

# MASTERARBEIT

## MACHINE LEARNING ZUR PRÄEMPTIVEN STEUERUNG IM SMART HOME

ausgeführt am



Studiengang

Informationstechnologien und Wirtschaftsinformatik

Von: Stefan Raminger

Personenkennzeichen: 1910320041

Graz, am 19. März 2021

.....  
Unterschrift

## **EHRENWÖRTLICHE ERKLÄRUNG**

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

.....

Unterschrift

*“As people’s expectations of what technology can do for them are changing, the vision of what a smart home entails is continuously evolving as well.” (Mennicken et al., 2014, S. 105)*

## **DANKSAGUNG**

An dieser Stelle möchte ich mich herzlich bei meinem Betreuer Dipl.-Ing. (FH) Günther Zwetti für seine Geduld mit mir und seiner fachlichen Unterstützung bedanken. Der größte Dank gilt jedoch meiner Verlobten, ohne ihr motivierendes Zutun wäre diese Arbeit niemals zustande gekommen. Weiter möchte ich mich bei meiner Familie, meinen Freunden und meinen Studienkollegen für ihre tatkräftige Unterstützung bei der Erreichung dieses Meilensteins danken.

## KURZFASSUNG

Smart Services sind Teil des globalen Megatrends der digitalen Transformation. Seit circa 2015 ermöglichen Smart Services, dass Unternehmen Dienstleistungen erbringen, bereits bevor Kund/innen den Bedarf für diese erkennen. Dies führt zu einer Effizienz- und Effektivitätssteigerung für alle Beteiligten, wodurch wiederum die Zufriedenheit aller gesteigert wird.

Die Ursprünge von Smart Homes reichen hingen bis in die 1960er Jahre zurück. Bereits damals wurden reaktive und proaktive Steuerungsmöglichkeiten für das „Wired Home“ aufgezeigt. Der hohe Energie- und Ressourcenbedarf wecken bei immer mehr Menschen das Bedürfnis, die Effektivität und die Effizienz ihres Lebensraumes zu steigern. Trotz der großen Weiterentwicklungen im Bereich des Internet of Things und dem Machine Learning, nennen Benutzer/innen ihren Lebensraum bereits smart, wenn sie diesen aus der Ferne steuern können.

Diese Arbeit überträgt die Implikationen des Begriffs „smart“ aus der Grundfunktionalität eines Smart Services auf Smart Homes. Damit soll die Frage beantwortet werden, ob mit Machine Learning eine präemptive Steuerung eines Smart Homes möglich ist.

Hierzu werden unterschiedliche Algorithmen verwendet, um mit einem Datensatz aus einer Laborumgebung Machine Learning Modelle zu erstellen. Diese werden auf Basis mehrerer Metriken miteinander verglichen, um festzustellen, ob es möglich ist, damit eine präemptive Steuerung für ein Smart Home umzusetzen.

## **ABSTRACT**

Smart services are part of the global megatrend of digital transformation. Since around 2015, smart services have enabled companies to initiate the execution of services even before customers recognise the need for them. This leads to an increase in efficiency and effectiveness for all parties involved, which in turn increases everyone's satisfaction.

The origins of smart homes date back all the way to the 1960s. Even then, reactive and proactive control options for the "wired home" were identified. The high demand for energy and resources is making more and more people want to increase the effectiveness and efficiency of their living space. Despite the great developments in the field of the Internet of Things and machine learning, users already call their living space smart when they can control it remotely.

This work transfers the implications of the term "smart" from the basic functionality of a smart service to smart homes. The aim is to answer the question of whether pre-emptive control of a smart home is possible with machine learning.

For this purpose, different algorithms are used to create machine learning models with a data set from a laboratory environment. These are compared with each other based on several metrics to determine whether it is possible to implement pre-emptive control for a smart home.

# INHALTSVERZEICHNIS

<b>1</b>	<b>EINLEITUNG.....</b>	<b>1</b>
1.1	Ausgangssituation .....	1
1.2	Zielsetzung.....	2
1.3	Vorgehensweise .....	2
1.4	Struktur .....	3
<b>2</b>	<b>GRUNDLAGEN UND DEFINITIONEN.....</b>	<b>4</b>
2.1	Ubiquitous Computing .....	4
2.2	Smart Home .....	5
2.2.1	Definition des Smart Home.....	7
2.2.2	Geschichte und Zukunft des Smart Homes.....	11
2.2.3	Komponenten des Smart Homes.....	14
2.2.4	Herausforderungen des Smart Homes .....	18
2.3	Smart Services .....	20
2.4	Machine Learning .....	24
2.5	Grundlegende Zusammenhänge zwischen Smart Home und Smart Service .....	30
<b>3</b>	<b>UNTERSUCHUNGSDESIGN .....</b>	<b>32</b>
3.1	Ziel empirischer Softwareentwicklung .....	32
3.2	Methodik .....	35
3.2.1	Geschäftsverständnis.....	38
3.2.2	Datenverständnis .....	41
3.2.3	Datenvorbereitung .....	46
3.2.4	Modellierung .....	48
3.2.5	Evaluierung.....	52
3.3	Informationen zur Hardware, Tools und Software.....	52
<b>4</b>	<b>IMPLEMENTIERUNG UND VALIDIERUNG .....</b>	<b>54</b>
4.1	Datenaufbereitung und -analyse .....	54
4.2	Modellanalyse .....	57

4.2.1	Hyperparameter Tuning .....	58
4.2.2	Evaluierung der Modelle.....	61
<b>5</b>	<b>CONCLUSIO.....</b>	<b>65</b>
	<b>ABKÜRZUNGSVERZEICHNIS .....</b>	<b>67</b>
	<b>ABBILDUNGSVERZEICHNIS .....</b>	<b>68</b>
	<b>TABELLENVERZEICHNIS .....</b>	<b>70</b>
	<b>LISTINGS .....</b>	<b>71</b>
	<b>LITERATURVERZEICHNIS .....</b>	<b>72</b>

# 1 EINLEITUNG

Es zeigt sich, dass für Menschen der Wunsch nach Erleichterung im Alltag, mehr Komfort und besserer Effizienz steigt (Begleitforschung Mittelstand-Digital, 2016). Aus diesem Grund werden im privaten Umfeld Smart Homes und im Dienstleistungskontext Smart Services eingesetzt. Bei Smart Services wird vermehrt ein präemptives Vorgehen angestrebt, wohingegen Smart Homes noch in reaktivem oder proaktivem Kontext stehen. Für die präemptive Steuerung werden Machine Learning Algorithmen eingesetzt, daher wird in dieser Arbeit geprüft, wie eine präemptive Steuerung in Smart Homes integriert werden kann.

## 1.1 Ausgangssituation

Smart Services und Smart Homes zeigen in deren Vorgehen Überschneidungen. Dies zeigt sich besonders im Vergleich der Vorstufe des Smart Services, dem Smart Space. Hierbei verfolgten Smart Spaces den gleichen reaktiven Charakter wie Smart Homes, und sollten eine Automatisierung ermöglichen. Smart Spaces wurden dabei im Unternehmenskontext zu Smart Services weiterentwickelt. Diese werden von Allmendinger und Lombreglia (2005, S. 132) wie folgt beschrieben:

*“For customers, smart services create an entirely new kind of value—the value of removing unpleasant surprises from their lives.”*

Sie beschreiben dabei eine Dienstleistung, die präemptiv anstelle von reaktiv oder proaktiv ist. Die Erbringung eines Smart Service wird ihrer Definition nach eingeleitet, sobald es Anzeichen gibt, dass das Ereignis bevorsteht (Allmendinger & Lombreglia, 2005).

Die Smart Home Association beschreibt Smart Homes hingegen lediglich als die Integration von Technologie und Diensten durch Heimvernetzung zum Erreichen einer besseren Lebensqualität (Roe, 2007). Wenn der Begriff Smart Home wörtlich übersetzt wird, so erwarten sich die Experten ein intelligentes Zuhause. Wie die Beschreibung der Smart Home Association zeigt, handelt es sich jedoch primär um Vernetzung der Technologien in einem Zuhause und der damit verbundenen Automatisierungsmöglichkeit, die einen reaktiven oder proaktiven Charakter hat.

Aufgrund der Überschneidung der ursprünglichen Form des Smart Services und dem Smart Home gilt somit der Rückschluss, dass auch das Smart Home sich zu einer präemptiven Form weiterentwickeln kann, um zu einem autonomen und hochentwickelten System erweitert zu werden.

