einer Abtastrate von 500 Hz dokumentiert, in Tabelle 7.12 die Ergebnisse für 25 Hz.

Algorithmus	Genauigkeit
KNN	0,91
SVM	0,89
DT	0,78
RF	0,87
NB	0,85

Tabelle 7.11: Ergebnisse für die möglichen Algorithmen bei einer Abtastrate von 500 Hz

Bei 500 Hz ergibt sich für den KNN-Ansatz der höchste Score mit 91 %, während ein Decision Tree nur 78 % erreicht. Die anderen Algorithmen bewegen sich auf vergleichbarem Niveau mit 85 - 89 %.

Bei einer Unterabtastung mit 25 Hz ändern sich die Ergebnisse deutlich: Der KNN verliert ungefähr 16 Prozentpunkte, auch die Erkennungsgenauigkeit der anderen Modelle verringert sich, nur das RF-Modell kann eine Genauigkeit von 87 % halten. Aufgrund dieser Robustheit wird für den weiteren Verlauf der Entwicklung das RF-Modell ausgewählt.

Das Modell wird auf dem Mikrocontroller unter Zuhilfenahme der *emlearn Machine Learning inference engine for Microcontrollers and Embedded Devices* Bibliothek [100] als C-header implementiert. Um eine Klassifizierung zeitnah

Algorithmus	Genauigkeit
KNN	0,75
SVM	0,82
DT	0,73
RF	0,87
NB	0,75

Tabelle 7.12: Ergebnisse für die möglichen Algorithmen bei einer Abtastrate von 25 Hz

zur eigentlichen Handlung während der Laufzeit durchführen zu können, wird ein *Sliding Window* (deutsch: Schiebefenster) Ansatz verfolgt. Hierzu wird bei 200 Hz Abtastrate alle drei Sekunden ein Buffer von 600 Beschleunigungsdatensets gefüllt, wobei die Sliding Window Länge 300 Sets beträgt, so dass alle 1,5 Sekunden eine neue Inferenz mit 300 vorangegangen und 300 aktuellen Daten durchgeführt werden kann. Tests haben gezeigt, dass 1,5 Sekunden ein guter Mittelweg zwischen Genauigkeit der Klassifikation und Aktualität sind.

Weiterhin wird zur Implementierung auf dem Controller die Anzahl an Merkmalen von 96 auf die 25 relevantesten reduziert und die Anzahl und Tiefe der Bäume im Random Forest minimiert. Dies geschieht in einem zweistufigen Verfahren: Zunächst werden mittels Pearson Korrelation zwei Features paarweise untersucht. Ist die Korrelation hoch, kann auf eines der beiden Merkmale verzichtet werden. Im zweiten Schritt kommt der Recursive Feature Elimination with Cross Validation (RFECV) Algorithmus zum Einsatz. Hierzu werden automatisiert so lange RF-Modelle mit Teilmengen der zur Verfügung stehenden Features trainiert und getestet, bis genügend Merkmale eliminiert wurden, um den Speicherbedarf ausreichend einzuschränken für eine Ausführung auf dem Mikrocontroller bei gleichzeitiger Erreichung der höchstmöglichen Genauigkeit. Die 25 ausgewählten Features in absteigender Relevanz sind in Tabelle 7.13 aufgelistet.

Feature, absteigende Relevanz
Interquartilabstand der Magnitude im Zeitbereich
Median der x-Beschleunigung im Zeitbereich
Median der Magnitude im Zeitbereich
Varianz der z-Beschleunigung im Zeitbereich
Mittlere absolute Abweichung vom Median der y-Beschleunigung im Frequenzbereich
Energie der x-Beschleunigung im Zeitbereich
Varianz der Magnitude im Zeitbereich
Maximum der x-Beschleunigung im Zeitbereich
Minimum der x-Beschleunigung im Zeitbereich
Schiefe der x-Beschleunigung im Zeitbereich
Interquartilabstand der z-Beschleunigung im Zeitbereich
Maximum der z-Beschleunigung im Zeitbereich
Interquartilabstand der y-Beschleunigung im Frequenzbereich
Maximum der y-Beschleunigung im Zeitbereich
Interquartilabstand der Magnitude im Frequenzbereich
Minimum der Magnitude im Zeitbereich
Schiefe der Magnitude im Zeitbereich
Varianz der x-Beschleunigung im Zeitbereich
Median der y-Beschleunigung im Frequenzbereich
Median der y-Beschleunigung im Zeitbereich
Minimum der z-Beschleunigung im Zeitbereich
Standardabweichung der Magnitude im Frequenzbereich
Median der z-Beschleunigung im Zeitbereich
Mittlere absolute Abweichung vom Median der x-Beschleunigung im Frequenzbereich
Mittelwert der Magnitude im Zeitbereich

Tabelle 7.13: Die 25 wichtigsten Features in absteigender Relevanz

In einem abschließenden Schritt wurde mittels Zufallssuche im Hyperparameterraum 200 verschiedene Kombinationen aus Anzahl von Bäumen und Tiefe der Bäume getestet, um so letztlich eine optimale Kombination zu erhalten. Diese beträgt eine Tiefe von 10 und eine Anzahl von 14 Entscheidungsbäumen im RF-Modell.

In einem weiteren Test, dokumentiert in Tabelle 7.14, wird einerseits abermals

die notwendige Abtastrate untersucht, als auch die Tauglichkeit für einen Einsatz auf dem Controller, da durch Anwendung der Featurereduktion und des Schiebefensteralgorithmus mit einer Verschlechterung der Klassifizierungsgenauigkeit zu rechnen ist.

Abtastrate	Genauigkeit
25 Hz	0,69
200 Hz	0,74
500 Hz	0,77

Tabelle 7.14: Ergebnisse des eingebetteten Random Forest Modells für verschiedene Abtastraten

Es zeigt sich im Vergleich mit den Tabellen 7.11 und 7.12 die erwartete Verschlechterung der Genauigkeit, auch bei gleichen Abtastraten gegenüber der Ausführung auf einem Desktop-PC. Diese Verschlechterung wird an dieser Stelle akzeptiert und eine Abtastfrequenz von 200 Hz beibehalten, da eine Erhöhung auf 500 Hz keine signifikante Verbesserung der Klassifizierung nach sich zieht. Eine Erhöhung auf 500 Hz und damit eine potentielle Erzeugung neuer Inferenzen alle 0,6 Sekunden bei einer fünfzigprozentigen Überlappung der Fenster wirkt sich in der Praxis sogar nachteilig aus, da der Versand der Datenpakete und das Scannen nach fremden Datenpaketen zur Messung der Signalstärke über Bluetooth ebenfalls Zeit benötigt, wodurch das zu häufige Auslösen der Interruptroutine für das Inferieren der Bluetoothkommunikation im Wege stehen würde. Es wird daher weiterhin eine Abtastfrequenz von 200 Hz eingesetzt. Zukünftig bestehen an dieser Stelle durch eine geeignete Parallelisierung aber weitere Optimierungspotentiale.

Es folgt nun abschließend der zweite technische Prätest mit Testpersonen in Kapitel 7.8.

7.8 Zweiter technischer Prätest

Abschließend wurde ein Test mit 13 Personen durchgeführt. Jede der Proband*innen schnitt acht Pilze, sechs Kartoffeln, drei Paprika und vier Zwiebeln, insgesamt wurden damit 104 Pilze, 78 Kartoffeln, 39 Paprikas und 52 Zwiebeln geschnitten. Im Durchschnitt führte jede Testperson 75 Pilzschnitte, 97 Kartofelschnitte, 98 Paprikaschnitte und 72 Zwiebelschnitte durch, insgesamt wurden von allen Personen also ca. 4400-mal geschnitten.

Bei jedem Schnitt war ein Smartpatch am Messer angebracht und die Inferenz wurde auf dem Sensorknoten durchgeführt. Die Proband*innen erhielten keine speziellen Instruktionen und durften die Aufgabe, das ihnen vorgelegte Gemüse zu schneiden, nach eigenen Vorstellungen erledigen. Tabelle 7.15 zeigt die Ergebnisse aus dieser Testung.

Gemüseart	Genauigkeit
Kartoffeln	0,76
Pilze	0,62
Zwiebeln	0,69
Paprika	0,72

Tabelle 7.15: Ergebnisse des eingebetteten Random Forest Modells für verschiedenes Gemüse

Kartoffelschnitte konnten vom eingebetteten Random Forest Modell zur Laufzeit mit einer Genauigkeit von 76 % erkannt werden, Paprika mit 72 %, Zwiebeln mit 69 % und Pilze mit 62 %, insgesamt ergibt sich also eine hohe Genauigkeit für die Klassifikation. Einen detaillierteren Blick lässt Abbildung 7.5 zu, welche die zugehörige Konfusionsmatrix zeigt.

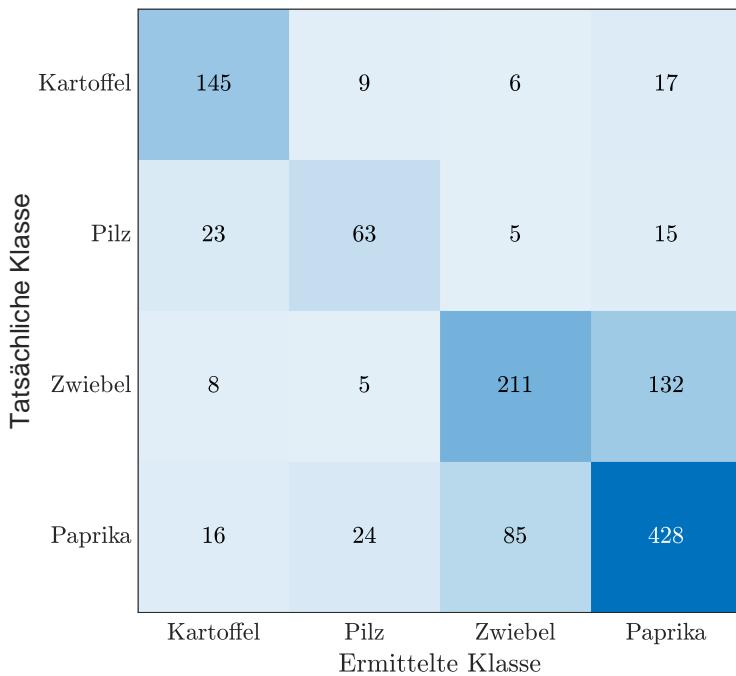


Abbildung 7.5: Konfusionsmatrix des eingebetteten Random Forest Modells

Auf den ersten Blick ist die gewünschte Diagonale, d.h. eine korrekte Klassifizierung zu erkennen. Die Klassen *Zwiebel* und *Paprika* zeigen eine erhöhte Anfälligkeit für Fehldetektion, das Schneiden dieser beiden Gemüsearten ist für das RF-Modell schwieriger zu differenzieren. Es ist möglich, dass hier eine weitere Erhöhung der Datenmenge und -varianz die Genauigkeit weiter verbessern könnte.

Insgesamt stellt dies ein sehr gutes Ergebnis dar, wenn man die Randbedingungen in Betracht zieht: Das Modell ist kleiner als ein Megabyte, kann alle 1,5 Sekunden eine neue Inferenz durchfuhren und verbraucht dabei nur ein Minimum an Energie. Zusammenfassend kann das Gesamtsystem nun hinreichend genau Vorhersagen zu Teilschritten im Kochrezept durchfuhren, so dass nun in Kapitel 8 die Endevaluation des Gesamtsystems mit neuen Rezepten und Testpersonen in einer Kuche erfolgen kann.

8 Evaluation und Ergebnisdiskussion

Die Teilsysteme, bestehend aus den Sensorknoten, dem sich daraus ergebenden Sensornetzwerk sowie dem Hintergrundsystem wurden in den vorhergehenden Kapiteln detailliert vorgestellt und bereits technische Tests zur Performanz durchgeführt. Diese Tests konnten zeigen, dass die Funktionalität des Gesamtsystems gegeben und folglich eine Unterstützung von Personen beim Kochen möglich ist. In diesem Kapitel wird eine Feldtestung vorgestellt und diskutiert, die zeigt, in wie weit das System tatsächlich der Anforderung genügt, dass es einen Nutzen für die anwendenden Personen entfaltet. Hierfür liegt dementsprechend der Fokus dieser Evaluation auf Betrachtungen zur Usability.

Zu Beginn des Kapitels wird der Testaufbau und der Testablauf beschrieben, gefolgt von einem Überblick der Messungen inklusive einer Beschreibung des Probandenkollektivs. Daraufhin werden die Testergebnisse zu den verschiedenen Messgrößen betrachtet und in einem letzten Schritt diskutiert.

8.1 Messaufbau und Ablauf

Die Studie wird in einer Küche des FZI Forschungszentrum Informatik durchgeführt, welche in Aufbau und Funktion einer typischen Privatküche entspricht. Für die Studie werden der Induktionsherd, das Waschbecken und die Arbeitsfläche verwendet, auf letzterer befindet sich auch ein Laptop zur Speicherung der Daten, welche über den per Kabel verbundenen Gateway-Knoten des Sensornetzwerks

empfangen werden. Die Sensorknoten selbst werden an Topf, Messer, Reibe, Salz und Löffel befestigt (vgl. Abbildung 8.1).

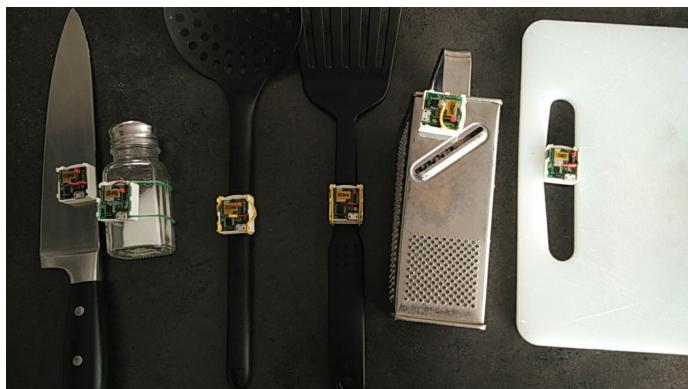


Abbildung 8.1: Die Sensoren an verschiedenen Utensilien

Damit die Küchenutensilien für jede Versuchsperson wieder am selben Ausgangspunkt liegen, werden die entsprechenden Startpositionen auf der Küchenzeile mittels Markern gekennzeichnet und dort, nachdem sie gesäubert wurden, vom Studienleiter nach jedem Versuchsdurchlauf wieder platziert. Teil der Rezepte sind darüber hinaus ein Teller, ein Messbecher, ein Schneidbrett, Besteck und eine Flasche Öl. An diesen Gegenständen befinden sich keine Sensorknoten. Außerdem sind die Zutaten der jeweiligen Rezepte noch Teil des Aufbaus, diese werden für die Durchführung ebenfalls auf der Arbeitsfläche bereitgestellt.