## 1.2 Zielsetzung

Aufgrund der Zusammenhänge des Smart Services, des Smart Homes und unter Berücksichtigung der Definition von Allmendinger und Lombreglia (2005) für Smart Services, wird im Rahmen dieser Arbeit geprüft, wie die Steuerung eines Smart Homes präemptiv gestaltet werden kann. Für die präemptive Steuerung von Smart Services wird Machine Learning eingesetzt. Daher soll ein vergleichbarer Ansatz für Smart Homes geprüft werden. Dementsprechend ist die Zielsetzung, einen Prototyp zu erstellen, der mittels Machine Learning die Steuerung von Aktoren durchführt, sobald es Anzeichen gibt, dass ein Ereignis eintreten wird. Dieses Machine Learning Modell wird für einen Prototypen vorbereitet, und anhand von Daten aus einem Smart Home genutzt. Daraus ergibt sich folgende Forschungsfrage für diese Arbeit:

„Wie und in welchem Umfang kann Machine Learning zur präemptiven Steuerung in ein Smart Home integriert werden?“

Dazu soll ein konkreter Prozess zur methodischen Vorgehensweise erarbeitet werden. Dies soll es ermöglichen, dass mittels Data-Science-Ansatz Machine Learning Modelle für einen Prototypen evaluiert werden. Dabei wird auch auf die Wahl eines Datensatzes geachtet, der für das Training der Machine Learning Algorithmen genutzt wird. Aus den zuvor genannten Punkten und der genannten Forschungsfrage lassen sich folgende Arbeitshypothesen ableiten, welche in dieser Arbeit geprüft werden sollen:

H<sub>0</sub>: Eine präemptive Smart Home Steuerung kann nicht mittels Machine Learning umgesetzt werden.

H<sub>1</sub>: Eine präemptive Smart Home Steuerung kann mittels Machine Learning umgesetzt werden.

## 1.3 Vorgehensweise

Einleitend wird eine Literaturrecherche durchgeführt, um bereits vorhandenes Wissen zu sichten und Grundlagen für diese Arbeit zu schaffen. Der Fokus liegt hierbei auf den Bereichen Smart Home, Smart Services und Machine Learning. Zusätzlich werden die grundlegenden Zusammenhänge zwischen Smart Home und Smart Services geprüft.

Zum methodischen Vorgehen wird einleitend betrachtet, wie Softwareentwicklung im empirischen Kontext zu verstehen ist, und in welcher Betrachtung dazu diese Arbeit definiert werden kann. Zusätzlich wird für das methodische Vorgehen ein Prozess definiert. Im Zuge dieses Prozesses wird ein konkreter Datensatz erarbeitet, der für das eigentliche Training der Machine Learning Modelle genutzt wird. Betreffend der Machine Learning Modelle wird anhand der Daten geprüft, welche vergleichbaren Datensätze bereits in Forschungen eingesetzt werden. Dies dient dazu, um die zuvor definierten Arbeitshypothesen zu prüfen.

Mit diesem methodischen Vorgehen soll konkret abgebildet werden, wie ein Prototyp in der empirischen Softwareentwicklung erarbeitet werden kann. Abschließend sollen die Ergebnisse

der Machine Learning Modelle aufgezeigt werden. Somit soll das Ziel dieser Arbeit mittels eines Proof-Of-Concept die Prüfung der Hypothese und der Forschungsfrage sein.

## **1.4 Struktur**

In dieser Arbeit werden in Kapitel 2 die Grundlagen und Definitionen genannt. Danach wird in Kapitel 3 das Untersuchungsdesign erläutert. Dabei wird einerseits auf die empirische Softwareentwicklung eingegangen, andererseits wird das konkrete methodischen Vorgehen beschrieben. Hierbei wird ein Prozess definiert, der sich mit den Themen zur Datenauswahl und den passenden Machine Learning Modellen befasst. In diesem Kapitel werden auch Informationen über genutzte Hardware, Tools und Software, sowie deren Versionen gelistet. In der Empirik (siehe Kapitel 4) werden die Ergebnisse der Datenaufbereitung und eine Analyse dieser aufgezeigt. Zusätzlich werden die Resultate der gewählten Machine Learning Modelle vorgestellt, sowie miteinander verglichen. Im letzten Kapitel 5 wird das Ergebnis dieser Arbeit zusammengefasst.

## 2 GRUNDLAGEN UND DEFINITIONEN

Nachfolgend wird ein grundlegendes Verständnis für Themen, die in Zusammenhang zu dieser Arbeit stehen, erläutert. Dabei werden das Ubiquitous Computing, Smart Home, Smart Service und Machine Learning definiert. Ein besonderer Fokus wird auf den grundlegenden Zusammenhang zwischen den Themen Smart Home und Smart Service gelegt. Dieser ist ein wesentliches Element für die Relevanz dieser Arbeit.

### 2.1 Ubiquitous Computing

*“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.” (Weiser, 1991, S. 94)*

Als Ubiquitous Computing wird die Allgegenwärtigkeit von Systemen bezeichnet. Wobei diese eine selbstverständliche Verwendung von datenverarbeitenden Elementen hat. Das Konzept des Ubiquitous Computing wurde 1991 von Weiser erstmals erwähnt. Das Pervasive Computing kann als Synonym dafür genannt werden. Zum Ubiquitous Computing zählen Trendfaktoren wie Wearables, Cloud Computing und das Internet of Things (IoT), aber auch Smart Environments und Smart Cities. Aus dem Ubiquitous Computing gehen weitere Felder (siehe Abbildung 1) hervor, wobei zum Beispiel folgende genannt werden können: Pervasive Data Science, Digitale Landwirtschaft und weitere (Brumitt et al., 2000; Cambra Baseca et al., 2019; Davies & Clinch, 2017; Forman & Zahorjan, 1994; Satyanarayanan, 2001).

Dieses Konzept kann auch als ubicomp bezeichnet werden. Wichtige Eigenschaften für den Erfolg des Ubiquitous Computing sind vor allem die Einfachheit der Verwendung, die Transparenz und die Mobilität. Unter Transparenz wird verstanden, dass die Benutzer/innen nicht mehr bewusst wahrnehmen, dass gerade ein computergestütztes System verwendet wird. Auch ein Kontextbewusstsein des Systems ist von Relevanz. Dieses Kontextbewusstsein sorgt dafür, dass den Benutzer/innen immer relevante Informationen zur Verfügung gestellt werden können. Wobei diese relevanten Informationen immer im Zusammenhang mit den von Benutzer/innen ausgeführten Tätigkeiten stehen. (Carvalho et al., 2017; Carvalho et al., 2018; Meyer et al., 2017; Ocegueda-Miramontes et al., 2019; Papaioannou et al., 2016; Temdee & Prasad, 2018)