Zu Beginn erhält jede Person eine kurze Einführung in die Funktionsweise des Systems und der weitere Ablauf wird erklärt. Während eines Versuchsdurchlaufs kocht jede Person drei Rezepte, jedes davon wird unmittelbar vor dem Kochvorgang durch den Studienleiter einmalig vorgelesen und soll im Anschluss auswendig, bzw. gegebenenfalls mit Hilfe des Feedbacksystems, gekocht werden. Sollte die Person während des Kochens nicht weiterwissen, gibt es notfalls die Möglichkeit das Rezept nachzulesen, da es ebenfalls auf der Tischplatte zur Verfügung liegt, jedoch in abgedeckter Form. Eine Verwendung des Rezeptabschrebs wird entsprechend vom Versuchsleiter für die weitere Auswertung notiert. Während

des Durchlaufs dokumentiert der Studienleiter zudem weitere Auffälligkeiten und Kommentare der Versuchspersonen im Sinne der Methode einer *teilnehmenden Beobachtung*. Dieser Methodik folgend, sind die Versuchspersonen daher auch angehalten, ihre Gedanken zum System laut auszusprechen.

Unmittelbar nach der Durchführung des dritten Rezepts sind mehrere Fragebögen zu beantworten: Zwei standardisierte Fragebögen, System Usability Scale und Raw Task Load Index (zu finden in Anhang A.2 und A.3) und ein weiterer Fragebogen mit Fragen zur Person. Sieben Teilnehmende erhielten außerdem einen zusätzlichen, nicht-standardisierten Fragebogen (dokumentiert in Anhang A.4). Des Weiteren wird in einem offenen Teil um allgemeine Kommentare bzw. Feedback gebeten.

Die drei gekochten Rezepte sind nachfolgend jeweils einerseits in der Form aufgelistet, wie sie auch die Testpersonen gehört haben, andererseits aber auch mit den aus Systemsicht relevanten Schritten tabellarisch dargestellt. Sie wurden so ausgewählt, dass die einzelnen Rezeptschritte von jeder Person gut durchführbar sind, keinen besonderen hygienischen Anforderungen genügt werden muss, wie beispielsweise bei der Zubereitung von Fisch, und jedes Rezept in circa 20 Minuten gekocht werden kann, so dass die Gesamtdauer eines Versuchsdurchlaufs handhabbar ist.

8.1.1 Rezept 1: Gedünstete Karotten

Das erste Rezept besteht aus den nachfolgenden 11 Schritten, in Tabelle 8.1 sind die relevanten Schritte aus Systemsicht aufgezählt und vermerkt, welcher Schritt ein sensorgestütztes Utensil enthält. Das Rezept ist das einfachste und als Einstieg für die Proband*innen gedacht, um sich mit der Funktionsweise vertraut zu machen.

Rezept 1: Gedünstete Karotten

Zutaten: 1 Karotte, Olivenöl, Wasser

Küchenutensilien mit Sensorknoten: Messer, Löffel, Salz

Weitere Utensilien: Topf, Messbecher, Schneidbrett, Gabel, Teller

1. Stelle den Topf auf den Herd.
2. Schneide die Karotte in Scheiben.
3. Erhitze das Öl im Topf.
4. Gib die Karottenscheiben in den Topf.
5. Rühre immer wieder mit dem Löffel um.
6. Gebe ein bisschen Salz zu den Karotten.
7. Rühre mit dem Löffel um.
8. Gib etwas Wasser zu den Karotten, damit sie nicht anbrennen.
9. Rühre die Karotten um.
10. Dünste die Karotten solange du möchtest, benutze die Gabel falls du testen möchtest ob die Karotten weich genug sind.
11. Serviere die Karotten mit dem Löffel.

Dieses Rezept wurde, wie alle nachfolgenden Rezepte, den Testpersonen zu Beginn in dieser Form einmal vorgelesen. Die Proband*innen waren informiert, dass es Teil der Aufgabe ist, das Rezept zu memorieren und danach selbstständig nachzukochen.

Schritt	Aktion	Sensorgestütztes Utensil
1	Topf auf Herd stellen	Topf
2	Karotte schneiden	Messer
3	Öl erhitzen	-
4	Karotte in Topf geben	-
5	Umrühren	Löffel
6	Salzen	Salzstreuer
7	Umrühren	Löffel
8	Wasser hinzugeben	-
9	Umrühren	Löffel
10	Dünsten	-
11	Servieren	Löffel

Tabelle 8.1: Schritte des ersten Rezepts aus Systemsicht

8.1.2 Rezept 2: Gemüsesuppe

In Tabelle 8.2 sind die insgesamt 12 Schritte des zweiten Rezept notiert, woraus letztlich eine einfache Gemüsesuppe entsteht. Es werden hierbei ähnliche Utensilien zu ähnlichen Zeitpunkten wie im ersten Rezept eingesetzt, wodurch eine Unterscheidung durch das System nicht trivial ist und damit eine erfolgreiche Unterscheidung ein Zeichen für Robustheit sind.

Rezept 2: Gemüsesuppe

Zutaten: 1 Karotte (geschält), 1 Kartoffel (geschält), halbe Zwiebel (geschält), Wasser

Küchenutensilien mit Sensorknoten: Messer, Salz, Suppenlöffel

Weitere Utensilien: Topf, Schneidbrett, Messbecher, Esslöffel, Suppenteller/Schüssel

1. Stelle den Topf auf den Herd.
2. Schneide die Karotte, Kartoffel und Zwiebel in Würfel.
3. Gib 0,5l Wasser in den Topf und lasse es aufkochen.
4. Wenn das Wasser kocht, füge 1 TL Salz hinzu.
5. Gib das Gemüse in den Topf.
6. Rühre das Gemüse um.
7. Lass die Suppe kochen solange du möchtest (ca. 10 Minuten).
8. Serviere die Suppe mit dem Löffel.

Schritt	Aktion	Sensorgestütztes Utensil
1	Topf auf Herd stellen	Topf
2	Karotten würfeln	Messer
3	Kartoffeln würfeln	Messer
4	Zwiebel in Scheiben	Messer
5	Wasser hinzufügen	-
6	Wasser kochen	-
7	Salz hinzufügen	Salzstreuer
8	Gemüse hinzugeben	-
9	Umrühren	Löffel
10	Bei Bedarf mehr Wasser hinzufügen	-
11	Suppe kochen	-
12	Servieren	Löffel

Tabelle 8.2: Schritte des zweiten Rezepts aus Systemseicht

8.1.3 Rezept 3: Nudeln mit Tomaten

Das dritte Rezept ist das umfangreichste mit 17 Schritten und stellt damit bereits mehr Anforderungen an das Gedächtnis der Versuchspersonen, umgekehrt wird der Nutzen des Systems dadurch aber auch deutlicher, wenn die Proband*innen im richtigen Moment eine Erinnerung durch das System erhalten.

Rezept 3: Nudeln mit Tomaten

Zutaten: 100g Nudeln, Käse (Stückchen), 1 Tomate

Küchenutensilien mit Sensorknoten: Reibe, Messer, Salz, Löffel

Weitere Utensilien: Topf, Teller, Messbecher, Gabel, Sieb

1. Reibe den Käse
2. Stelle den Topf auf den Herd.
3. Gib Wasser in den Topf und lasse es aufkochen.
4. Gib Salz ins Wasser.
5. Gib die Nudeln ins Wasser.
6. Rühre die Nudeln mit dem Löffel um.
7. Lasse die Nudeln 11 Minuten kochen.
8. Gieße die Nudeln mithilfe des Siebs ab, stelle den Topf zurück auf seinen Platz – benutze die Gabel falls noch Nudeln im Topf kleben.
9. Schneide die Tomate in Stückchen.
10. Gib etwas Öl in den Topf und erhitze es.
11. Gib die Tomatenstückchen in den Topf.
12. Rühre mit dem Löffel.
13. Gib die Nudeln in den Topf.
14. Rühre mit dem Löffel.
15. Serviere das Essen mit dem Löffel und verteile den Käse obendrauf.

Schritt	Aktion	Sensorgestütztes Utensil
1	Käse reiben	Reibe
2	Topf auf Herd stellen	Topf
3	Wasser hinzufügen	-
4	Wasser kochen	-
5	Salzen	Salzstreuer
6	Nudeln hinzufügen	-
7	Umrühren	Löffel
8	Nudeln kochen	-
9	Nudeln abgießen	Topf
10	Tomaten schneiden	Messer
11	Topf auf Herd stellen	Topf
12	Öl hinzufügen	-
13	Nudeln und Tomaten in den Topf geben	-
14	Umrühren	Löffel
15	Käse hinzufügen	-
16	Umrühren	Löffel
17	Servieren	Löffel

Tabelle 8.3: Schritte des dritten Rezepts aus Systemsicht

Für alle Tabellen ist zu vermerken, dass je nach Proband*in die eingesetzten sensorgestützten Utensilien von obigen Tabellen abweichen können, so benutzen beispielsweise manche Versuchspersonen das Messer, um Zutaten in den Topf vom Schneidbrett zu schieben. Hierzu wurden seitens des Versuchsleiters jedoch keine Vorgaben gemacht, um die Interaktion und den Kochprozess so natürlich wie möglich zu gestalten.

8.2 Übersicht der Messungen

Das Probandenkollektiv umfasst 19 Personen, hiervon sechs weiblich und 13 männlich mit einem Alter zwischen 24 und 35 Jahren ($\mu = 27,84$ Jahre, $\sigma = 2,91$ Jahre, $n = 19$). Alle kochten die gleichen Rezepte und hielten den selben Ablauf, wie zuvor beschrieben, ein, allerdings wurde bei sechs Personen das Feedbacksystem nicht verwendet, um eine Kontrollgruppe für die Auswertung mit einzubeziehen.

Von den 19 Personen sagten drei, dass sie selten kochen, drei ca. einmal pro Woche und 13 Personen gaben an täglich zu kochen. Alle außer einer Person schätzten sich selbst als technikaffin ein. Auf die Frage, ob sie bereits ein Assistenzsystem im Alltag eingesetzt haben, wurden vor allem Fahrerassistenzsysteme benannt, Sprachassistenz sowie Multifunktionsküchengeräte mit Automatikfunktionen. Insgesamt wurden mit 19 Personen und drei Rezepten pro Durchlauf demnach zusammen 57 Kochprozesse durchgeführt, die Gesamtmessdauer beträgt ca. 18,5 Stunden, mit einer Durchschnittsmessdauer von 12,1 Minuten für Rezept 1, 19,4 Minuten für Rezept 2 und schließlich 23,8 Minuten für Rezept 3.

8.3 Ergebnisse der Befragung

Nachdem im vorangegangenen Abschnitt der Versuchsaufbau und eine Übersicht der Messungen dargelegt wurde, folgt nun die eigentliche Auswertung getrennt nach System Usability Scale und Task Load Index, inklusive einer jeweiligen Diskussion der Ergebnisse.

8.3.1 Auswertung und Diskussion der System Usability Scale

Von den 19 Proband*innen haben 13 Personen mit Feedbacksystem gekocht, entsprechend wurde durch diese Personen der System Usability Scale-Fragebogen ausgefüllt. Die Ergebnisse sind in Tabelle 8.4 dargestellt.

Gruppe	μ	\tilde{x}	σ
Gesamt	84,3	86,7	9,4
Täglich	85,3	86,7	9,3
Nicht täglich	81,7	85	10,4

Tabelle 8.4: Vergleich des Mittelwerts μ , Medians \tilde{x} und der Standardabweichung σ des SUS-Scores zwischen verschiedenen Testgruppen

Insgesamt erreichte das System einen SUS-Score von 84,3 im Durchschnitt, bzw. 86,7 im Median mit einer Standardabweichung von 9,4. Bei den täglich kochenden Personen ($n = 9$) konnte ein Ergebnis von durchschnittlich 85,3 erzielt werden, oder 86,7 im Median, mit einer Standardabweichung von 9,3. Nicht täglich kochende Personen ($n = 4$), d.h. Proband*innen, die von sich angaben *selten* oder *wöchentlich* zu kochen, bewerteten das System im Schnitt mit 81,7, beziehungsweise mit 85 im Median, mit einer Standardabweichung von 10,4.

Zunächst ist zu konstatieren, dass nach der typischen Skaleneinteilung des SUS-Scores (vgl. Kapitel 2.2.2) die Usability des Systems mit *gut* zu bezeichnen ist. Betrachtet man den Median, dann kann sie sogar als *exzellent* bezeichnet werden, da diese Bewertung ab einem SUS-Score von 85 gegeben ist.

Eine exakte Erklärung, warum täglich kochende Personen das System besser bewerten als nicht-täglich kochende lässt sich nicht geben, es ist nicht ausgeschlossen, dass der kleine Stichprobenumfang hier zu Verzerrungen führt. Eine mögliche Überlegung wäre jedoch, dass die Benutzungsfreundlichkeit des Systems für Personen, die im Umgang mit Küchenutensilien geübt sind, offensichtlicher ist, als für Personen, die weniger geübt sind und die durch den gewöhnlichen Kochvorgang, der nicht vom System unterstützt wird, einer größeren Herausforderung

gegenüberstanden. Alltägliche Funktionen, wie beispielsweise das Bedienen des Herdes, fallen so mehr ins Gewicht und könnten die Gesamtbewertung negativ beeinflusst haben.

Es folgt eine detailliertere Betrachtung der einzelnen Fragen, der Überblick ist in Tabelle 8.5 dokumentiert.

Frage	μ	\tilde{x}	σ
1: Ich kann mir sehr gut vorstellen, das System regelmäßig zu nutzen.	6,2	6,7	2,2
2: Ich empfinde das System als unnötig komplex.	9,3	10	1,4
3: Ich empfinde das System als einfach zu nutzen.	8,3	10	2,2
4: Ich denke, dass ich technischen Support brauchen würde, um das System zu nutzen.	9,0	10	2,0
5: Ich finde, dass die verschiedenen Funktionen des Systems gut integriert sind.	7,9	6,7	1,7
6: Ich finde, dass es im System zu viele Inkonsistenzen gibt.	7,4	6,7	2,7
7: Ich kann mir vorstellen, dass die meisten Leute das System schnell zu beherrschen lernen.	10,0	10	0
8: Ich empfinde die Bedienung als sehr umständlich.	9,8	10	0,9
9: Ich habe mich bei der Nutzung des Systems sehr sicher gefühlt.	7,1	6,7	1,8
10: Ich musste eine Menge Dinge lernen, bevor ich mit dem System arbeiten konnte.	9,3	10	1,4

Tabelle 8.5: Mittelwert μ , Median \tilde{x} und Standardabweichung σ der Einzelfragen des SUS

Jede der Einzelfragen kann gemäß der Umrechnung für das Scoring (vgl. Kapitel 2.2.2) eine Maximalpunktzahl von 10 erhalten. Je höher der Wert, desto besser; dies gilt auch für die Fragen 2, 4, 6, 8 und 10, die negiert gestellt sind, d.h. ein hoher Wert bedeutet hier, dass die Person **nicht** zustimmt. Für die Tabelle wurden

für jede Einzelfrage aller 13 Proband*innen, die den SUS-Fragebogen ausfüllten, der Mittelwert, Median und die Standardabweichung gebildet.