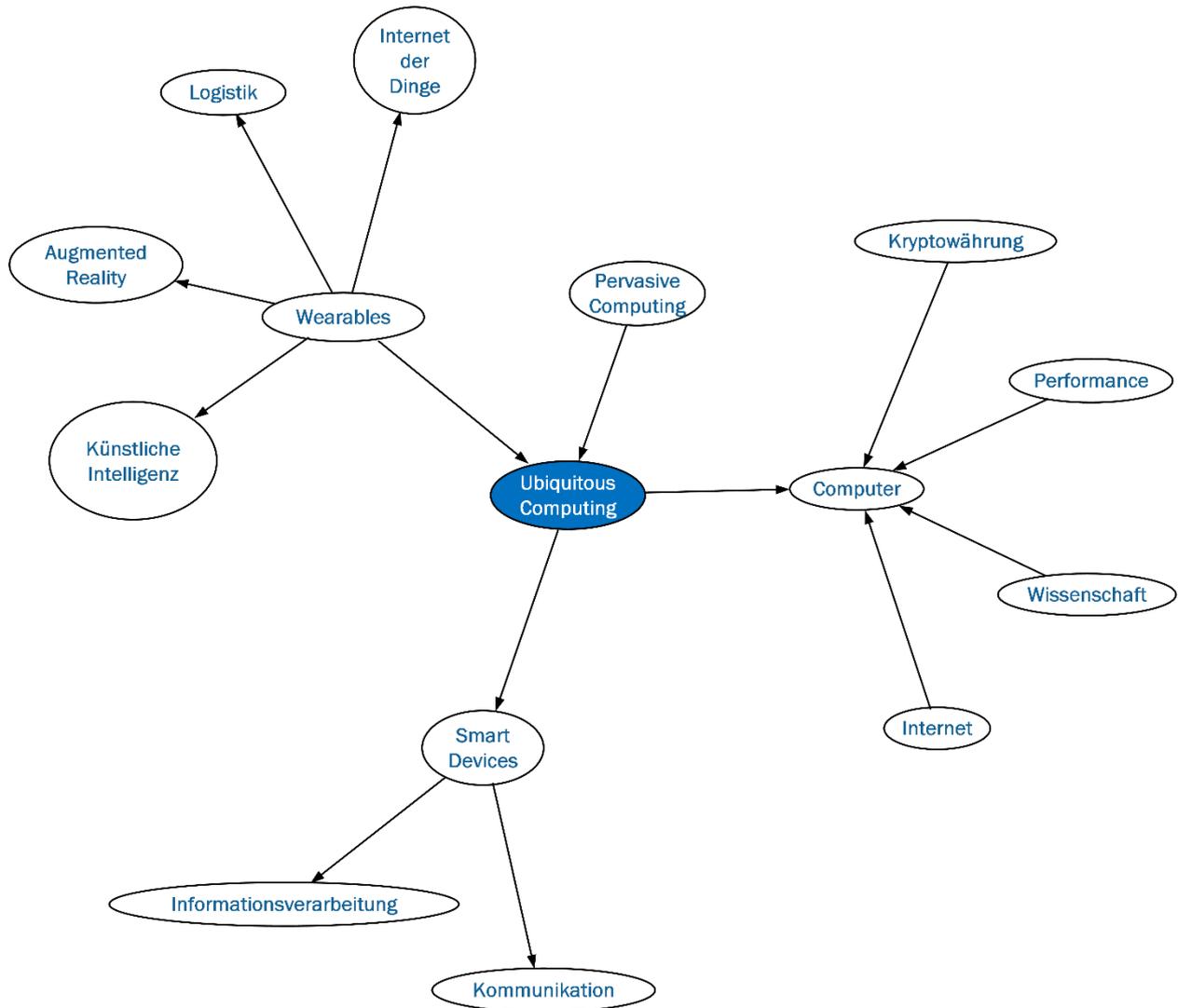


Abbildung 1: Zusammenhänge des Ubiquitous Computing laut Springer Gabler (vgl. Lackes & Siepermann)

## 2.2 Smart Home

*“How smart does the bed in your house have to be before you are afraid to go to sleep at night?”*

*(Gold, zitiert nach Gibbs, 2000, S. 40)*

Smart Homes sind ein Anwendungsgebiet von Ubiquitous Computing beziehungsweise Pervasive Computing (Alam et al., 2012). Ein zentrales Element ist ein Netzwerk aus Sensoren und Aktoren. Die Sensoren erfassen durchgehend Daten und senden diese zu einem zentralen Computer. Dieser wiederum steuert basierend auf den erfassten Daten oder durch Interaktion der Anwendenden die Aktoren (siehe Abbildung 2) (Lashkari et al., 2019).

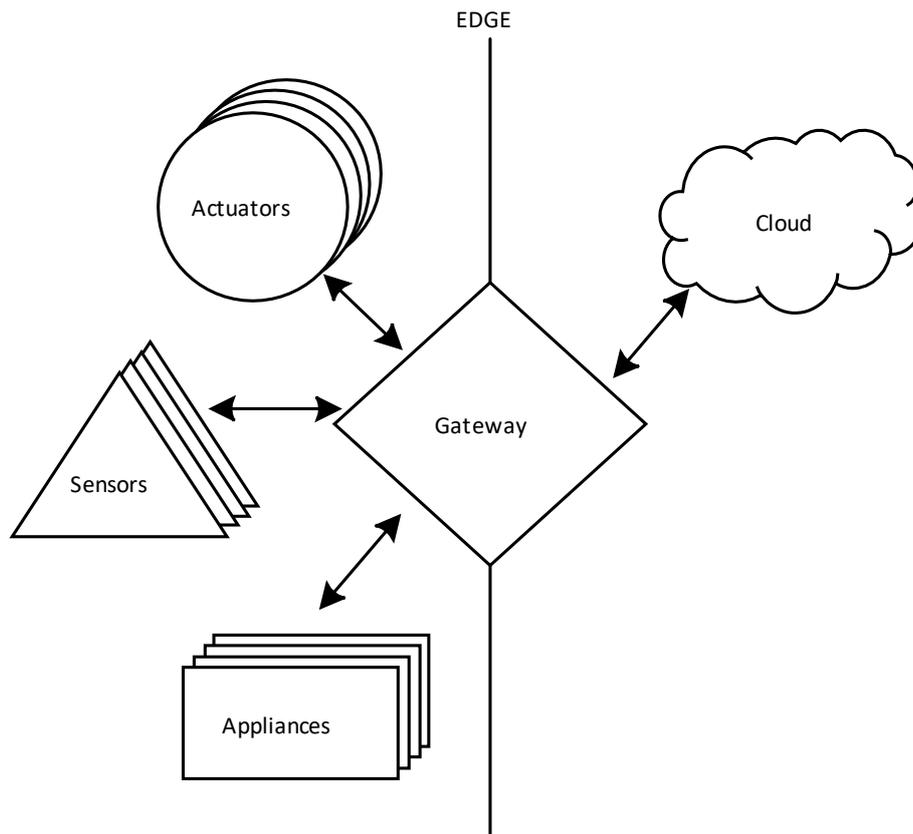


Abbildung 2: Generelle Cloud-Architektur eines Smart Home (in Anlehnung an Mocrii et al., 2018, S. 84)

Die Verknüpfung und der Informationsaustausch der Sensoren und Aktoren über das Netzwerk ermöglichen den Bewohner/innen neue Funktionen, welche zuvor noch nicht möglich waren, oder eine Steigerung von Effektivität und Effizienz von bestehenden Funktionen (Balta-Ozkan et al., 2014). Balta-Ozkan et al. (2014) nennen als Beispiel für neue Funktionen: Betreutes Wohnen. Silva et al. (2012) nennen hierbei konkret als Beispiel die Feststellung, ob ein älterer Bewohner/in allein, ist und sich nicht wohl fühlt. Für die Verbesserung von Funktionen nennen Balta-Ozkan et al. (2014) das ferngesteuerte Ein- und Ausschalten von Sicherheitssystemen. Silva et al. (2012) nennen das Einschalten von Beleuchtungselementen, wenn eine Person einen Raum betritt. Des Weiteren gliedern Balta-Ozkan et al. (2014) diese Funktionen in folgende drei breite, übergreifende, aber miteinander verbundene Kategorien (siehe Abbildung 3):

- Energieverbrauch und -management
- Sicherheit
- Lifestyle-Support

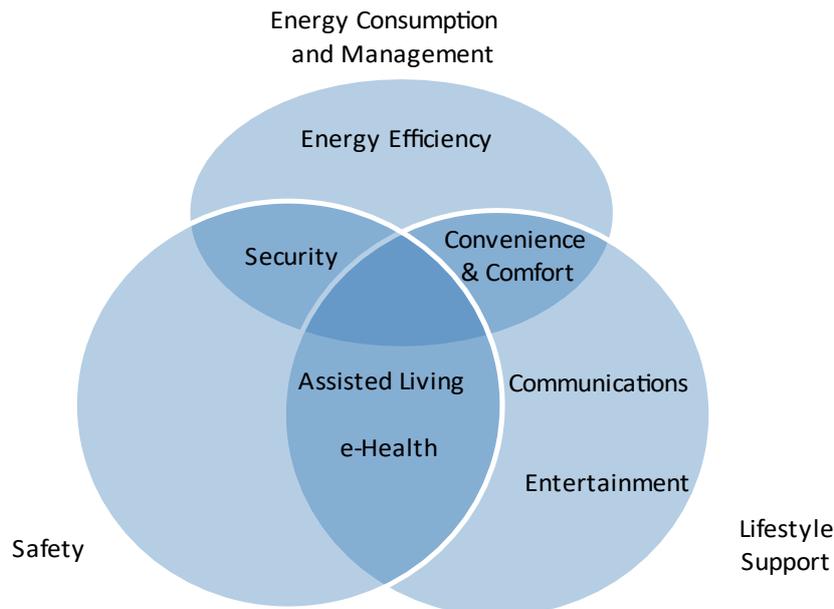


Abbildung 3: Typen eines Smart Home Services (vgl. Balta-Ozkan et al., 2013, S. 362)

## 2.2.1 Definition des Smart Home

*“Several definitions of Smart Homes highlight that its concept is to connect sensors, appliances and devices through a communications network, in order to remotely monitor, access or control the residential environment.” (Fabi et al., 2017, S. 760)*

Seit den ersten Anwendungen von Sensoren in Wohnräumen in den 1960er Jahren wurden immer wieder Bezeichnungen und Definitionen für Smart Homes definiert, wobei deren Inhalt teils variiert und in ihrem Umfang unterschiedlich breit gefächert ist. Strese et al. (2010) stellten ebenfalls fest, dass es noch keine allgemein anerkannte Begriffsbestimmung für Smarte Home gibt. In der Literatur gibt es einige Überschneidungen und aufbauende Betrachtungen zum Smart Home.

Bereits Scott (2007) weist auf den entscheidenden Unterschied von einem Zuhause zu einem Smart Home hin. Denn erst durch die Vernetzung einzelner intelligenter Komponenten erreicht ein Zuhause das, was im Grundverständnis ein Smart Home ist. Exemplarisch kann genannt werden, dass die Nutzung eines Smart Meter allein noch kein Smart Home ausmacht. (Scott, 2007) Erst, wenn einzelne Komponenten innerhalb des Netzwerkes miteinander in Interaktion treten, und so für den Kund/innen einen Synergieeffekt schaffen, kann das im Wesentlichen als Smart Home verstanden werden.

Um ein Verständnis zum Smart Home zu ermöglichen, wurden die Werke von Lutolf (1992), van Berlo und Allen (1999), Winkler (2002), van Berlo (2002), King (2003), Harper (2003), Li Jiang et al. (2004), van Berlo (2005), Satpathy (2006), Briere und Hurley (2007), Scott (2007), Taylor et al. (2007), Strese et al. (2010), Silva et al. (2012), Alam et al. (2012), Balta-Ozkan et al. (2014), Suryadevara und Mukhopadhyay (2015), Fabi et al. (2017) und Ali und Awad (2018) eingesehen.

Anhand dieser Durchsicht der genannten Werke, konnten gewisse wiederkehrende Themen erkannt werden.

Als ein wiederkehrendes Thema beschäftigen sich einige Autoren mit der Thematik, wo das Smart Home generell zum Einsatz kommt. Als weiteres Thema zur Definition eines Smart Homes konnte die Beschreibung dessen erfasst werden, was konkret in einem Smart Home enthalten ist. Dabei liegt der Fokus teilweise auf den Komponenten selbst und teilweise auf deren erweiterten Eigenschaften, beispielsweise das proaktive Handeln. Als wesentliches Thema in den Werken wird auch der erfasste Mehrwert genannt, den ein Smart Home liefern kann. Neben den definitionsschaffenden Werken wurden auch jene identifiziert, die sich auf eine zusammenfassende Rolle konzentrieren, vorangegangene Literatur erfassen und deren Inhalte kombinieren, um eine erweiterte Sicht auf das Smart Home und dessen Verständnis zu gewinnen.

Als Beispiel, wo ein Smart Home zum Einsatz kommt, nennt Lutolf (1992) als Anwendungsbereich das Zuhause des Menschen. Hier weiten van Berlo und Allen (1999) den Bereich auf die Arbeits- und Wohnumgebung aus. Strese et al. (2010) definieren für den Anwendungsbereich des Smart Homes ein privat genutztes Heim, wobei explizit Wohnhäuser und Wohnungen genannt werden. Für den Anwendungsbereich werden diverse Synonyme wie Connected Home, Elektronisches Haus, Intelligentes Wohnen, Smart House, Smart Environment, Home of the Future, Smart Living, Aware Home genannt (Strese et al., 2010). Silva et al. (2012) versuchen den Anwendungsbereich des Smart Homes allgemein als wohnliche Umgebung zu definieren. Somit zeigt sich in der Literatur, dass das Smart Home primär im privaten Wohnbereich angewendet wird, jedoch auch Ideen für das Arbeitsumfeld in diesem Kontext bestehen. Des Weiteren zeigt sich, dass es für den Anwendungsbereich des Smart Homes einige Synonyme gibt.

Betreffend des Bereichs, welcher sich konkret mit dem befasst, was in einem Smart Home enthalten ist, zeigt sich, dass die Komponenten oder Dienste im Grunde über ein gemeinsames Kommunikationssystem in einen Wohnraum integriert werden (Lutolf, 1992). Dies wurde von van Berlo und Allen (1999) zusätzlich betreffend der Technologie, die die automatische Steuerung der Geräte und Systeme umfasst, definiert. Für King (2003) hingegen ist die Verbindung von wichtigen elektrischen Geräten innerhalb eines Wohnraums in einem Kommunikationsnetzwerk relevant. Laut Harper (2003) ist es erforderlich, dass ein Wohnraum mit Computer- und Informationstechnologie ausgestattet wird, und dessen Verwaltung innerhalb des Zuhauses sowie darüber hinaus eine Verbindung zur Außenwelt aufweist. Für Satpathy (2006) erstreckt sich die Funktionalität des Smart Homes als ein Netzwerk, in dem alle mechanischen und digitalen Geräte miteinander verbunden sind. Diesen Gedanken nennen auch Briere und Hurley (2007) als Zusammenschluss von Geräten und Fähigkeiten auf Basis der Heimvernetzung. Denselben Ansatz verfolgen auch Strese et al. (2010), indem sie eine Vernetzung von Haushaltstechnik, Konsumelektronik und Kommunikationseinrichtungen beschreiben. Alam et al. (2012) greifen die Definition von King (2003) auf und ergänzen, dass für ein Smart Home ein internes Netzwerk, eine intelligente Steuerung und eine Automatisierung der Produkte und Dienste in und um den Wohnraum erforderlich ist, um als Umgebungszintelligenz, Hausfernsteuerung oder Hausautomatisierung genutzt werden zu können. Dieselbe Betrachtung haben Silva et al. (2012), dass ein Smart Home über eine Umgebungszintelligenz und eine automatische Steuerung verfügt. Erweitert betrachten Alam et al. (2012) das Smart Home in ihrer

Definition als eine Anwendung des Ubiquitous Computings. Winkler (2002) geht ebenfalls weiter, und nennt die proaktive Eigenschaft vom Smart Home, mit der es seine Umgebung verändert. Auch Harper (2003) nennt die Fähigkeit vom Smart Home, die Bedürfnisse der Bewohner/innen zu antizipieren und auf diese zu reagieren. Für Strese et al. (2010) sind die Orientierung an den Bedürfnissen der Bewohner/innen wesentlich, um Gegenständen Intelligenz zu verleihen. Somit kann erfasst werden, dass der Umfang der Komponenten, die in einem Smart Home enthalten sind, variiert. Das Wesen des Smart Homes jedoch ein Netzwerk dieser Komponenten voraussetzt, um ein Kommunikationssystem zu schaffen. Innerhalb dieses Netzwerks können dann die einzelnen Komponenten interagieren. Im Laufe der Zeit wurden verschiedene Definitionen genannt: Von einer simplen Kommunikation für eine Steuerung, über eine Fernwartung, über eine Hausautomatisierung, bis hin zu einer Umgebungsintelligenz. Daher scheint der Umfang vom Smart Home mit den steigenden Möglichkeiten zu wachsen.

Das Thema, welches sich vermehrt mit dem Mehrwert des Smart Homes beschäftigt, zeigt, dass eingehend King (2003) den Mehrwert auf die Fernsteuerung, Überwachung oder den Zugriff konzentriert, wobei die Fernsteuerung innerhalb des Wohnraums und auch darüber hinaus erfolgen kann. Bereits Lutolf (1992) nennt den Mehrwert eines Smart Homes. Laut ihm wird damit ein wirtschaftlicher, sicherer und komfortabler Betrieb gewährleistet und beinhaltet ein hohes Maß an intelligenter Funktionalität und Flexibilität. Betreffend Komfort greift Winkler (2002) den Mehrwert auf, um für Benutzer/innen eines Smart Homes Dienstleistungen anzubieten, die einen unabhängigen Lebensstil fördern. Harper (2003) nennt neben dem Komfort zusätzlich die Bequemlichkeit, Sicherheit und die Unterhaltung der Bewohner/innen. Von Satpathy (2006) wird der Mehrwert eines unabhängigen und komfortablen Lebensstils genannt, dessen Netzwerk einen interaktiven Raum als Kommunikation mit den Benutzer/innen schafft. Nach Taylor et al. (2007) liegt der Mehrwert im Angebot von Diensten, die den Bedürfnissen der Bewohner/innen entsprechen. Der Ansatz von Strese et al. (2010) besagt, dass durch die Vernetzung der Gegenstände Mehrwert, der über den einzelnen Nutzen der Gegenstände hinausgeht, geschaffen wird, indem neue Assistenzfunktionen und Dienste bereitstehen. Für Alam et al. (2012) liegt der Mehrwert in den kontextbezogenen, automatisierten oder unterstützenden Diensten, die den Benutzer/innen zur Verfügung stehen. Diese Betrachtung teilen auch Silva et al. (2012), indem auf das Verhalten von Bewohner/innen reagiert wird und ihnen unterschiedliche Funktionen bereitgestellt werden.