Zunächst fällt das Ergebnis von **Frage 1** ins Auge, das mit einem Durchschnittswert von 6,2/10 Punkten schwächer ausfällt. Die Frage zielt auf die Nutzungs-wahrscheinlichkeit der Proband*innen ab. Jede Testperson erhielt zu Beginn der Durchführung eine kurze Einweisung und auch eine Erläuterung zum Hintergrund und potentiellen Anwendungsfällen des Projekt, beispielsweise im Kontext von älteren Personen oder Personen mit leichten kognitiven Einschränkungen. Es ist daher anzunehmen, dass eine Notwendigkeit für den persönlichen Alltag der jüngeren Probandengruppe weniger gesehen wurde.

Über **Frage 9**, mit einem Wert von 7,1/10 Punkten, wird eine gewisse Unsicherheit der Probandengruppe deutlich. Ein möglicher Einflussfaktor ist hierbei das Design und die Anbringung der Sensorknoten selbst: Im Sinne einer prototypischen Entwicklung verfügen die Sensorsysteme über kein versiegeltes Gehäuse, eindringen von Wasser kann die Sensoren beschädigen. Die Proband*innen haben daher darauf geachtet, sorgsam mit den elektronischen Komponenten umzugehen, was aber letztlich für eine Unsicherheit im Umgang geführt hat.

Die Interpretation von **Frage 6** (7,4/10 Punkte) ist vielfältig, da eine Frage nach „Inkonsistenzen“ sowohl auf die technische Benutzung, als auch auf den Sinn und Zweck des Anwendungsfalls bezogen werden kann. Die unklare Zielstellung der Frage zeigt sich auch in der hohen Standardabweichung ($\sigma = 2,7$), der höchsten aller SUS-Einzelfragen.

Die Frage nach der Integration der verschiedenen Funktionen, **Frage 5**, erhält 7,9/10 Punkte. Das Proband*innenkollektiv empfindet damit die verschiedenen Funktionen bereits als gut integriert, Potentiale bestehen aber in Integration weiterer Feedbackoptionen, beispielsweise einem Blinken oder einem Farbwechsel für unterschiedliche Hilfestellung.

Demgegenüber steht die **Frage 2**, denn mit 9,3/10 Punkten wurde das System als nicht unnötig komplex bewertet. Eine Integration weiterer Funktionalitäten und Optionen könnte im Gegenzug diesen Wert negativ beeinflussen.

Die Versuchspersonen empfanden das System überwiegen auch einfach zu nutzen (**Frage 3**, 8,3/10 Punkten), wobei die Bewertungen mit einer Standardabweichung $\sigma = 2,2$ größere Abweichungen zwischen den Proband*innen enthielt.

Nur wenige Nutzende erwarten einen Bedarf an technischem Support (**Frage 4**, 9,0/10 Punkten), was ebenfalls für eine einfache Anwendbarkeit des Systems spricht.

Auch der Bedarf, Neues Lernen zu müssen, um mit dem System arbeiten zu können, wurde nur sehr gering eingeschätzt (**Frage 10**, 9,3/10 Punkten), die Proband*innen sahen sich also schnell in die Lage versetzt das System anzuwenden. Die überwiegende Mehrheit empfindet das System als wenig umständlich in der Bedienung, was sich in einer Bewertung von **Frage 8** mit 9,8/10 Punkten bei einer Standardabweichung von $\sigma = 0,9$ zeigt.

Frage 7 wurde mit einer maximalen möglichen Punktzahl von 10,0/10 Punkten bewertet, alle Teilnehmenden der Studie waren sich einig, dass das System schnell zu beherrschen ist.

Abstriche sind insgesamt in der technischen Umsetzung (Fragen 5, 6 und 9) gegeben, die aufgrund der prototypischen Implementierung im Rahmen dieser Arbeit noch nicht das Niveau eines Produkts erreichen kann. Hier bestehen Potentiale für die weitere Ausgestaltung eines Systems, welches auch in der Praxis durch eine Integration aller notwendigen Funktionalitäten und einer einfachen Handhabbarkeit, beispielsweise durch eine Spülmaschinentauglichkeit, bestehen kann.

Im Ganzen spricht das Ergebnis jedoch für ein System, welches der zu erwartenden Zielgruppe von Senior*innen und Personen mit leichten kognitiven Einschränkungen gerecht werden kann. Dies zeigt sich vor allem in den Fragen nach der Einfachheit der Bedienung (Fragen 2, 3, 4 und 8) und Erlernbarkeit (Fragen 7 und 10), die sehr gut bewertet wurden.

8.3.2 Auswertung und Diskussion des nicht-standardisierten Fragebogens

Von den 13 Personen, die mit Feedbacksystem kochten, beantworteten sieben Proband*innen zusätzlich einen nicht-standardisierten Fragebogen (vgl. Anhang A.4) mit Fragen zur Nützlichkeit und dem Feedbacksystem selbst. Die Ergebnisse sind in Tabelle 8.6 dargestellt. Die Auswertung ist an das Scoringverfahren der System Usability Scale angelehnt: Die vier Antwortmöglichkeiten werden von 0 bis 10 skaliert und anschließend die Einzelfragen aufsummiert. Da es jedoch nur sechs Fragen sind, beträgt der maximal erreichbare Gesamtwert 60 Punkte (im Gegensatz zu 100 Punkten beim SUS-Score).

Gruppe	μ	\tilde{x}	σ
Gesamt	46,7	46,7	6,9
Täglich	47,5	46,7	6,9
Nicht täglich	45,6	46,7	8,4

Tabelle 8.6: Vergleich des Mittelwerts μ , Medians \tilde{x} und der Standardabweichung σ des Scorings für den nicht-standardisierten Fragebogen zwischen verschiedenen Testgruppen

Insgesamt wurde das System mit einem Score von 46,7 im Durchschnitt und Median bewertet, die Standardabweichung beträgt 6,9. Die täglich kochenden Personen ($n = 4$) bewerteten das System leicht besser ($\mu = 47,5$, $\tilde{x} = 46,7$, $\sigma = 6,9$) als die Gruppe der nicht-täglich Kochenden ($n = 3$, $\mu = 45,6$, $\tilde{x} = 46,7$, $\sigma = 6,9$). Insgesamt ist jedoch zu vermerken, dass keine große Variation in den Antworten aller Personen vorlag, was sich in einer niedrigen Standardabweichung zeigt.

Es folgt eine detailliertere Betrachtung der einzelnen Fragen, der Überblick ist in Tabelle 8.7 dokumentiert.

Für die negiert gestellten Fragen 2, 4 und 6 gilt ebenfalls wie beim SUS-Scoring, dass ein hoher Wert bedeutet, dass die Person **nicht** zustimmt, sie also einen

Frage	μ	\tilde{x}	σ
1: Ich kann mir vorstellen, z.B. meinen Großeltern das System zu kaufen.	7,1	6,7	3,0
2: Das System hat mich beim Kochen gestört.	8,1	6,7	1,8
3: Durch das System konnte ich die richtige Reihenfolge leichter einhalten.	7,6	6,7	1,6
4: Die LEDs haben mich abgelenkt.	9,5	10	1,26
5: Mehr Utensilien sollten mit Sensorknoten ausgestattet sein.	8,6	10	2,6
6: Eine andere Art von Feedback, z.B. Sprache oder Vibration, stelle ich mir hilfreicher vor.	5,7	6,7	1,6

Tabelle 8.7: Mittelwert μ , Median \tilde{x} und Standardabweichung σ der Einzelfragen des nicht-standardisierten Fragebogens

positiven Eindruck vom System bezüglich der Frage hat. Jede Frage kann abermals eine Maximalpunktzahl von 10 erhalten, die Tabelle zeigt die Mittelwerte, Mediane und Standardabweichungen für jede Frage getrennt.

Mit einem Wert von 5,7/10 Punkten zeigt sich in **Frage 6**, dass die Proband*innen von einem Feedback rein mittels LED nicht vollständig überzeugt waren, mit einer Standardabweichung von 1,6 waren sich die Befragten hierzu auch einig. Es besteht Potential für zukünftige Arbeiten, die einerseits andere Kommunikationskanäle wie Vibration oder Piezosummer nutzen, oder aber auch eine Variation des LED-Feedbacks selbst, beispielsweise durch andere Farben oder Blinkfrequenzen. Beides ist mit den aktuellen Sensorknoten möglich, da in Vorbereitung weiterer Untersuchungen eben jene Ausgänge des Mikrocontrollers zur Ansteuerung zusätzlicher Peripherie herausgeführt wurden.

Die **Frage 1**, die letztlich eine erwartete Nützlichkeit für die Zielgruppe abfragt, wurde mit 7,1/10 Punkten deutlicher beantwortet, das Probandenkollektiv beantwortet die Frage also mit *stimme zu* oder *stimme voll und ganz zu*. Eine Kaufbereitschaft für ein solches System zeigt, dass die Anwendbarkeit für ältere Personen gesehen bzw. erwartet wird.

Ebenfalls Zustimmung herrscht bei **Frage 3** mit 7,6/10 Punkten: Die Testpersonen waren sich einig, dass sie durch das System die Reihenfolge der Kochschritte einfacher einhalten konnten. Dies ein Zeichen, dass die Funktion des Systems, eine Erinnerungsstütze zu sein, funktioniert und nicht zwingend ein Bedarf an komplexeren Feedbacksystemen vorliegt.

Die Proband*innen fühlten sich durch das System weitestgehend ungestört, entsprechend wurde **Frage 2** mit 8,1/10 Punkten bewertet. Die Sensoren sind also klein und unauffällig genug, so dass ihre Anbringung an handelsüblichen Kochtensilien zu keiner Beeinträchtigung beim Kochen führt.

Tatsächlich fühlten sich die Testpersonen so wenig gestört, dass sie mit **Frage 5** (8,6/10 Punkten) sogar den Wunsch nach mehr Sensorknoten ausdrückten. Dies könnte dazu führen, dass eine noch detaillierte Unterstützung beim Kochen möglich wird, allerdings steht dem die Notwendigkeit des Akkuladens bei einem Praxiseinsatz gegenüber. Müssten zu viele Sensorknoten einzeln geladen werden, schränkte dies die Usability wiederum stark ein - hierfür werden dann entsprechende Energiekonzepte oder neuartige Akkutechnologien benötigt, um die Laufzeit so zu erhöhen, dass dies nicht ins Gewicht fällt.

Schließlich hatten die Nutzenden nicht das Gefühl, durch das System abgelenkt zu werden, die **Frage 4** wurde entsprechend mit 9,5/10 Punkten bewertet.

8.3.3 Auswertung und Diskussion zum Raw Task Load Index

Wie bereits beim SUS-Score kochten die selben 13 Personen mit und sechs Personen ohne das Feedbacksystem, beide Fragebögen wurden wie bereits beschrieben direkt nach Versuchsdurchführung durch die Versuchspersonen ausgefüllt. Allgemein wurde der Raw Task Load Index-Fragebogen von allen 19 Personen ausgefüllt, in der Bewertung kann nun, wie in Tabelle 8.8 dargestellt, zwischen der Belastung von Personen mit und ohne System unterschieden werden.

Gruppe	μ	\tilde{x}	σ
Gesamt	25,0	15	21,3
Mit System	22,4	15	18,6
Ohne System	32,5	17,5	26,5

Tabelle 8.8: Vergleich des Mittelwerts μ , Medians \tilde{x} und der Standardabweichung σ des RLTX-Scores zwischen verschiedenen Testgruppen

Insgesamt ergab sich ein TLX-Score für diesen Versuchsaufbau von 25,0 im Durchschnitt, 15 im Median, mit einer Standardabweichung von 21,3. Bei den Personen, die ohne das System kochten ($n = 6$) wurde eine Gesamtarbeitslast von 32,5 errechnet, bzw. 17,5 im Median, mit einer Standardabweichung von 26,5. Personen ($n = 13$), die mit dem System kochten erzielten einen Belastungswert von 22,4 im Durchschnitt, 15 im Median, wobei die Standardabweichung 18,6 beträgt.

Nach der gebräuchlichen Klassifizierung des TLX-Scores (vgl. Kapitel 2.2.3), ergab sich damit insgesamt für die Proband*innen ohne Systeme eine *erhöhte* Belastung, wohingegen Personen, die mit dem in dieser Arbeit entwickelten System kochten, eine *mittlere* Arbeitslast dokumentierten. Schon auf den ersten Blick zeigt sich damit, dass das System dazu beitragen konnte den Kochvorgang zu erleichtern.

Analog zu der Auswertung des SUS-Scores folgt nun eine detaillierte Betrachtung

der einzelnen Arbeitslast-Kategorien, die Zusammenfassung ist in Tabelle 8.9 für die Gesamtkohorte dokumentiert und für die beiden Untergruppen, die mit bzw. ohne System kochten in Tabelle 8.11, respektive Tabelle 8.10.

Frage	μ	\tilde{x}	σ
1: Geistige Anforderung	32,6	25	20,8
2: Körperliche Anforderung	14,7	15	9,9
3: Zeitliche Anforderung	26,6	25	22,2
4: Leistung	23,1	15	22,6
5: Anstrengung	32,1	25	21,8
6: Frustration	21,0	5	24,2

Tabelle 8.9: Mittelwert μ , Median \tilde{x} und Standardabweichung σ der Einzelfragen des RTLX, Gesamtgruppe

Die Gesamtschau der Ergebnisse in Tabelle 8.9 zeigt, dass über alle Versuchsgruppen hinweg die geistige Anforderung für die gestellte Aufgabe mit einem Durchschnitt von 32,6 am höchsten war. Dies ist für eine Kochaufgabe ebenso zu erwarten gewesen, wie die Tatsache, dass die körperliche Anforderung mit 14,7 am niedrigsten bewertet wurde. Ebenso niedrig, mit 21,0, wurde die Frustration bewertet, die weiteren Fragen nach der zeitlichen Anforderung, der Leistung und der Anstrengung lagen mit 22,3 - 32,1 im mittleren Bereich. Es wird ebenso deutlich, dass das Probandenkollektiv insgesamt uneins in der Bewertung der Belastung war, erkennbar an den hohen Standardabweichungen. Einzelne Ausreißer führten hier zu einer starken Schwankungsbreite der Ergebnisse.

Diese Übersicht dient nur der Orientierung, es folgt nun eine detaillierte Betrachtung der Punktebewertung, getrennt nach den beiden Versuchsgruppen, die *mit* und *ohne System* kochten.