Es zeigt sich somit, dass die Werke einen Mehrwert durch ein Smart Home einerseits mit der Fernsteuerung, Überwachung, dem Zugriff von Komponenten, und durch die Interaktionsmöglichkeiten mit Benutzer/innen sehen. Ebenfalls zeigen sie den Mehrwert von Komfort, Funktion und Flexibilität, beispielsweise die bequemen, sicheren und unterhaltungswertstiftenden Eigenschaften des Smart Homes. Diese Eigenschaften können auch mittels Dienstleistungen unterstrichen werden, die durch die Technologien ermöglicht werden, und mittels Smart Home angeboten werden können. Denn in diesem Zusammenhang wird der eigentliche Synergieeffekt, der durch die Kombination einzelner Komponenten im Smart Home genutzt werden kann, deutlich. Dieser Synergieeffekt wird auch die Möglichkeit gestärkt, dass durch Informationen, die anhand von Komponenten gesammelt werden können, auf das

Verhalten von Benutzer/innen reagiert werden kann, um den zuvor genannten proaktiven Ansatz von Winkler (2002) aufzugreifen.

Wie einleitend genannt, gibt es neben den bisher definitionsschaffenden Werken auch jene, die ihren Fokus auf eine zusammenfassende Rolle legen. Diese versuchen aus einzelnen Definitionen und Erkenntnissen eine inhaltliche Kombination zu schaffen, um die Sicht auf Smart Home weiter zu schärfen. Li Jiang et al. (2004) verweisen gesamt für eine Definition des Smart Homes auf King (2003). Die Basis des Mehrwerts, weist auf die zentrale Rolle des Netzwerks innerhalb des Smart Homes hin. Die Smart Homes Foundation hat in Zusammenarbeit mit van Berlo (2005) die Ansätze von Winkler (2002) und van Berlo (2002) aufgegriffen und verdichtet, wobei die proaktive Eigenschaft des Smart Homes nicht explizit aufgegriffen wird. Sie legen jedoch Wert auf die integrierte Kombination von Technologie und Dienstleistungen, um durch die Heimvernetzung die Lebensqualität zu steigern (van Berlo, 2005). Balta-Ozkan et al. (2014) haben aufbauend auf die Definition von King (2003) und dem Mehrwertverständnis von Taylor et al. (2007) das Smart Home und das Verständnis der Wohnform spezifiziert, sowie die Geräte, die sich in einem Smart Home wiederfinden ausdefiniert. Zusätzlich haben Balta-Ozkan et al. (2014) den Gedanken von Li Jiang et al. (2004), betreffend dem Grundverständnis nach King (2003) und der erweiterten Betrachtung des Netzwerks als zentrales Element, aufgegriffen. Ali und Awad (2018) versuchen die bestehende Literatur zum Thema der Definition vom Smart Home zu erfassen, und verweisen mitunter auf die Definition von Fabi et al. (2017). Sie nutzen das Verständnis von Suryadevara und Mukhopadhyay (2015) betreffend der Dienste innerhalb einer smarten Umgebung. Des Weiteren greifen sie das Verständnis für die Komponenten und dem damit schaffbaren Mehrwert von Harper (2003) auf. Durch die Betrachtung von einzelnen Komponenten innerhalb des Smart Services als Umweltdienste, kann anhand der Abbildung 4 veranschaulicht werden, inwiefern ein Smart Home als Vernetzung einzelner Umweltdienste verstanden wird.

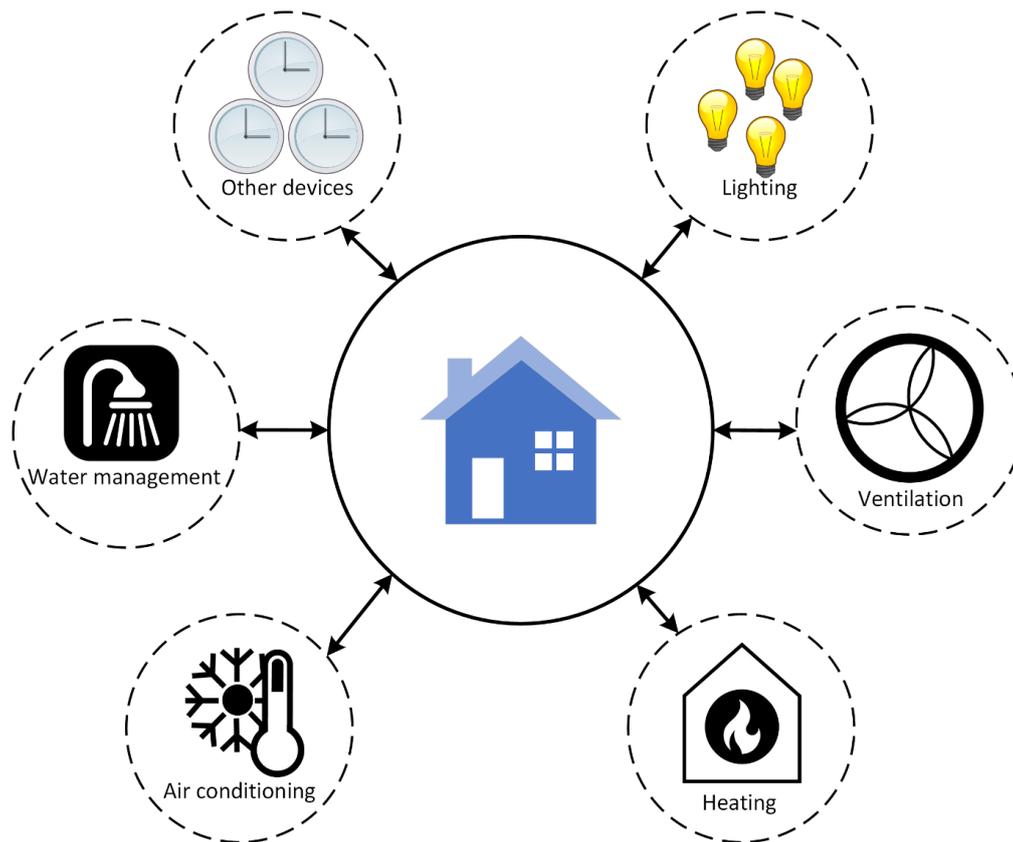


Abbildung 4: Ein Beispiel für die kontrollierten Umweltdienste eines Smart Home (in Anlehnung an Ali & Awad, 2018, S. 3)

Die Definition eines Smart Homes befindet sich noch in der Entstehung. Aus den bestehenden Definitionen kann jedoch abgeleitet werden, dass sich der Begriff des Smart Homes mit der örtlichen Komponente beschäftigt. Des Weiteren wird für diese örtliche Komponente im engeren und erweiterten Sinne definiert, was diese Komponente beinhaltet und der erreichte Mehrwert wird ebenso aufgezeigt. Wesentlich ist, dass Smart Home von den Begriffen Smart Building oder Intelligentes Gebäude abzugrenzen ist. In der Regel sind darunter mehrere Bauten zu verstehen, die räumlich getrennt sind. Zusätzlich weist deren Charakter einen betriebswirtschaftlichen Fokus auf, und benötigt daher andere Schwerpunkte als das Smart Home für Privatbenutzer/innen. (Strese et al., 2010)

### 2.2.2 Geschichte und Zukunft des Smart Homes

Harper (2003) nennt als wichtigste Neuerungen in unserer Lebensweise, die Smart Homes ermöglicht haben, die Einführung von Elektrizität und Informationstechnologien. Strengers (2014) nennt als erste Beispiele die Homes of Tomorrow der 1930er, welche damals noch nie dagewesene Ausmaße an Luxus, Entspannung und Genuss präsentiert haben. Harper (2003) nennt hingegen, dass die Ursprünge mindestens bis in die 1960er Jahre reichen, als sogenannte Wired-Homes aufkamen.