Bei den Personen, die ohne System kochten (Tabelle 8.10, $n = 6$) ergab sich im Durchschnitt eine geistige Anforderung von 42 (**Frage 1**), was nach typischer

Frage	μ	\tilde{x}	σ
1: Geistige Anforderung	42	35	24,4
2: Körperlische Anforderung	11	15	5,5
3: Zeitliche Anforderung	37	25	31,1
4: Leistung	17	15	7,6
5: Anstrengung	42	25	33,1
6: Frustration	46	65	31,3

Tabelle 8.10: Mittelwert μ , Median \tilde{x} und Standardabweichung σ der Einzelfragen des RTLX, Gruppe der Personen, die ohne System kochten

Skalierung bereits einem mittleren Bereich der *erhöhten* Last entspricht, die Standardabweichung war hier mit 24,4 allerdings auch groß. Es kann angenommen werden, dass bereits an dieser Stelle deutlich wird, dass der Versuchsaufbau, welcher ein Auswendiglernen des Rezepts erforderte, höhere geistige Anforderungen für einzelne Personen stellte.

Erwartungsgemäß liegt die körperliche Anforderung beim Kochen (**Frage 2**) eher niedrig mit einer *mittleren* Belastung von 11 und einer geringen Standardabweichung von 5,5, während die zeitliche Anforderung (**Frage 3**) bereits im *erhöhten* Bereich liegt, allerdings mit einer sehr hohen Standardabweichung, was für eine stark unterschiedliche Einschätzung zwischen den Proband*innen spricht ($\mu = 37$, $\sigma = 31,1$).

Die Kombination aus auswendig lernen und einem Kochprozess, der nicht pausiert werden kann, schlägt sich hier in der Bewertung nieder. Personen, die ohne System kochten, schätzten sich selbst so ein, dass sie die Ziele des Versuchs (**Frage 4**) gut erreichten, mit einem Durchschnitt von 17 und einer geringen Standardabweichung.

Die Anstrengung wurde in **Frage 5** mit 42 im Durchschnitt als *erhöht* bewertet, die Proband*innen beantworteten die Fragen jedoch sehr unterschiedlich ($\sigma = 33,1$). Das Frustrationspotential war mit 46 (**Frage 6**) sogar noch höher, aber auch hier gingen die Einschätzungen zwischen den Versuchspersonen stark auseinander, was sich in der Standardabweichung von 31,3 zeigt.

Insgesamt wird durch die hohe Standardabweichung deutlich, dass die Fragen von den Testpersonen sehr unterschiedlich beantwortet wurden. Einzelne Ausreißer führen hier zu großen Schwankungen, da die Probandenzahl mit sechs Personen klein war.

Frage	μ	\tilde{x}	σ
1: Geistige Anforderung	29,3	25	19,2
2: Körperliche Anforderung	16,1	15	11,0
3: Zeitliche Anforderung	22,9	22,5	18,0
4: Leistung	25,4	15	25,8
5: Anstrengung	28,6	25	16,5
6: Frustration	12,1	5	13,7

Tabelle 8.11: Mittelwert μ , Median \tilde{x} und Standardabweichung σ der Einzelfragen des RTLX, Gruppe der Personen, die mit System kochten

Die Gruppe der Proband*innen ($n = 13$), die mit dem in dieser Arbeit entwickelten Assistenzsystem kochten, beantworteten die Fragen ebenfalls, die Ergebnisse zu den Einzelfragen sind in Tabelle 8.11 dokumentiert.

Auch hier ist die geistige Anforderung erwartungsgemäß höher (**Frage 1**, $\mu = 29,3$, $\sigma = 19,2$) als die körperliche (**Frage 2**, $\mu = 16,1$, $\sigma = 11,0$). Die Proband*innen waren nur wenig frustriert bei der Durchführung (**Frage 6**, $\mu = 12,1$, $\sigma = 13,7$). Die Fragen nach der zeitlichen Anforderung (**Frage 3**) und der Leistung (**Frage 4**) lagen im Mittelfeld, mit einem Durchschnitt von 22,9 ($\sigma = 18,0$) für die zeitliche Belastung und einem Durchschnitt von 25,4 ($\sigma = 25,8$) für die Bewertung der Leistung.

Die Anstrengung (**Frage 5**) wurde mit einem Schnitt von 28,6 am zweithöchsten bewertet, bei einer Standardabweichung von 16,5.

Im Gesamten zeigt sich, dass die Standardabweichung in dieser Gruppe deutlich geringer ausfällt. Die höhere Probandenzahl von 13 Personen trägt erwartbar dazu bei, das potentielle Ausreißer weniger ins Gewicht fallen. Einerseits heißt das aber auch, dass sich die großen Schwankungsbreiten in der Gesamtgruppe hauptsächlich aus den Antworten der Personen, die ohne System kochten ergeben und nicht

durch die Personen, die mit System kochten hervorgerufen werden. Andererseits bedeutet dies auch, dass diese Gruppe sich einiger war, was die Einschätzung der Belastung mit System betrifft.

Frage	$\Delta\mu$	$\Delta\tilde{x}$
1: Geistige Anforderung	-12,7	-10
2: Körperliche Anforderung	+5,1	0
3: Zeitliche Anforderung	-14,1	-2,5
4: Leistung	+8,4	0
5: Anstrengung	-13,4	0
6: Frustration	-33,9	-60

Tabelle 8.12: Abweichungen der Mittelwerte $\Delta\mu$ und Mediane $\Delta\tilde{x}$ der einzelnen Fragen zwischen den Gruppen, die mit und ohne System kochten

Über die Differenz der Bewertungen zwischen den beiden Testgruppen kann nun die Auswirkung der Unterstützung durch das Smartpatchsystem abgeschätzt werden. Hierzu wurden die in Tabelle 8.12 verzeichneten Werte berechnet. Ein negativer Wert bedeutet eine *Verminderung der Anforderung*, d.h. eine geringere Belastung für Personen, die mit System kochten im Vergleich zu Proband*innen ohne Feedback. Entsprechende Werte wurden zur Übersicht grün hinterlegt. Ein positiver Wert gibt analog eine Vergrößerung der Anforderung für Personen mit System gegenüber Personen ohne System wider; diese Differenzen sind rot hinterlegt.

Die Frage nach der Leistung (**Frage 4**) wurde von den Personen mit Feedback als schlechter bewertet ($\Delta\mu = +8,4$, $\Delta\tilde{x} = 0$) als von der Gruppe ohne System. Die Frage zielt darauf ab, wie erfolgreich sich die jeweilige Person bei der Erreichung der vom Versuchsleiter oder der eigens gesteckten Ziele sieht. Die Personen mit System waren also unzufriedener mit ihrer Zielerreichung. Es kann vermutet werden, dass die Erwartungen an sich selbst höher sind, wenn durch den Versuchsaufbau bekannt ist, dass das System beim Kochen unterstützen soll. Diese Personen könnten dann dazu neigen, einen höheren Anspruch an das gekochte

Gericht und damit an sich selbst zu legen, weil letztlich immer noch die Person das Rezept eigenhändig kocht und das System nur als Erinnerungsstütze dient.

Die körperlichen Anforderungen (**Frage 2**) wurde ebenfalls etwas höher bewertet ($\Delta\mu = +5,1$, $\Delta\tilde{x} = 0$). Die Fragestellung beinhaltet die Einschätzung nach der körperlichen Aktivität, die benötigt wird bezüglich der Aufgabenerfüllung. Da das System selbst im Versuchsaufbau überhaupt keine körperliche Aktivität benötigt, weil die Sensoren bereits angebracht waren und das Gewicht eben dieser das Küchenutensil nicht signifikant verändert, ist davon auszugehen, dass die Frage rein mit Bezug auf den Kochvorgang selbst beantwortet wurde. Daher kann die Veränderung zwischen den Gruppen als Messrauschen betrachtet werden.

Der Mittelwert ($\Delta\mu = -12,7$) und Median ($\Delta\tilde{x} = -10$) der geistigen Belastung (**Frage 1**) konnte durch das Sensornetzwerk signifikant reduziert werden. Die Testpersonen empfanden durch die Unterstützung des Systems die Aufgaben als weniger anspruchsvoll. Die Erinnerungsstütze in Form eines visuellen Feedbacks genügte also, um damit die Belastung durch das Kochen eines unbekannten Rezepts, welches zuvor nur einmal vorgelesen wurde, deutlich zu mindern.

Ebenso konnte die zeitliche Anforderung (**Frage 3**) verringert werden ($\Delta\mu = -14,1$, $\Delta\tilde{x} = -2,5$), Proband*innen empfanden somit einen geringeren Zeitdruck, der Kochvorgang war für die Gruppe weniger hektisch.

Für Personen mit Assistenz durch das System war der Kochvorgang weniger anstrengend (**Frage 5**): Der Mittelwert sank um 13,4, der Median um 0. Die empfundene Anstrengung durch das auswendige Kochen ging folglich zurück.

Am deutlichsten sank die Frustration (**Frage 6**) die für Proband*innen, wenn sie mit dem System kochten, der Durchschnitt verringerte sich um 33,9, der Median um 60. Mit diesem Teilscore wird abgefragt, wie unsicher, entmutigt, irritiert, gestresst oder verärgert sich eine Person beim Versuch gefühlt hat. Es zeigt sich erwartbar, dass der Versuchsaufbau, bei dem Personen einmalig ein Rezept hören und es im Anschluss nachkochen müssen, hohen Frust auslöst. Personen, die jedoch Küchenutensilien mit Sensor nutzten, empfanden deutlich weniger Frustration, das System konnte die Aufgabe also erfolgreich erleichtern.

8.3.4 Sonstige Beobachtungen

Ein weiteres Maß für die Akzeptanz des Systems ist die Häufigkeit der Nachfragen. In der Gruppe der Personen, die mit Feedbacksystem kochten, kam es nur zu drei Nachfragen bezüglich der nächsten Schritte des Rezepts, während in der Gruppe ohne Feedbacksystem 13 Nachfragen vom Versuchsleiter vermerkt wurden. Dies bedeutet, dass Nachfragen in über 80% der Fälle aus der Gruppe der Personen stammten, die ohne Feedbacksystem kochten. Weiterhin ist zu beachten, dass die Gruppe der Personen mit Feedbacksystem ungefähr doppelt so groß war wie die Gruppe der Personen ohne Unterstützung. Dies lässt also insgesamt darauf schließen, dass das Feedbacksystem eine tatsächliche Unterstützung bietet, da die Testpersonen weniger Nachfragen stellen mussten, um ihre Aufgabe erfolgreich zu erledigen.

9 Zusammenfassung und Ausblick

Im vorangegangenen Kapitel konnte gezeigt werden, dass die Vorüberlegungen und Entwicklungen bezüglich der verschiedenen Teilsysteme korrekt waren. In einer Usability-Evaluation mit 19 Testpersonen mit einer Gesamtkochdauer von über 18 Stunden und unter Anwendung von SUS-, RTLX- und nicht-standardisierten Fragebögen wurde deutlich, dass die Anwendbarkeit und Nützlichkeit der Sensorknoten gegeben ist, das System nicht vom Kochvorgang ablenkt und das Feedback als hilfreich und zielführend wahrgenommen wird.

Abschließend folgt in diesem Kapitel nun die Zusammenfassung der Arbeit mit Rückbezug auf die zugrundeliegende Anforderungsanalyse, um einen Abgleich aus den zu Beginn postulierten Rahmenbedingungen mit den letztlich entstandenen Systemen durchzuführen. Außerdem wird ein Ausblick über mögliche Potentiale zu Erweiterung und Optimierung einzelner Komponenten des Systems gegeben.

9.1 Zusammenfassung

Die Gesellschaft altert weltweit immer weiter, was sich beispielsweise darin zeigt, dass der Anteil der Personen, die älter als 65 Jahren sind, bis 2050 auf mehr als 1,5 Milliarden geschätzt wird. Bei einer bis dahin erwarteten Gesamtbevölkerung von 9,6 Milliarden Menschen entspricht dies einem Anteil von ca. 15%. Einhergehend mit einer alternden Gesellschaft steigt auch die Belastung für das Gesundheitssystem, da die Wahrscheinlichkeit für Erkrankungen im Alter zunehmend steigt. Hieraus ergibt sich eine Überbelastung des vorhandenen medizinischen

und pflegerischen Fachpersonals, da immer mehr Personen von einer im Verhältnis schrumpfenden Anzahl an Fachkräften versorgt werden müssen. Darüber hinaus zeigt sich, dass die fachliche Qualität von Pflegefachkräften zunehmend sinkt, da durch den steigenden Bedarf Standards bei der Auswahl von geeignetem Personal gesenkt werden. Auf diese Weise wird eine angemessene und würdige Versorgung im Alter eine zunehmende Herausforderung. Die Europäische Union hat zur Milderung dieser Herausforderungen bereits Anfang dieses neuen Jahrtausends verschiedene Programme aufgelegt, um mittels Technologie Abhilfe zu schaffen.

Eines der erklärten Ziele der EU ist es hierbei zu ermöglichen, dass die pflegebedürftigen Personen sich möglichst lange in ihrer bevorzugten Umgebung, z.B. ihrer Wohnung, selbstständig versorgen können. Ein wesentlicher Bestandteil eines eigenständigen Lebens stellt gemäß den *Activities of Daily Living* die Nahrungsversorgung dar, wozu auch das Kochen von Mahlzeiten zählt. Ein System, welches die Notwendigkeit einer Betreuung bei der Nahrungszubereitung verringern kann, ermöglicht es somit unmittelbar durch einen geringeren Bedarf an Pflegesachleistungen die volkswirtschaftlichen Kosten zu senken.

Im Rahmen dieser Arbeit wurde deshalb der Leitfrage nachgegangen, auf welche Weise Technologie im Smart Home unterstützen kann, um die Autonomie von Senior*innen im Alltag zu verbessern. In diesem Kontext wurden auch weitere Fragen aufgeworfen: Wie kann KI beim Kochen unterstützen? Welche Anforderungen stellen sich dabei an die Hardware und inwiefern können die erforschten Methoden gestaltet werden, um einen Einsatz auch in anderen Domänen zu ermöglichen?

Zur Beantwortung dieser Fragen wurde ein System basierend auf den erhobenen Anforderungen A01 - A24 (Kapitel 5.2) entwickelt, welches die Signalstärke zwischen einzelnen Sensorknoten misst. Diese sind an verschiedenen Küchentensilien, beispielsweise einem Topf oder Messer, angebracht. Über Signalstärkemessungen während eines Kochvorgangs zwischen allen Sensorknoten kann so indirekt die Interaktion mit beziehungsweise zwischen den Gegenständen gemessen werden. Die Knoten sind Eigenentwicklungen auf Basis eines Bluetooth-System-on-Module, einer der Knoten dient dabei auch als Gateway zu einer Recheneinheit. Die Knoten wurden so entwickelt, dass eine möglichst lange Laufzeit

ohne Nachladen des Akkus möglich wird. Zudem verfügt jeder Knoten über eine Leuchtdiode, um den Nutzenden Informationen mitteilen zu können. Damit schließlich aus den Signalstärkemessungen auf einzelne Teilschritte im Kochprozess und das eigentliche Kochrezept geschlossen werden kann, wurden Methoden des maschinellen Lernens angewandt und hierfür ein Hintergrundsystem entwickelt. Dieses KI-System wird auf der zuvor erwähnten Recheneinheit ausgeführt und inferiert zur Laufzeit, in welchem Schritt sich die kochende Person befindet, was der nächste Teilschritt ist und welches Rezept gekocht wird.