Die American Association of House Builders (zitiert nach Harper, 2003) verwendete als Erstes den Begriff Smart Home im Jahr 1984. Im selben Jahr gründete die National Association of Home Builders eine Interessensgruppe mit dem selben Namen und dem Ziel, die Verwendung der nötigen Technologien in Neubauten voranzutreiben (zitiert nach Harper, 2003).

Derzeit ist das Smart Home in dessen Form noch in Entwicklung, da auch die Erwartungen der Benutzer/innen an die Technologien im Alltag kontinuierlich wachsen (Mennicken et al., 2014). In der Vergangenheit war eine reine Erkennung von Bewohner/innen und deren individuellen Einstellungen, beispielsweise für Licht oder Wärme, ausreichend (Harper, 2003). Dennoch zeigen Strese et al. (2010) bereits auf, dass sich das Smart Home in Zukunft stärker vernetzen wird (siehe Abbildung 5). Dabei sind Einflüsse innerhalb des Eigenheims, anderen Umgebungen, sowie im mobilen Alltag bedacht. Dieser Gedanke basiert darauf, dass der Alltag des Menschen stärker vernetzt ist, und auch sein Komfort dadurch von übergreifenden Informationen profitieren kann (Strese et al., 2010).

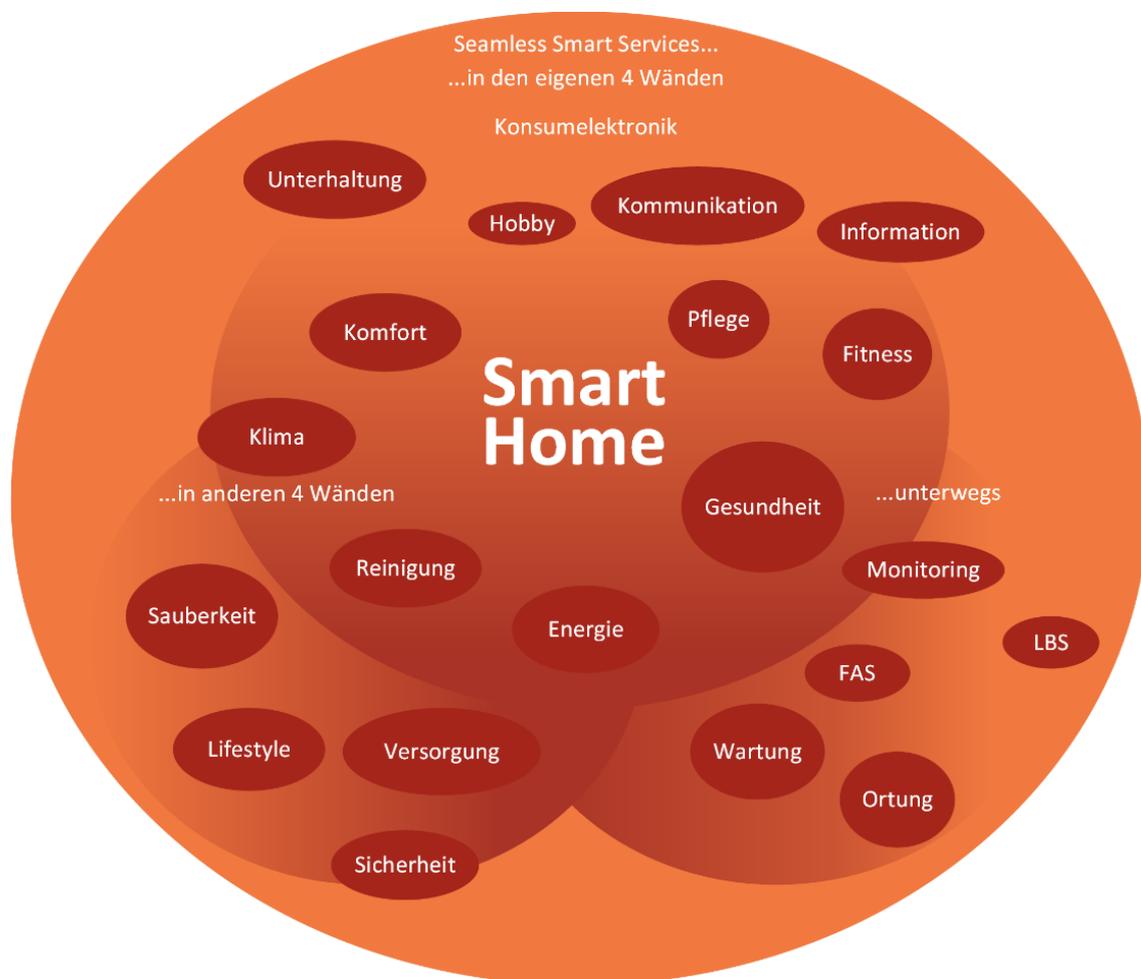


Abbildung 5: Die weitere Entwicklung führt schließlich zur Konvergenz der Dienste. (vgl. Strese et al., 2010, S. 9)

Statista prognostiziert bis 2025 eine durchschnittliche jährliche Wachstumsrate des Umsatzes mit Smart Homes von 16,1% auf über 195 Milliarden USD (siehe Abbildung 6). Die Marktdurchdringung wird hierbei voraussichtlich 21,3% betragen, das entspricht in etwa 481 Millionen Haushalte (siehe Abbildung 7). Auch Strese et al. (2010) erwarten bis 2030, dass sich die Nutzung von Smart Home im Alltag steigern wird und der damit verbundene Komfort wächst.

Daher zeigt sich bereits, dass das Smart Home Segment zwar noch im Wachstum ist, jedoch als grober Ausblick zu erwarten ist, dass ein Smart Home den Menschen der Zukunft aktiv begleiten wird. Ein Smart Home wird in der Zukunft ein Teil des Alltags jedes Individuums werden.

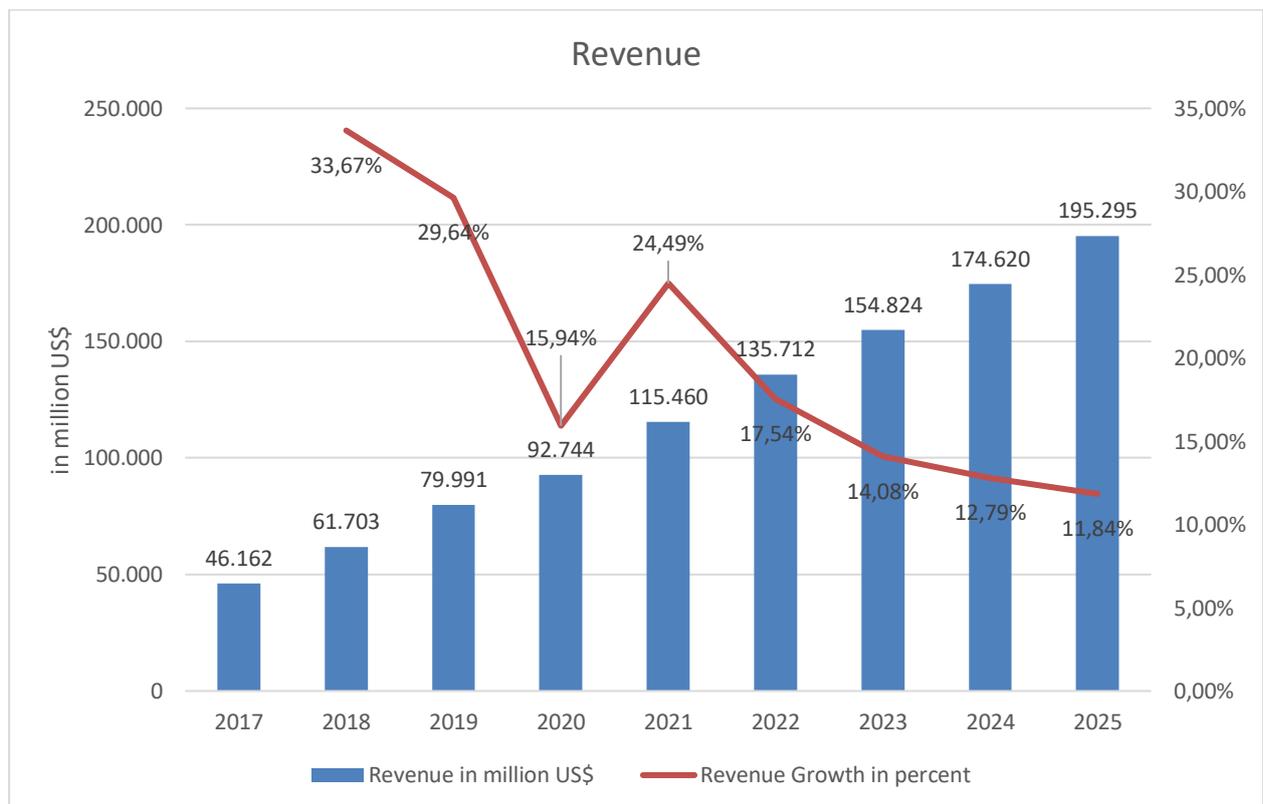


Abbildung 6: Umsatz mit Smart Homes weltweit (in Anlehnung an Statista, 2020)