Mit dem Wissen über den nächsten Teilschritt kann den Nutzenden ein Feedback in Form einer leuchtenden LED gegeben werden, um auf diese Weise als Erinnerungsstütze zu dienen. Im Zuge dieser Arbeit wurde dabei sowohl die Platine für die Sensorknoten entworfen und umgesetzt, die Sensorknotensoftware zur Ablaufsteuerung und Kommunikation entwickelt, als auch die Architektur für die KI-Modelle implementiert. Darüber hinaus wurden die verschiedenen KI-Modelle auf ihre Tauglichkeit hin untersucht.

Zur Evaluation der Entwicklungen haben insgesamt 19 Personen jeweils drei verschiedene Rezepte gekocht, zusammen wurden demnach also 57 Kochprozesse durchgeführt, die Gesamtmessdauer beträgt ca. 18,5 Stunden. Von diesen 19 Testenden kochten 13 mit dem hier entwickelten Sensornetzwerk, sie bildeten die Testgruppe, und sechs ohne das Feedbacksystem, diese ergaben die Kontrollgruppe.

Die zu kochenden Rezepte waren den Proband*innen vor der Testung nicht bekannt und wurden ihnen vor jedem Durchlauf einmal vorgelesen, während den Durchläufen musste dann jedes Rezept auswendig gekocht werden. Im Anschluss wurde zur Bewertung von jeder Person in der Testgruppe der System Usability Scale (SUS) Fragebogen und von allen Testpersonen der Raw Task Load Index (RTLX) Fragebogen ausgefüllt. Die Auswertung dieser Untersuchungsbögen er gab einen SUS-Score von 84,3 im Durchschnitt, bzw. von 85,3 für Personen, die täglich kochen. Die Usability des Systems, d.h. die Benutzerfreundlichkeit und Anwendungstauglichkeit, kann damit basierend auf dem SUS-Score nach üblicher Bewertungsskala als *gut bis exzellent* bezeichnet werden.

Die Auswertung ergab weiterhin einen RTLX-Score von 22,4 im Durchschnitt für

Personen, die mit System kochten und 32,5 im Durchschnitt für Personen, die ohne System kochten. Der RTLX zeigt damit eine Verminderung der Belastung für Proband*innen, die mit System kochten, ausgehend von einer *erhöhten* Belastung, hin zu einer *mittleren* Arbeitslast, gemessen am Mittelwert des Gesamtscores. Die entwickelten Sensorknoten erfüllen alle Anforderungen A01 - A10 hinsichtlich ihrer möglichen Betriebszeit von über einer Woche, Lademöglichkeiten über einen USB-Anschluss, Abmessungen von weniger als 25 mm x 25 mm x 1 5mm, Kosten von weniger als 10€ und Einsatzmöglichkeiten eines Feedbacksystems mittels LED.

Die Kommunikation zwischen den Sensorknoten erfolgt drahtlos und über die eingesetzte standardkonforme Funktechnologie kann ebensfalls eine Abstandsbestimmung durchgeführt werden, den Anforderungen A11 - A13 konnte somit ebenfalls nachgekommen werden.

Die Anforderungen A14 - A17 beziehen sich auf das KI-System, welches eine nutzerunabhängige Klassifizierung ermöglicht, sowohl den aktuellen Teilschritt erkennt als auch den nächsten vorhersagen kann und ebenfalls in der Lage ist, das Gesamtzept zu prädizieren. Diesen Anforderungen wurde ebenso entsprochen. Für das Feedbacksystem wurden die Anforderungen A18 - A21 aufgestellt, welche eine einfache Wahrnehmung, eine rechtzeitige Anzeige und eine Erinnerungsfunktion fordern; auch ihnen wurde vollständig Rechnung getragen.

Den sonstigen Anforderungen A22 - A24 wurde ebenfalls genüge getan, hier wurde eine hilfreiche Bewertung von den Nutzenden und ein Feedback zur Laufzeit gefordert.

Zusammenfassend kann somit im Hinblick auf die vorliegende Arbeit gesagt werden, dass es mittels eines Sensornetzwerks möglich ist, Personen individuelle Unterstützung zu geben, um einen Kochprozess zu erleichtern. Vor allem im Kontext von Gemeinschaftsküchen in Pflegeeinrichtungen, aber auch in privaten Seniorenwohnungen bietet dies vielfältige Potentiale.

9.2 Ausblick

In der vorgelegten Arbeit wurde die prinzipielle Umsetzbarkeit eines Sensorsystems demonstriert, welches einen Kochvorgang KI-gestützt begleiten kann und der kochende Person durch individuelle Rückmeldungen Hilfestellung bietet. Für eine umfassende und zielgerichtete Unterstützung bedarf es jedoch noch weiterer Forschung, vor allem im Hinblick auf das Feedbacksystem. Im aktuellen System wurde eine RGB-LED integriert und *grün* als Farbe verwendet. Bei einem produktiven Einsatz im Umfeld von Senior*innen muss jedoch untersucht werden, ob dies die optimale Wahl ist, sowohl im Hinblick auf die Farbe selbst, als auch auf die Art des Feedbackkanals. So ist es denkbar, statt oder zusätzlich zum visuellen Feedback auch weitere Informationskanäle zu nutzen, beispielsweise über Piezo-Signalgeber oder Vibrationsmotoren. Gerade hinsichtlich auf die oftmals nachlassende Sehkraft im Alter stellt dies eine wichtige Erweiterung dar. In diesem Zusammenhang ist auch eine Gesamtsystemevaluation mit Senioren und Seniorinnen wichtig, um beispielsweise weitere Verbesserung für die Benutzerfreundlichkeit und Gebrauchstauglichkeit im Alltag erkennen und implementieren zu können.

Für eine zukünftige Weiterentwicklung der Sensorknoten sind auch weitere Optimierungen der Energieeffizienz und des Energiemanagements wünschenswert. Eine große Herausforderung für die Praxistauglichkeit eines sensorbasierten Systems stellt die Notwendigkeit der Wiederaufladung der Sensorakkus dar. Zwar konnte in dieser Arbeit bereits eine für einen Prototypen sehr lange Laufzeit von neun Tagen erreicht werden, für einen praktikablen Einsatz sollte dieser Zeitraum aber soweit wie möglich ausgedehnt werden. Darüber hinaus gilt es zu untersuchen, in wie weit der Ladevorgang selbst vereinfacht werden kann, beispielsweise durch induktives Laden oder Magnete, die die Sensoren in der Schublade über Federkontakte mit einem Netzteil verbinden.

Weiteres Optimierungspotential besteht in der Auswahl und dem Training der KI-Algorithmen der modularen Backendarchitektur. In der Vorabtestung zeigte sich zwar bereits, dass es zielführend ist, einzelne Schritte durch unterschiedliche Algorithmen zu inferieren, allerdings ist es möglich, dass sich bei Verfügbarkeit

von größeren Datensätzen für das Training der jeweiligen Modelle auch andere Algorithmen als vorteilhafter erwiesen hätten. Hier gilt es, mit einem variatenreichen, größeren Datensatz abermals die optimalen KI-Algorithmen für die einzelnen Module zu bestimmen.

A Anhang

A.1 Testrezepte

Käseomelette

Zutaten: Zwei Eier, Käse, Butter

Utensilien: Pfanne, Reibe, Salzstreuer, Pfannenwender

1. Stelle die Pfanne auf den Herd
2. Schlage die Eier in eine Schüssel auf
3. Füge Salz hinzu
4. Reibe etwas Käse darüber
5. Mische die Zutaten in der Schüssel
6. Gib etwas Butter in die Pfanne
7. Erhitze die Pfanne auf Herdstufe 7 bis die Butter schäumt
8. Verteile die Butter in der Pfanne, indem du die Pfanne bewegst oder den Pfannenwender benutzt
9. Gib die Eiermischung in die Pfanne und nutze den Pfannenwender während des Kochens
10. Wenn du möchtest, kannst du das Omelette nach deinen Vorlieben formen oder verrühren
11. Serviere, sobald du möchtest

Kartoffelrösti

Zutaten: Drei geschälte, mittelgroße Kartoffeln, ein Ei, Stärke, Öl

Utensilien: Pfanne, Reibe, Salzstreuer, Pfannenwender

1. Stelle die Pfanne auf den Herd
2. Reibe die Kartoffeln in eine Schüssel
3. Presse das Wasser auf den Kartoffeln und gib sie in eine andere Schüssel
4. Füge ein Ei, Salz und einen Esslöffel Stärke hinzu
5. Vermenge die Masse vorsichtig mit dem Esslöffel
6. Gib einen Esslöffel Öl in die Pfanne und erhitze sie auf Herdstufe 7
7. Verteile das Öl in der Pfanne, indem du die Pfanne bewegst oder den Pfannenwender benutzt
8. Gib etwas Kartoffelmasse in die Pfanne, um zu erkennen wann sie heiß ist.
Füge die restliche Masse hinzu sobald sie heiß ist.
9. Drücke die Kartoffelmasse mit dem Pfannenwender flach und teile sie in Viertel
10. Wende die Rösti, sobald sie nach deinen Wünschen braun sind
11. Füge einen weiteren Esslöffel Öl in die Pfanne hinzu
12. Serviere die Rösti, sobald die andere Seite nach deinen Wünschen braun ist

Gebratene Pilze

Zutaten: 130g Pilze, Öl

Utensilien: Pfanne, Messer, Salzstreuer, Pfannenwender

1. Stelle die Pfanne auf den Herd
2. Schneide die Pilze in Scheiben
3. Gib einen Esslöffel Öl in die Pfanne und erhitze sie auf Herdstufe 7
4. Verteile das Öl in der Pfanne, indem du die Pfanne bewegst oder den Pfannenwender benutzt
5. Gib etwas Pilz in die Pfanne um zu erkennen, wenn sie heiß ist. Füge die restlichen Scheiben hinzu sobald sie heiß ist.
6. Salze die Pilze
7. Nutze den Pfannenwender, um die Pilze während des Bratens zu bewegen
8. Serviere, sobald zu möchtest

Gebratene Paprika mit Zwiebeln

Zutaten: Eine Paprika, eine Zwiebel, Öl

Utensilien: Pfanne, Messer, Salzstreuer, Pfannenwender

1. Stelle die Pfanne auf den Herd
2. Schneide die Zwiebeln und anschließend die Paprika in Streifen
3. Gib einen Esslöffel Öl in die Pfanne und erhitze sie auf Herdstufe 7
4. Verteile das Öl in der Pfanne, indem du die Pfanne bewegst oder den Pfannenwender benutzt
5. Gib etwas Zwiebel oder Paprika in die Pfanne um zu erkennen, wenn sie heiß ist. Füge das restliche Gemüse hinzu sobald sie heiß ist.

6. Salze das Gemüse
7. Nutze den Pfannenwender, um das Gemüse während des Bratens zu bewegen
8. Serviere, sobald zu möchtest

Bratkartoffeln

Zutaten: Drei geschälte Kartoffeln, Öl

Utensilien: Pfanne, Messer, Salzstreuer, Pfannenwender

1. Stelle die Pfanne auf den Herd
2. Schneide die Kartoffeln in ca. 5 - 10 mm dicke Scheiben
3. Gib einen Esslöffel Öl in die Pfanne und erhitze sie auf Herdstufe 7
4. Verteile das Öl in der Pfanne, indem du die Pfanne bewegst oder den Pfannenwender benutzt
5. Gib etwas Kartoffel in die Pfanne um zu erkennen, wenn sie heiß ist. Füge die restlichen Kartoffeln hinzu sobald sie heiß ist.
6. Verteile die Kartoffeln so, dass eine einzelne Schicht in der Pfanne ist
7. Salze das Gemüse
8. Nutze den Pfannenwender, um die Kartoffeln zu wenden, sobald sie nach deinen Wünschen braun sind
9. Stelle den Herd auf Stufe 8
10. Serviere die Bratkartoffeln, sobald die andere Seite nach deinen Wünschen braun ist

Nudeln kochen

Zutaten: Nudeln, Soße

Utensilien: Topf, Salzstreuer, Löffel

1. Fülle den Topf mit 1L warmen Wasser und stelle ihn auf den Herd
2. Füge einen Teelöffel Salz hinzu
3. Stelle den Herd auf Stufe 9 bis das Wasser kocht, reduziere die Hitze dann auf Stufe 6
4. Füge die Nudeln hinzu und rühre mit dem Löffel um
5. Koche die Nudeln bis zur gewünschten Härte, rühre zwischendurch mit dem Löffel um (ca. 11 Minuten)
6. Hole die Nudeln mit dem Löffel aus dem Topf und serviere sie mit der Soße

Gekochte Kartoffeln

Zutaten: Drei Kartoffeln

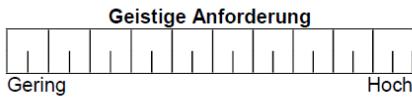
Utensilien: Topf, Salzstreuer, Löffel

1. Fülle den Topf mit 1L warmen Wasser und stelle ihn auf den Herd
2. Stelle den Herd auf Stufe 9 bis das Wasser kocht, reduziere die Hitze dann auf Stufe 6
3. Füge die Kartoffeln hinzu und rühre mit dem Löffel um
4. Koche die Kartoffeln bis zur gewünschten Härte, rühre zwischendurch mit dem Löffel um (ca. 15 - 30 Minuten)
5. Hole die Kartoffeln mit dem Löffel aus dem Topf und serviere sie

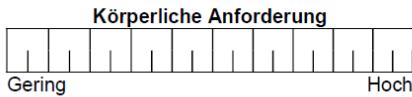
A.2 SUS Fragebogen

	Stimme überhaupt nicht zu	Stimme nicht zu	Stimme zu	Stimme voll und ganz zu
Ich kann mir sehr gut vorstellen, das System regelmäßig zu nutzen.				
Ich empfinde das System als unnötig komplex.				
Ich empfinde das System als einfach zu nutzen.				
Ich denke, dass ich technischen Support brauchen würde, um das System zu nutzen.				
Ich finde, dass die verschiedenen Funktionen des Systems gut integriert sind.				
Ich finde, dass es im System zu viele Inkonsistenzen gibt.				
Ich kann mir vorstellen, dass die meisten Leute das System schnell zu beherrschen lernen.				
Ich empfinde die Bedienung als sehr umständlich.				
Ich habe mich bei der Nutzung des Systems sehr sicher gefühlt.				
Ich musste eine Menge Dinge lernen, bevor ich mit dem System arbeiten konnte.				

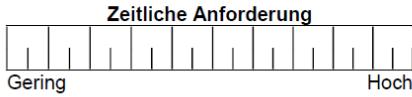
A.3 RTLX Fragebogen



Wie viel geistige Anforderung war bei der Informationsaufnahme und bei der Informationsverarbeitung erforderlich (z.B. Denken, Entscheiden, Rechnen, Erinnern, Hinsehen, Suchen ...)? War die Aufgabe leicht oder anspruchsvoll, einfach oder komplex, erfordert sie hohe Genauigkeit oder ist sie fehler tolerant?



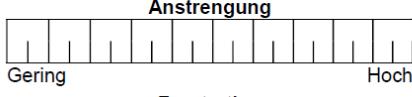
Wie viel körperliche Aktivität war erforderlich (z.B. ziehen, drücken, drehen, steuern, aktivieren ...)? War die Aufgabe leicht oder schwer, einfach oder anstrengend, erholsam oder mühselig?



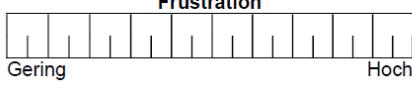
Wie viel Zeitdruck empfanden Sie hinsichtlich der Häufigkeit oder dem Takt mit dem die Aufgaben oder Aufgabenelemente auftraten? War die Aufgabe langsam und geruhsam oder schnell und hektisch?



Wie erfolgreich haben Sie Ihrer Meinung nach die vom Versuchsleiter (oder Ihnen selbst) gesetzten Ziele erreicht? Wie zufrieden waren Sie mit Ihrer Leistung bei der Verfolgung dieser Ziele?



Wie hart müssten Sie arbeiten, um Ihren Grad an Aufgabenerfüllung zu erreichen?



Wie unsicher, entmutigt, irritiert, gestresst und verärgert (versus sicher, bestätigt, zufrieden, entspannt und zufrieden mit sich selbst) fühlten Sie sich während der Aufgabe?

A.4 Nicht-standardisierter Fragebogen

	Stimme überhaupt nicht zu	Stimme nicht zu	Stimme zu	Stimme voll und ganz zu
Ich kann mir vorstellen, z.B. meinen Großeltern das System zu kaufen.				
Das System hat mich beim Kochen gestört.				
Durch das System konnte ich die richtige Reihenfolge leichter einhalten.				
Die LEDs haben mich abgelenkt.				
Mehr Utensilien sollten mit Sensorknoten ausgestattet sein.				
Eine andere Art von Feedback, z.B. Sprache oder Vibration, stelle ich mir hilfreicher vor.				

Abbildungsverzeichnis

2.1	Übersetzung des SUS-Scores	14
2.2	Schematischer Aufbau eines MEMS-Beschleunigungssensors	20
2.3	Netzwerktopologien in Sensornetzwerken	24
2.4	Lokalisierungsmethoden in Sensornetzwerken	25
2.5	Schematische Darstellung eines Entscheidungsbaumes	31
2.6	Schematische Darstellung einer SVM-Klassifizierung	33
2.7	Schematische Darstellung eines MLP	34
2.8	Schematische Darstellung einer LSTM-Zelle	36
3.1	Vicon Blue Trident	39
3.2	Das Tempo Armband der Firma CarePredict	39
3.3	Universaltransmitter von Steute	41
3.4	Feedbacksystem mittels LEDs von Basharudin	42
3.5	Versuchsaufbau von Kosch et al.	43
3.6	Ausschnitt aus der Generierung des HARUS-Datensets	45
3.7	Einzelbild aus dem CADMMA-Datensatz	46
3.8	Einzelbild bei verschiedenen Lichtbedingungen aus dem DHA-Datensatz	47
3.9	Darstellung der automatisch generierten Dictionaries	49
3.10	Die farblich unterscheidbaren Utensilien zur Zubereitung der Rezepte	49
3.11	Die in Zou et al. untersuchten Räume mit Positionen der WLAN-Router	50
3.12	Beschleunigungen beim Schneiden in Pham et al.	51
3.13	<i>It's LiFe!</i> Smartphone und Aktivitätssensor von Weegen et al.	54
5.1	Gesamtsystemübersicht	65
6.1	Schaltplan der Strom- und Laderegelung	77

6.2	Schaltplan des Beschleunigungssensors und der RGB-LED	79
6.3	Schaltplan der Chipantenne und PCB-Antenne	80
6.4	Beschaltung des Mikrocontrollers nRF52840	82
6.5	PCB-Design der minimalen Smartpatch Lite Variante	83
6.6	PCB-Design der regulären Smartpatch Variante	83
6.7	Vorderseite der beiden Smartpatch-Varianten	84
6.8	Rückseite der beiden Smartpatch-Varianten	84
6.9	Ablaufdiagramm der Sensorknotensoftware	86
6.10	Befehlsweitergabe für Feedback	88
6.11	RSSI-Messung zwischen zwei Sensorknoten an verschiedenen Orten .	90
6.12	Energieverbrauchsmessung des Smartpatch	91
7.1	Datenfluss zwischen den KI-Modulen	98
7.2	Konfusionsmatrix für die Testergebnisse des SubstepClassifier Moduls	106
7.3	Konfusionsmatrix für die Testergebnisse des ProcessClassifier Moduls	107
7.4	Spektrum des Beschleunigungssignals in x-Achse eines beispielhaften Kartoffelschnitts	109
7.5	Konfusionsmatrix des eingebetteten Random Forest Modells	116
8.1	Die Sensoren an verschiedenen Utensilien	120

Tabellenverzeichnis

2.1	Die Einteilung der Arbeitslast gemäß TLX-Score, basierend auf [51]	16
2.2	Darstellung einer Konfusionsmatrix	28
6.1	Gegenüberstellung der verschiedenen distanzbasierten Lokalisierungsmethoden	70
6.2	Vergleich der verschiedenen Feedbackmöglichkeiten	72
6.3	Vergleich der verschiedenen SoCs	74
6.4	Beispiel einer RSSI Messungen, alle Werte in dBm	89
7.1	Teilfunktionalitäten und ihre Bedeutung für die Gesamtarchitektur . .	95
7.2	Definition der Teilschritte und ihre Enkodierung	96
7.3	Mögliche Algorithmen für die Module	99
7.4	Definition der Testrezepte und ihre Enkodierung	100
7.5	Kriterien für die Auswahl der Prätestrezepte	100
7.6	Ergebnisse für die möglichen Algorithmen für das SubstepClassifier Modul	101
7.7	Ergebnisse für die möglichen Algorithmen für das SubstepInferencer Modul	102
7.8	Ergebnisse für die möglichen Algorithmen für das ProcessClassifier Modul	102
7.9	Ausgewählte Algorithmen für die einzelnen Module	103
7.10	Ergebnisse der einzelnen Module mit optimalem Algorithmus auf einem Validierungsdatensatz	104
7.11	Ergebnisse für die möglichen Algorithmen bei einer Abtastrate von 500 Hz	111

7.12	Ergebnisse für die möglichen Algorithmen bei einer Abtastrate von 25 Hz	112
7.13	Die 25 wichtigsten Features in absteigender Relevanz	113
7.14	Ergebnisse des eingebetteten Random Forest Modells für verschiedene Abtastraten	114
7.15	Ergebnisse des eingebetteten Random Forest Modells für verschiedenes Gemüse	115
8.1	Schritte des ersten Rezepts aus Systemsicht	123
8.2	Schritte des zweiten Rezepts aus Systemsicht	125
8.3	Schritte des dritten Rezepts aus Systemsicht	127
8.4	Vergleich des Mittelwerts μ , Medians \tilde{x} und der Standardabweichung σ des SUS-Scores zwischen verschiedenen Testgruppen	129
8.5	Mittelwert μ , Median \tilde{x} und Standardabweichung σ der Einzelfragen des SUS	130
8.6	Vergleich des Mittelwerts μ , Medians \tilde{x} und der Standardabweichung σ des Scorings für den nicht-standardisierten Fragebogen zwischen verschiedenen Testgruppen	133
8.7	Mittelwert μ , Median \tilde{x} und Standardabweichung σ der Einzelfragen des nicht-standardisierten Fragebogens	134
8.8	Vergleich des Mittelwerts μ , Medians \tilde{x} und der Standardabweichung σ des RLTX-Scores zwischen verschiedenen Testgruppen	136
8.9	Mittelwert μ , Median \tilde{x} und Standardabweichung σ der Einzelfragen des RLTX, Gesamtgruppe	137
8.10	Mittelwert μ , Median \tilde{x} und Standardabweichung σ der Einzelfragen des RLTX, Gruppe der Personen, die ohne System kochten	138
8.11	Mittelwert μ , Median \tilde{x} und Standardabweichung σ der Einzelfragen des RLTX, Gruppe der Personen, die mit System kochten .	139
8.12	Abweichungen der Mittelwerte $\Delta\mu$ und Mediane $\Delta\tilde{x}$ der einzelnen Fragen zwischen den Gruppen, die mit und ohne System kochten	140

Eigene Veröffentlichungen

Konferenzbeiträge

- [1] Gergely Biri, Dennis Birkenmaier, Marc Schroth, Tao Hu, Maximilian Hoffmann, Marco Giurgiu, Simon Woll, and Wilhelm Stork. Predicting affective states using wearable technology: A machine learning approach. In *2024 IEEE International Workshop on Sport, Technology and Research (STAR)*, pages 199–204. IEEE, July 2024. doi: [10.1109/star62027.2024.10635909](https://doi.org/10.1109/star62027.2024.10635909).
- [2] Jens Juhl, Korbinian F. Rudolf, Lukas Kohout, Marc B. Schroth, Friedrich Gauger, Christoph Zimmermann, and Wilhelm Stork. Using augmented reality and artificial intelligence for an efficient and safe preparation of individual drug assortments in nursing homes. In *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECC-ME)*, pages 1–6, 2022. doi: [10.1109/ICECCM55909.2022.9988015](https://doi.org/10.1109/ICECCM55909.2022.9988015).
- [3] Arne Rönnau, Jakob Weinland, Marc Schroth, Christoph Zimmermann, Robert Klebbe, Nicole Strutz, Luis Perotti, Michael Maier, Felix Messmer, and Benjamin Maidel. AuRorA - Interaktive Roboter unterstützen im Smart Home. *Autonome Roboter für Assistenzfunktionen: Interaktive Grundfertigkeiten-Ergebnisse und Forschungsperspektiven*, 2020.
- [4] Johannes Schneider, Marc Schroth, Maik Holzhey, Timon Blöcher, and Wilhelm Stork. An approach to improve impedance plethysmography on the wrist by using adaptive feedback control. In *2017 IEEE Sensors Applications Symposium (SAS)*, pages 1–6, 2017. doi: [10.1109/SAS.2017.7894063](https://doi.org/10.1109/SAS.2017.7894063).

- [5] Johannes Schneider, Marc Schroth, Jorg Ottenbacher, and Wilhelm Stork. A novel wearable sensor device for continuous monitoring of cardiac activity during sleep. In *2018 IEEE Sensors Applications Symposium (SAS)*. IEEE, mar 2018. doi: 10.1109/sas.2018.8336725.
- [6] Marc Schroth, Timuçin Etkin, and Wilhelm Stork. A novel approach for human activity recognition using object interactions and machine learning. In *2021 IEEE Sensors Applications Symposium (SAS)*, pages 1–6, 2021. doi: 10.1109/SAS51076.2021.9530029.
- [7] Marc Schroth, Andreas Ilg, Lukas Kohout, and Wilhelm Stork. A method for designing an embedded human activity recognition system for a kitchen use case based on machine learning. In *2022 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pages 249–254, 2022. doi: 10.1109/IAICT55358.2022.9887452.
- [8] Marc Schroth, Dennis Birkenmaier, and Wilhelm Stork. Towards a modular machine learning architecture for human activity recognition. In *2024 IEEE Sensors Applications Symposium (SAS)*, pages 1–6. IEEE, July 2024. doi: 10.1109/sas60918.2024.10636463.

Literaturverzeichnis

- [1] DIN IEC 60050-351:2014-09, Internationales Elektrotechnisches Wörterbuch - Teil 351: Leittechnik (IEC 60050-351:2013).
- [2] Ageing well in the information society: The Ambient Assisted Living (AAL) Programme | EUR-Lex, abgerufen Oktober 2024. URL <https://eur-lex.europa.eu/legal-content/EL/ALL/?uri=LEGISSUM:em0006>.
- [3] § 36 SGB 11 - Einzelnorm, abgerufen Oktober 2024. URL https://www.gesetze-im-internet.de/sgb_11/_36.html.
- [4] Maryam Abbasi, Goncalo Sarmento, Mickael Mota, Pedro Martins, Filipe Sa, and Filipe Cardoso. Comparing user experience: An analysis of usability testing methods for mobile applications. In *2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–10. IEEE, June 2023. doi: 10.23919/cisti58278.2023.10211987.
- [5] Mohammad Afaneh. *Intro to bluetooth low energy*. Novel Bits, Fishers, 2018. ISBN 1790198151.
- [6] Charu C. Aggarwal. *Data classification*. Number Volume 35 in Chapman & Hall. CRC Press, Taylor & Francis Group, Boca Raton, Florida, online-ausg. edition, 2015. ISBN 9781466586741. Includes bibliographical references. - Description based on print version record.
- [7] Joan Aranda and Manuel Vinagre. Anticipating human activities from object interaction cues. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, volume 21, pages 58–63. IEEE, August 2016. doi: 10.1109/roman.2016.7745091.

- [8] Mario N. Armenise, Caterina Ciminelli, Francesco Dell’Olio, and Vittorio M. N. Passaro. *Advances in Gyroscope Technologies*. Springer Berlin Heidelberg, 2011. ISBN 9783642154942. doi: 10.1007/978-3-642-15494-2.
- [9] Katrin Arning and Martina Ziefle. “*Get that Camera Out of My House!*” *Conjoint Measurement of Preferences for Video-Based Healthcare Monitoring Systems in Private and Public Places*, pages 152–164. Springer International Publishing, 2015. ISBN 9783319193120. doi: 10.1007/978-3-319-19312-0_13.
- [10] Sofiane Atek, Filippo Bianchini, Corrado De Vito, Vincenzo Cardinale, Simone Novelli, Cristiano Pesaresi, Marco Eugeni, Massimo Mecella, Antonello Rescio, Luca Petronzio, Aldo Vincenzi, Pasquale Pistillo, Gianfranco Giusto, Giorgio Pasquali, Domenico Alvaro, Paolo Villari, Marco Mancini, and Paolo Gaudenzi. A predictive decision support system for coronavirus disease 2019 response management and medical logistic planning. *DIGITAL HEALTH*, 9, January 2023. ISSN 2055-2076. doi: 10.1177/20552076231185475.
- [11] Aaron Bangor, Phil Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *J. Usability Stud.*, 4:114–123, 04 2009.
- [12] N.W. Basharudin, K.N.F. Ku Azir, A. M. Khairuddin, and P. Ehkan. Ambient cues of kitchen counter in guiding cooking activities for alzheimer’s patient. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, volume 23, pages 375–378. IEEE, March 2019. doi: 10.1109/percomw.2019.8730829.
- [13] Indronil Bhattacharjee and Pryonti Bhattacharja. Stock price prediction: A comparative study between traditional statistical approach and machine learning approach. In *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*. IEEE, December 2019. doi: 10.1109/eict48899.2019.9068850.