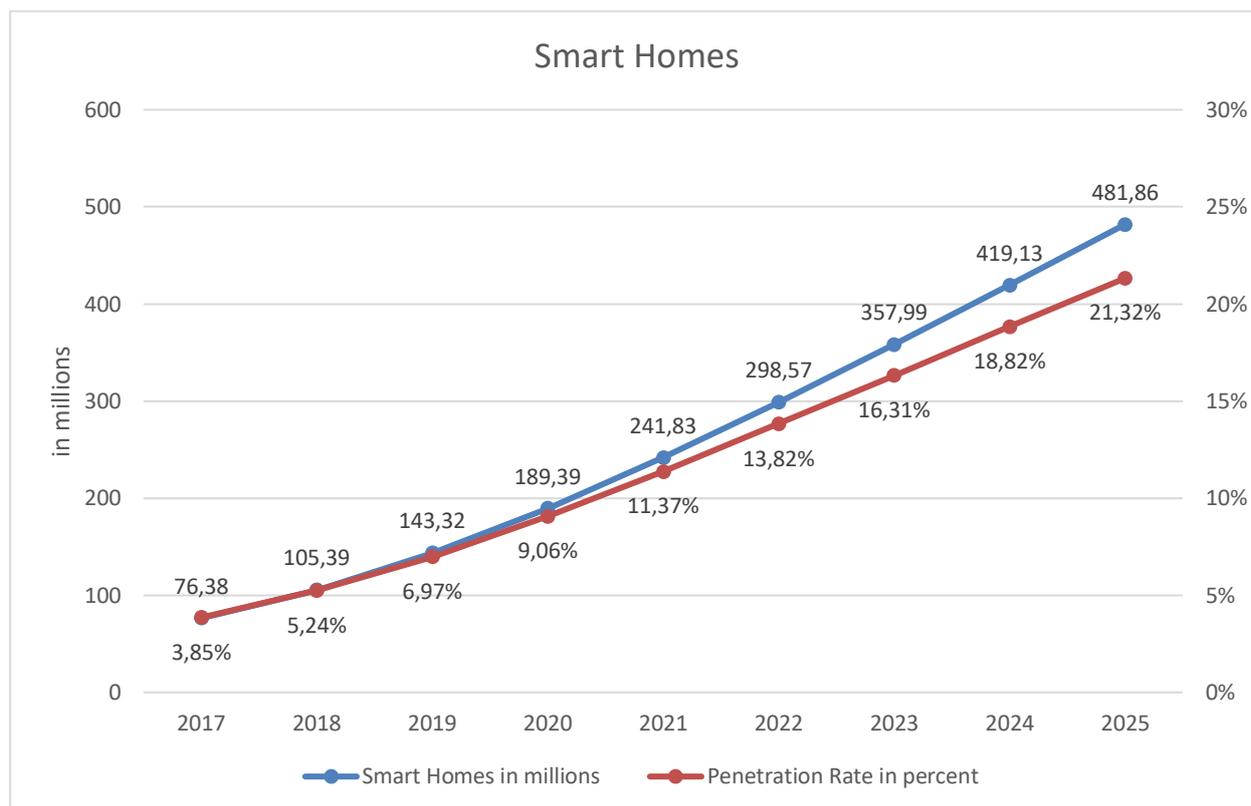


Abbildung 7: Anzahl Smart Homes weltweit (in Anlehnung an Statista, 2020)

### 2.2.3 Komponenten des Smart Homes

Das Smart Home Netzwerk besteht aus zwei Elementen: einer „physischen“ Verbindung der Komponenten – meistens eine kabelgebundene Verbindung oder ein Funksignal – und einer gemeinsamen Sprache, über welche die Daten mit dem zentralen Computer und den anderen Komponenten ausgetauscht werden können. (Balta-Ozkan et al., 2014) Robles und Kim (2010) gruppieren zusätzlich noch die Komponenten des Smart Homes in folgende zwei Kategorien:

- Empfänger
- Sender

Als Empfänger bezeichnen Robles und Kim (2010) alle Apparate und Geräte, die gesteuert werden sollen. Strese et al. (2010) beschreiben die Klassifizierungen der beiden Empfänger detaillierter. Als Apparate werden Komponenten der Hausautomatisierung bezeichnet (siehe Tabelle 1), und als Geräte Produkte und Dienste, welche direkt auf die Bewohner/innen bezogen sind (siehe Tabelle 2).

Für Sender wiederum nennen Robles und Kim (2010) direkt als Beispiel Fernsteuerungen und Tastenfelder. Des Weiteren kann in dieser Kategorie auch ein Hausenergie-Management-System, beispielsweise von Lobaccaro et al. (2016), genannt werden (siehe Tabelle 3).

Teilsysteme	Produkte und Dienste
<b>Heizung</b>	Heizungsanlage, Temperaturregelung, integrierte Wetterstation
<b>Lüftung/Klima</b>	Zu- und Abluftregelung, Schadstoffableitung, Ventilation
<b>Sanitär</b>	Trink-, Brauch-, Abwasser; Installation, Armaturen
<b>Elektrik</b>	Installation; Verteilung
<b>Energiemanagement</b>	Lastverteilung und -prognose, eventuell Eigenversorgung durch alternative Energien
<b>Licht</b>	Beleuchtung, Lichtmanagement/Szenarien, Storen/Rollos
<b>Zutritt</b>	Zutrittskontrolle, Klingelanlage, Schlösser, Anwesenheits- und Bewegungserfassung
<b>Überwachung</b>	Technische Alarmer: Feuer, Rauch, Gas; Intrusion: Glasbruchmelder, Video; Babyphon, Urlaubswachschutz, ...
<b>Notfall</b>	Sprinkleranlagen, unabhängige Stromversorgung, Fluchtwegsystem
<b>Metering</b>	Verbrauchszähler für Strom, Gas, Wasser, Wärme; Energiedienstleistungen wie Fernablesen, genaue tagesaktuelle Abrechnung (auch bei stark differenzierten Tarifen)
<b>Umfeld</b>	Grünflächen-/Gartenberegnung, -düngung

Tabelle 1: Komponenten der Heimautomatisierung (in Anlehnung an Strese et al., 2010, S. 8)

Teilsysteme	Produkte und Dienste
<b>Kommunikation</b>	Mobil- und Festnetztelefon, Antennenanlage/Kabel/Satellitenempfang, LAN, WAN, Inter-/Intranet
<b>Konsumelektronik</b>	TV, Video, Audio, IKT mit PC, Internet, Teledienste, Kamera, ...
<b>Hausgeräte</b>	Kühlschrank, Waschmaschine, Trockner, Spülmaschine, Staubsauger, Serviceroboter; Hausgerätemonitoring, -diagnostik und -fernbedienung
<b>Gesundheit/Pflege</b>	Medizinische Diagnostik und Vorsorge, Vitalparametermessung (zum Beispiel intelligentes WC, vernetzte Waage, Blutdruckmessgerät), Kranken-/Alten-/Behindertenbetreuung, Sturzerkennung, Aktivitätsmonitoring
<b>Heimlogistik</b>	Einkaufs- und Speiseplanung, Lieferservice; häusliche Dienste
<b>Hobby</b>	(elektronische) Spiele; elektrische Eisenbahn, Haustierversorgung, Aquarienmanagement
<b>Mobilität</b>	PKW mit Diagnostik/telemedizinischer Überwachung, Navigationssystem für local based services, Info-/Entertainmentangebote; e-Mobile als Teil des Energiemanagementsystems („rollende Batterie“) im Smart Home

Tabelle 2: Direkt auf die Bewohner/innen bezogene Produkte (in Anlehnung an Strese et al., 2010, S. 9)