- [14] Ralf Bohnsack, Winfried Marotzki, and Michael Meuser, editors. *Hauptbegriffe qualitativer Sozialforschung*. Number 8226 in UTB. Verlag Barbara Budrich, Opladen, 3., durchgesehene auflage edition, 2011. ISBN 9783825282264. Literaturverzeichnis: Seite 169-196.
- [15] Harrington M. Pass L. Reisner E. Bookman, A. *Family Caregiver Handbook*. Cambridge, MA: Massachusetts Institute of Technology., 2007.
- [16] T. Boren and J. Ramey. Thinking aloud: reconciling theory and practice. *IEEE Transactions on Professional Communication*, 43(3):261–278, 2000. ISSN 0361-1434. doi: 10.1109/47.867942.
- [17] Bosch Sensortec. Bma400 3-axes ultra-low power accelerometer, February 2023. URL <https://www.bosch-sensortec.com/media/boschsenortec/downloads/datasheets/bst-bma400-ds000.pdf>.
- [18] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT92. ACM, July 1992. doi: 10.1145/130385.130401.
- [19] Bruno Bouchard, Kevin Bouchard, and Abdenour Bouzouane. A smart cooking device for assisting cognitively impaired users. *Journal of Reliable Intelligent Environments*, 6(2):107–125, April 2020. ISSN 2199-4676. doi: 10.1007/s40860-020-00104-3.
- [20] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996. ISSN 1573-0565. doi: 10.1007/bf00058655.
- [21] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification And Regression Trees*. Routledge, October 2017. ISBN 9781315139470. doi: 10.1201/9781315139470.
- [22] John Brooke. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189, 11 1995.

- [23] Hans-Bernd Brosius. *Methoden der empirischen Kommunikationsforschung*. Springer eBook Collection. Springer VS, Wiesbaden, 8., vollständig überarbeitete und erweiterte auflage edition, 2022. ISBN 9783658341954.
- [24] Martin Bujard and Gunnar Andersson. Fertility declines near the end of the covid-19 pandemic: Evidence of the 2022 birth declines in germany and sweden. *European Journal of Population*, 40(1), January 2024. ISSN 1572-9885. doi: 10.1007/s10680-023-09689-w.
- [25] Bundesministerium für Gesundheit. Pflegeversicherung, Zahlen und Fakten, July 2024. URL <https://www.bundesgesundheitsministerium.de/themen/pflege/pflegeversicherung-zahlen-und-fakten.html>.
- [26] Chiara Buratti, Andrea Conti, Davide Dardari, and Roberto Verdone. An overview on wireless sensor networks technology and evolution. *Sensors*, 9(9):6869–6896, August 2009. ISSN 1424-8220. doi: 10.3390/s90906869.
- [27] James C. Byers, Alvah C. Bittner, and Susan G. Hill. Traditional and raw task load index (tlx) correlations: Are paired comparisons necessary? in a. 1989. URL <https://api.semanticscholar.org/CorpusID:58504578>.
- [28] Oswald Campesato. *Artificial Intelligence, Machine Learning, and Deep Learning*. Mercury Learning and Information, Dulles, VA, 2020. ISBN 9781683924654.
- [29] CarePredict. Actionable Insights to Improve Senior Home Care | CarePredict Benefits, 8 2023. URL <https://www.carepredict.com/home-care/>.
- [30] Pai-Hsuen Chen, Chih-Jen Lin, and Bernhard Schölkopf. A tutorial on v-support vector machines. *Applied Stochastic Models in Business and Industry*, 21(2):111–136, March 2005. ISSN 1526-4025. doi: 10.1002/asmb.537.
- [31] Long Cheng, Chengdong Wu, Yunzhou Zhang, Hao Wu, Mengxin Li, and Carsten Maple. A survey of localization in wireless sensor network. *International Journal of Distributed Sensor Networks*, 8(12):962523, December 2012. ISSN 1550-1477. doi: 10.1155/2012/962523.

- [32] Alexey Ya. Chervonenkis. *Early History of Support Vector Machines*, pages 13–20. Springer Berlin Heidelberg, 2013. ISBN 9783642411366. doi: 10.1007/978-3-642-41136-6_3.
- [33] Kaare Christensen, Gabriele Doblhammer, Roland Rau, and James W Vaupel. Ageing populations: the challenges ahead. *The Lancet*, 374(9696):1196–1208, October 2009. ISSN 0140-6736. doi: 10.1016/s0140-6736(09)61460-4.
- [34] Diane J. Cook, Juan C. Augusto, and Vikramaditya R. Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4):277–298, August 2009. ISSN 1574-1192. doi: 10.1016/j.pmcj.2009.04.001.
- [35] W Cottingham and D Greenwood. *Electromagnetic waves in a vacuum*, pages 45–53. Cambridge University Press, Cambridge, 1991.
- [36] Diodes Incorporated. Ap2112 600ma cmos ldo regulator with enable, 2017. URL <https://predictabledesigns.com/review-of-bluetooth-low-energy-ble-solutions/>.
- [37] Joseph S. Dumas, Janice C. Redish, and Janice Redish. *A practical guide to usability testing*. Human/computer interaction. Ablex Pub. Corp, Norwood, NJ, 2nd print. edition, 1994. ISBN 0893919918. Includes bibliographical references (p. 375-381) and indexes.
- [38] Peter F. Edemekong, Deb L. Bomgaars, Sukesh Sukumaran, and Caroline Schoo. Statpearls. January 2024.
- [39] Espressif Systems. Esp32-c6 series. URL https://www.espressif.com/sites/default/files/documentation/esp32-c6_datasheet_en.pdf.
- [40] G. Fenza, D. Furno, and V. Loia. Hybrid approach for context-aware service discovery in healthcare domain. *Journal of Computer and System Sciences*, 78(4):1232–1247, July 2012. ISSN 0022-0000. doi: 10.1016/j.jcss.2011.10.011.

- [41] Kraig Finstad. The usability metric for user experience. *Interacting with Computers*, 22(5):323–327, September 2010. ISSN 0953-5438. doi: 10.1016/j.intcom.2010.04.004.
- [42] Carlos Flores-Vazquez and Joan Aranda. Human activity recognition from object interaction in domestic scenarios. In *2016 IEEE Ecuador Technical Chapters Meeting (ETCM)*. IEEE, October 2016. doi: 10.1109/etcm.2016.7750844.
- [43] Giles M. Foody. Challenges in the real world use of classification accuracy metrics: From recall and precision to the matthews correlation coefficient. *PLOS ONE*, 18(10):e0291908, October 2023. ISSN 1932-6203. doi: 10.1371/journal.pone.0291908.
- [44] Jörg Frochte. *Maschinelles Lernen*. Hanser eLibrary. Hanser, München, 2., aktualisierte auflage edition, 2019. ISBN 3446459979.
- [45] Qi Fu, Lina Liu, and Shiwei Ma. Human action recognition based on self-learned key frames and features extraction. In *2017 Chinese Automation Congress (CAC)*, volume 1, pages 3498–3502. IEEE, October 2017. doi: 10.1109/cac.2017.8243386.
- [46] Peter Georgieff. Ambient assisted living. *Marktpotenziale IT-unterstützter Pflege für ein selbstbestimmtes Altern, FAZIT Forschungsbericht*, 17:9–10, 2008.
- [47] F.A. Gers. Learning to forget: continual prediction with lstm. In *9th International Conference on Artificial Neural Networks: ICANN '99*, volume 1999, pages 850–855. IEE, 1999. doi: 10.1049/cp:19991218.
- [48] Google. Wie zeichne ich meine Aktivität mit meinem Fitbit-Gerät auf? - Fitbit-Hilfe, abgerufen November 2024. URL <https://support.google.com/fitbit/answer/14236510?hl=de&zippy=%2Cwelche-aktivit%C3%A4ten-zeichnet-mein-fitbit-ger%C3%A4t-%C3%A4t-automatisch-auf>.

- [49] Carla Graf. The lawton instrumental activities of daily living scale. *AJN, American Journal of Nursing*, 108(4):52–62, April 2008. ISSN 0002-936X. doi: 10.1097/01.naj.0000314810.46029.74.
- [50] Sandra G. Hart. Nasa-task load index (nasa-tlx); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9):904–908, October 2006. ISSN 1071-1813. doi: 10.1177/154193120605000909.
- [51] Sandra G. Hart and Lowell E. Staveland. *Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research*, pages 139–183. Elsevier, 1988. doi: 10.1016/s0166-4115(08)62386-9.
- [52] Keith C. Hendy, Kevin M. Hamilton, and Lois N. Landry. Measuring subjective workload: When is one scale better than many? *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 35(4):579–601, December 1993. ISSN 1547-8181. doi: 10.1177/001872089303500401.
- [53] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997. ISSN 1530-888X. doi: 10.1162/neco.1997.9.8.1735.
- [54] Zawar Hussain, Michael Sheng, and Wei Emma Zhang. Different approaches for human activity recognition: A survey. *Journal of Network and Computer Applications*, 167:102738, October 2019. ISSN 1084-8045. doi: 10.1016/j.jnca.2020.102738.
- [55] Marina Iacovou, Deanna C Pattieson, Helen Truby, and Claire Palermo. Social health and nutrition impacts of community kitchens: a systematic review. *Public Health Nutrition*, 16(3):535–543, June 2012. ISSN 1475-2727. doi: 10.1017/s1368980012002753.
- [56] Infineon Technologies AG. Psoc 4 mcu with airoc bluetooth le. URL https://www.infineon.com/dgdl/Infineon-PSoC_4_4200_BLE_Family_Datasheet_Programmable_System-on-Chip-DataSheet-v03_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0ee5f2a26ddd.

- [57] Johanson Technology. 2.4 ghz mini chip antenna for 802.11, wifi, wlan, iot with small area and gnd size, ultra low profile, 2017. URL <https://www.johansontechnology.com/datasheets/2450AT14A0100/2450AT14A0100.pdf>.
- [58] Satish R. Jondhale, R. Maheswar, and Jaime Lloret. *Received Signal Strength Based Target Localization and Tracking Using Wireless Sensor Networks*. Springer International Publishing, 2022. ISBN 9783030740610. doi: 10.1007/978-3-030-74061-0.
- [59] Davide Anguita Jorge Reyes-Ortiz. Human activity recognition using smartphones, 2013.
- [60] Davide Anguita Jorge Reyes-Ortiz. Smartphone-based recognition of human activities and postural transitions, 2015.
- [61] Sidney Katz. Assessing self-maintenance: Activities of daily living, mobility, and instrumental activities of daily living. *Journal of the American Geriatrics Society*, 31(12):721–727, December 1983. ISSN 1532-5415. doi: 10.1111/j.1532-5415.1983.tb03391.x.
- [62] V. Kecman. *Support Vector Machines – An Introduction*, pages 1–47. Springer Berlin Heidelberg, April 2005. ISBN 9783540323846. doi: 10.1007/10984697_1.
- [63] Kingbright. Aptf1616lseezgkqbkc 1.6 x 1.6 mm full-color surface mount led, April 2023. URL <https://www.kingbrightusa.com/images/catalog/SPEC/APTF1616LSEEZGKQBKC.pdf>.
- [64] Jurek Kirakowski. The software usability measurement inventory: background and usage. *Usability evaluation in industry*, pages 169–178, 1996.
- [65] Thomas Kleinberger, Martin Becker, Eric Ras, Andreas Holzinger, and Paul Müller. *Ambient Intelligence in Assisted Living: Enable Elderly People to Handle Future Interfaces*, pages 103–112. Springer Berlin Heidelberg, 2007. ISBN 9783540732815. doi: 10.1007/978-3-540-73281-5_11.

- [66] Thomas Kosch, Kevin Wennrich, Daniel Topp, Marcel Muntzinger, and Albrecht Schmidt. The digital cooking coach: using visual and auditory in-situ instructions to assist cognitively impaired during cooking. In *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '19. ACM, June 2019. doi: 10.1145/3316782.3321524.
- [67] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190, November 2006. ISSN 1573-7462. doi: 10.1007/s10462-007-9052-3.
- [68] Sotiris Kotsiantis. Supervised machine learning: A review of classification techniques. *Informatica* 31, pages 249–268, 2007.
- [69] Satish Kumar, Kanika Sharma, and Kamal Kumar Gola. *A Review on Localization Algorithm for Wireless Sensor Networks*, pages 23–35. Springer Singapore, 2021. ISBN 9789813348622. doi: 10.1007/978-981-33-4862-2_3.
- [70] Miguel A. Labrador and Pedro M. Wightman. *Topology Control in Wireless Sensor Networks*. SpringerLink. Springer Netherlands, Dordrecht, 2009. ISBN 9781402095856.
- [71] Paula Lago, Shingo Takeda, Kohei Adachi, and Sozo Inoue. Cooking activity dataset with macro and micro activities, 2020.
- [72] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 1476-4687. doi: 10.1038/nature14539.
- [73] Felix Levinzon. *Piezoelectric Accelerometers with Integral Electronics*. Springer International Publishing, 2015. ISBN 9783319080789. doi: 10.1007/978-3-319-08078-9.
- [74] Curt A. Levis. *Radiowave propagation*. Wiley, Hoboken, N.J., 2010. Includes bibliographical references and index. - Description based on print version record.

- [75] James R Lewis. Psychometric evaluation of the post-study system usability questionnaire: The pssuq. In *Proceedings of the human factors society annual meeting*, volume 36, pages 1259–1260. Sage Publications Sage CA: Los Angeles, CA, 1992.
- [76] James R. Lewis. Usability testing, March 2012.
- [77] Ping Li, Ramy Meziane, Martin J.-D. Otis, Hassan Ezzaidi, and Philippe Cardou. A smart safety helmet using imu and eeg sensors for worker fatigue detection. In *2014 IEEE International Symposium on Robotic and Sensors Environments (ROSE) Proceedings*, pages 55–60. IEEE, October 2014. doi: 10.1109/rose.2014.6952983.
- [78] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 140:1–55, 1932.
- [79] Yan-Ching Lin, Min-Chun Hu, Wen-Huang Cheng, Yung-Huan Hsieh, and Hong-Ming Chen. Human action recognition and retrieval using sole depth information. In *Proceedings of the 20th ACM international conference on Multimedia*, MM ’12. ACM, October 2012. doi: 10.1145/2393347.2396381.
- [80] Huan Liu. *Feature Engineering for Machine Learning and Data Analytics*. CRC Press, March 2018. ISBN 9781315181080. doi: 10.1201/9781315181080.
- [81] YILI LIU and CHRISTOPHER D. WICKENS. Mental workload and cognitive task automaticity: an evaluation of subjective and time estimation metrics. *Ergonomics*, 37(11):1843–1854, November 1994. ISSN 1366-5847. doi: 10.1080/00140139408964953.
- [82] LOGEMAS. Getting Started with Vicon Blue Trident IMUs - Logemas, 8 2021. URL <https://logemas.com/knowledge-base/faq/getting-started-with-vicon-blue-trident-imus/>.
- [83] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge Univ. Press, Cambridge [u.a.],

- reprinted edition, 2009. ISBN 9780521865715. Includes bibliographical references and index.
- [84] Inocencio Maramba, Arunangsu Chatterjee, and Craig Newman. Methods of usability testing in the development of ehealth applications: A scoping review. *International Journal of Medical Informatics*, 126:95–104, June 2019. ISSN 1386-5056. doi: 10.1016/j.ijmedinf.2019.03.018.
 - [85] Prof. Dr. Michael Wagner Marcella Reissmann. *Werthaltungen hochaltriger Menschen und ihre Wünsche für die eigene Lebenssituation und das gesellschaftliche Zusammenleben*. Number 9. Cologne Center for Ethics, Rights, Economics, and Social Sciences of Health (ceres), July 2022.
 - [86] Stephen Marsland. *Machine learning*. A Chapman & Hall book. CRC Press/Taylor & Francis, Boca Raton, Fla., second edition edition, 2015. ISBN 9781466583283.
 - [87] James Clerk Maxwell. *Philosophical Transactions of the Royal Society of London*, 155:459–512, December 1865. ISSN 2053-9223. doi: 10.1098/rstl.1865.0008.
 - [88] John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI Magazine*, 27(4):12, Dec. 2006. doi: 10.1609/aimag.v27i4.1904. URL <https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1904>.
 - [89] Microchip Technology. Atbtlc1000-qfn ultra-low power ble atbtlc1000-qfn data sheet. URL <https://ww1.microchip.com/downloads/en/DeviceDoc/Ultra-Low-Power-BLE-ATBTLCL1000-QFN-Data-Sheet-DS70005391A.pdf>.
 - [90] Microchip Technology. Mcp73831/2 miniature single-cell, fully integrated li-ion, li-polymer charge management controllers, 2020. URL <https://predictabledesigns.com/review-of-bluetooth-low-energy-ble-solutions/>.

- [91] Puneet Misra and Arun Singh Yadav. Improving the classification accuracy using recursive feature elimination with cross-validation. *Int. J. Emerg. Technol.*, 11(3):659–665, 2020.
- [92] Andreas Christian Müller. *Einführung in Machine Learning mit Python*. O'Reilly, Heidelberg, 1. auflage edition, 2017. ISBN 9783960101116. Description based on publisher supplied metadata and other sources.
- [93] Lawrence Mosley. *A balanced approach to the multi-class imbalance problem*. PhD thesis, Iowa State University.
- [94] Gholamreza Nakhaeizadeh. *Data Mining: Theoretische Aspekte und Anwendungen*. Physica-Verlag HD, 1998. ISBN 9783642860942. doi: 10.1007/978-3-642-86094-2.
- [95] Hamit Taner Ünal and Fatih Başçiftçi. Evolutionary design of neural network architectures: a review of three decades of research. *Artificial Intelligence Review*, 55(3):1723–1802, July 2021. ISSN 1573-7462. doi: 10.1007/s10462-021-10049-5.
- [96] Newton, Isaac. *Philosophiae Naturalis Principia Mathematica. Auctore Js. Newton ... Jussu Societatis Regiae ac Typis Josephi Streater ...*, 1687. doi: 10.3931/E-RARA-440.
- [97] Ha Duong Ngo. *MEMS Accelerometers*. MDPI - Multidisciplinary Digital Publishing Institute, [Erscheinungsort nicht ermittelbar], 2019. ISBN 9783038974147.
- [98] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 0125184050.
- [99] Ruixin Niu, Aditya Vempaty, and Pramod K. Varshney. Received-signal-strength-based localization in wireless sensor networks. *Proceedings of the IEEE*, 106(7):1166–1182, July 2018. ISSN 1558-2256. doi: 10.1109/jproc.2018.2828858.

- [100] Jon Nordby. emlearn: Machine learning inference engine for microcontrollers and embedded devices, 2019.
- [101] Nordic Semiconductor. nrf52840 product specification v1.1,. URL https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.1.pdf.
- [102] Nordic Semiconductor. Power profiler kit ii v1.0.1,. URL https://infocenter.nordicsemi.com/pdf/PPK_2_User_Guide_20201201.pdf.
- [103] Henry Friday Nweke, Ying Wah Teh, Mohammed Ali Al-garadi, and Uzoma Rita Alo. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications*, 105:233–261, September 2018. ISSN 0957-4174. doi: 10.1016/j.eswa.2018.03.056.
- [104] Nikolaos A. Pantazis, Stefanos A. Nikolidakis, and Dimitrios D. Vergados. Energy-efficient routing protocols in wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials*, 15(2):551–591, 2013. ISSN 1553-877X. doi: 10.1109/surv.2012.062612.00084.
- [105] Kaiqiao Peng, Yuliang Zhao, Xiaopeng Sha, Wenqian Ma, Yufan Wang, and Wen J. Li. Accurate recognition of volleyball motion based on fusion of mems inertial measurement unit and video analytic. In *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, July 2018. doi: 10.1109/cyber.2018.8688133.
- [106] Cuong Pham and Patrick Olivier. *Slice&Dice: Recognizing Food Preparation Activities Using Embedded Accelerometers*, pages 34–43. Springer Berlin Heidelberg, 2009. ISBN 9783642054082. doi: 10.1007/978-3-642-05408-2_4.
- [107] Jonny B Pornel and Giabelle A Saldana. Four common misuses of the likert scale. *Philippine Journal of Social Sciences and Humanities*, 18(2):12–19, 2013.

- [108] David M. W. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *International Journal of Machine Learning Technology* 2:1 (2011), pp.37-63, October 2020. doi: 10.48550/ARXIV.2010.16061.
- [109] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, March 1986. ISSN 1573-0565. doi: 10.1007/bf00116251.
- [110] J. Ross Quinlan. *C4.5*. Ebrary online. Morgan Kaufmann Publishers, San Mateo, California, 2014. ISBN 9780080500584. Dateiformat Volltext: PDF.
- [111] Renesas Electronics. Da14580 bluetooth low energy 4.2 soc. URL <https://www.renesas.com/us/en/document/dst/da14580-datasheet?r=1600281>.
- [112] C. J. Rijsbergen. *Information retrieval*. Butterworth, London [u.a.], 2. ed., repr edition, 1981. ISBN 0408709294.
- [113] RKI. Warum ist das Tragen von Uhren und Schmuck durch Beschäftigte im Gesundheitswesen nicht zulässig?, abgerufen Oktober 2024. URL https://www.rki.de/SharedDocs/FAQ/Krankenhaushyg/Schmuck_Piercing/FAQ_01.html.
- [114] Susana Rubio, Eva Díaz, Jesús Martín, and José M. Puente. Evaluation of subjective mental workload: A comparison of swat, nasa-tlx, and workload profile methods. *Applied Psychology*, 53(1):61–86, December 2003. ISSN 1464-0597. doi: 10.1111/j.1464-0597.2004.00161.x.
- [115] Maliwan Rueankam, Supalak Khemthong, Winai Chatthong, Tharadon Rodkaeo, and Sasicha Chanworavit. Cooking groups for older adults: Implications for functional abilities. *Activities, Adaptation & Aging*, 47(4):535–549, November 2022. ISSN 1544-4368. doi: 10.1080/01924788.2022.2148425.
- [116] Artem Saakian. *Radio wave propagation fundamentals*. Artech House radar series. Artech House, Boston, second edition edition, 2021. ISBN 9781630818449.

- [117] Fariba Sadri. Ambient intelligence: A survey. *ACM Computing Surveys*, 43(4):1–66, October 2011. ISSN 1557-7341. doi: 10.1145/1978802.1978815.
- [118] Jeff Sauro. *Quantifying the user experience*. Morgan Kaufmann, [Place of publication not identified], 2016. ISBN 9780128025482. Includes bibliographical references and index.
- [119] Jeff Sauro and James R. Lewis. Correlations among prototypical usability metrics: evidence for the construct of usability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’09. ACM, April 2009. doi: 10.1145/1518701.1518947.
- [120] Martin Sauter. *From GSM to LTE*. Wiley, Chichester, West Sussex, U.K., 2011. ISBN 9780470667118. Includes bibliographical references and index.
- [121] Bertram Scheufele and Ines Engelmann. *Empirische Kommunikationsforschung*. UTB basics. UVK-Verlagsges., Konstanz, 2009. ISBN 9783825242756. ISBN 978-3-8252-4275-6 ersch. mit 2 weiteren Bdn. als Studien-Starter-Pack UTB. Kommunikationswissenschaft ; 4275.
- [122] John S. Seybold. *Introduction to RF propagation*. Wiley, Hoboken, N.J., 2005. ISBN 9780471743682. Includes bibliographical references and index.
- [123] C.E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, January 1949. ISSN 0096-8390. doi: 10.1109/jrproc.1949.232969.
- [124] Adrian J. Shepherd. *Multi-Layer Perceptron Training*, pages 1–22. Springer London, 1997. ISBN 9781447109532. doi: 10.1007/978-1-4471-0953-2_1.
- [125] Zhou Shuchang. A survey on human action recognition. December 2023. doi: 10.48550/ARXIV.2301.06082.
- [126] Johannes Siegl. Pflegesachleistungen: Anspruch, Höhe und Antrag, 10 2024. URL <https://www.pflege.de/pflegekasse-pflegefinanzierung/pflegeleistungen/pflegesachleistungen/>.

- [127] Petros Spachos. Raw rssI dataset for indoor localization fingerprinting, 2020.
- [128] Sunil Srinivasa and Martin Haenggi. Path loss exponent estimation in a large field of interferers. February 2008. doi: 10.48550/ARXIV.0802.0351.
- [129] steute. RF 96 ST-NET. URL <https://www.steute-leantec.com/en/system/sensors-and-actors/universal-transmitter-rf-96-st-net>.
- [130] Chun-Fang Su, Li-Chen Fu, Yi-Wei Chien, and Ting-Ying Li. Activity recognition system for dementia in smart homes based on wearable sensor data. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, volume 14, pages 463–469. IEEE, November 2018. doi: 10.1109/ssci.2018.8628688.
- [131] Hong Sun, Vincenzo De Florio, Ning Gui, and Chris Blondia. Promises and challenges of ambient assisted living systems. In *2009 Sixth International Conference on Information Technology: New Generations*. IEEE, 2009. doi: 10.1109/itng.2009.169.
- [132] Qiao-yan Sun, Xue-jiao Liu, Yu-mei Sun, Mei-chun Wang, Xiang Han, and Xiang-guang Chen. A security wireless monitoring and automatic protection system for ccel. *Wireless Communications and Mobile Computing*, 2021(1), January 2021. ISSN 1530-8677. doi: 10.1155/2021/6652246.
- [133] Shan Suthaharan. *Machine learning models and algorithms for big data classification*. Number volume 36 in Integrated series in information systems. Springer, New York, NY, 2016. ISBN 9781489976413. Description based on publisher supplied metadata and other sources.
- [134] John Teel. Review of bluetooth low energy (ble) solutions (updated 2022), 2022. URL <https://predictabledesigns.com/review-of-bluetooth-low-energy-ble-solutions/>.
- [135] Texas Instruments. Cc2650 simplelink multistandard wireless mcu. URL <https://www.ti.com/lit/ds/symlink/cc2650.pdf>.

- [136] David Tse. *Fundamentals of wireless communication*. Cambridge University Press, Cambridge, 2012. ISBN 9780511807213. Title from publisher's bibliographic system (viewed on 05 Oct 2015).
- [137] United Nations Department of Economic and Social Affairs Population Division. World population prospects 2024. URL <https://population.un.org/wpp/Graphs/Probabilistic/POP/65plus/900>.
- [138] van der Weegen, Renée Verwey, Huibert Tange, Marieke Spreeuwenberg, and Luc de Witte. Usability testing of a monitoring and feedback tool to stimulate physical activity. *Patient Preference and Adherence*, page 311, March 2014. ISSN 1177-889X. doi: 10.2147/ppa.s57961.
- [139] Ruth Vang-Mata, editor. *Multilayer perceptrons*. Computer science, technology and applications. Nova Science Publishers, Hauppauge, New York, 2020. ISBN 9781536173659. Includes bibliographical references and index.
- [140] Renee Verwey, Sanne Van der Weegen, Huibert Tange, Marieke Spreeuwenberg, Trudy Van der Weijden, and Luc De Witte. Get moving: the practice nurse is watching you! *Journal of Innovation in Health Informatics*, 20(4): 289–298, September 2013. ISSN 2058-4555. doi: 10.14236/jhi.v20i4.19.
- [141] Michalis Vrigkas, Christophoros Nikou, and Ioannis A. Kakadiaris. A review of human activity recognition methods. *Frontiers in Robotics and AI*, 2, November 2015. ISSN 2296-9144. doi: 10.3389/frobt.2015.00028.
- [142] Jin-Mao Wei, Xiao-Jie Yuan, Qing-Hua Hu, and Shu-Qin Wang. A novel measure for evaluating classifiers. *Expert Systems with Applications*, 37(5): 3799–3809, May 2010. ISSN 0957-4174. doi: 10.1016/j.eswa.2009.11.040.
- [143] Cynthia Williams. *CURRENT Diagnosis & Treatment in Family Medicine*, chapter Healthy Aging & Assessing Older Adults. McGraw-Hill Education Ltd, 3e edition, 2010. ISBN 0071624368.
- [144] Don Williams, Gavin Kelly, and Lisa Anderson. Msn 9: new user-centered desirability methods produce compelling visual design. In *CHI '04 Extended*

- Abstracts on Human Factors in Computing Systems*, CHI04, pages 959–974. ACM, April 2004. doi: 10.1145/985921.985952.
- [145] Eva Wolfangel. *Mehr Gesundheit für eine Gesellschaft des langen Lebens*. Robert Bosch Stiftung, Stuttgart, 2021. ISBN 3939574694.
- [146] Alice Zheng. *Feature engineering for machine learning*. O'Reilly, Beijing, 2018. ISBN 9781491953242. Revision history for the first edition: 2018-03-23: first release.
- [147] Martina Ziefle, Simon Himmel, and Wiktoria Wilkowska. *When Your Living Space Knows What You Do: Acceptance of Medical Home Monitoring by Different Technologies*, pages 607–624. Springer Berlin Heidelberg, 2011. ISBN 9783642253645. doi: 10.1007/978-3-642-25364-5_43.
- [148] Han Zou, Yuxun Zhou, Jianfei Yang, Hao Jiang, Lihua Xie, and Costas J. Spanos. Deepsense: Device-free human activity recognition via autoencoder long-term recurrent convolutional network. In *2018 IEEE International Conference on Communications (ICC)*. IEEE, May 2018. doi: 10.1109/icc.2018.8422895.
- [149] Moshe Zviran, Chanan Glezer, and Itay Avni. User satisfaction from commercial web sites: The effect of design and use. *Information & Management*, 43(2):157–178, March 2006. ISSN 0378-7206. doi: 10.1016/j.im.2005.04.002.