Kategorie	Technologie	Beschreibung	Beispiele
Steuergeräte	Zentralisiert	Es ermöglicht die Kommunikation mit mehreren Geräten in der häuslichen Umgebung. Die Benutzer können sie von einem einzigen Standort aus und oft über mehrere Eingänge verwalten	Hausautomatisierungssysteme; Lichtsteuerungssysteme für das ganze Haus Steuerungssysteme; Sicherheitssysteme
	Gerät/Anlage-Ebene	Benutzer steuert ein einzelnes Gerät oder Funktion und autonome Steuerung	Beleuchtungssteuerung mit Bewegungsmeldern, Dimmern, Fernbedienungen, Zeitplanung; Thermostate; Intelligente Steckdosen; Intelligente Steckdosenleisten
	On-Board (automatisiert auf Geräteebene)	Steuerungsfunktionalität im Gerät integriert	Intelligente Geräte, die z. B. auf das Stromnetz reagieren; Bürogeräte Energiemanagement; Intelligente Glühbirnen
Grafische Benutzeroberflächen	Home Energy Display (HED)	Freistehendes In-Home-Display; oft tragbar	Viele auf dem Markt befindliche HEDs verfügen über kompatible Web-Dashboards oder können mit Freeware von Drittanbietern verbunden werden; Versorgungsunternehmen, die intelligente Zähler implementieren, stellen ihren Kunden oft Web-Dashboards zur Verfügung
	Web-Dashboard / Portal	Online-Oberfläche, die von jedem internetfähigen Gerät aus zugänglich ist	
	Smartphone Applikation	Gerätespezifische Oberflächen für iPhones, Android-Telefone und andere	Aktuelle Produkte ziehen typischerweise Informationen von einer Hausenergieanzeige und erzeugen verarbeitete Daten;
	Andere (z.B. TV)		Multimedia-Ansätze umfassen Kombinationen aus Display/Web/Telefon-Feedback
Ermöglichende Technologien	Erfassen	Erfassen von dynamischen Variablen innerhalb der häuslichen Umgebung	Intelligente Zähler; Temperatursensoren; Anwesenheitssensoren
	Kommunikation	Physikalische Geräte, die zur Unterstützung des Netzwerks erforderlich sind	Gateways; Reichweitenverlängerer; Home Area Networks
	Kommunikationsprotokolle	Standards, die es einzelnen Knoten innerhalb eines Netzwerks ermöglichen, zu kommunizieren	Insteon; Z-wave; ZigBee; Andere

Tabelle 3: Beschreibung und Beispiele für Hausenergie-Management-Systeme (in Anlehnung an Lobaccaro et al., 2016, S. 13ff)

Es gibt eine große Anzahl unterschiedlicher Protokolle mithilfe derer die Komponenten miteinander kommunizieren können. Lobaccaro et al. (2016) nennen zum Beispiel folgende kabellose Protokolle, welche damals die weitverbreitetsten waren:

- 6LoWPAN
- Bluetooth
- DASH7
- EnOcean Technology
- GSM
- MyriaNed
- NeuRFon™ Netform
- RFID
- UWB
- Wi-Fi
- WLAN
- Z-Wave
- ZigBee

Als relevanteste kabelgebundene Protokolle nennen Mendes et al. (2015) folgende:

- Home Plug
- X10
- Insteon
- ITU G.hn
- MoCA
- M-Bus
- KNX
- LonWorks

Diese Protokolle sind meist nicht interoperabel, was bedeutet, dass wenn von den Komponenten nicht ein gemeinsames Protokoll unterstützt wird, ein Gateway zum Einsatz kommen muss. Solch ein Gateway ist in der Lage, die Kommunikation durch Übersetzen über mehrere Protokolle hinweg zu ermöglichen (Mendes et al., 2015). Die Abbildung 8 zeigt ein beispielhaftes Framework eines Home Area Netzwerkes.

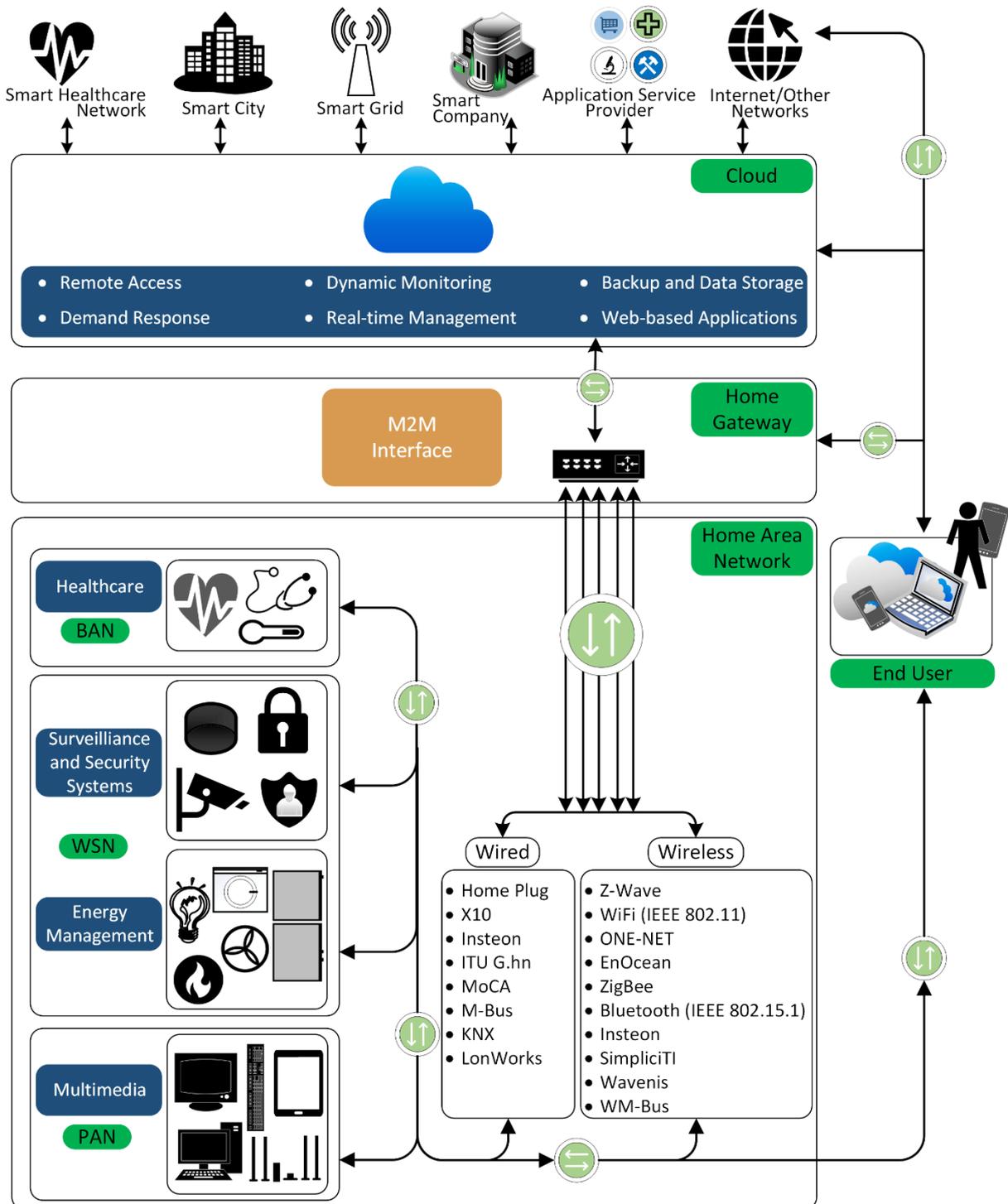


Abbildung 8: Zugriff auf Smart Home Geräte über ein globales Netzwerk (in Anlehnung an Espinilla, Medina & Nugent, 2018, S. 7287)

## 2.2.4 Herausforderungen des Smart Homes

Bereits 1998 wurde explizit darauf hingewiesen, dass die Smart Home Industrie einige Kriterien erfüllen muss, bevor die Konsumenten eine Kaufbereitschaft für diese Produkte aufweisen (Barlow & Gann, 1998). Dabei wurde 1999 erläutert, dass die Verbreitung von Smart Home langsam sein wird, da es diverse Herausforderungen und Hindernisse gibt. Einerseits wurden

Hindernisse in finanzieller, technischer und planungsseitiger Hinsicht erkannt, andererseits einem fehlenden Verständnis für die Kundenanforderungen. (Gann et al., 1999) Im Wesentlichen kann für diese Herausforderungen erfasst werden, dass zuerst die realen Kundenbedürfnisse erkannt werden müssen. Die daraus entwickelten Lösungen müssen auf drei Ebenen operieren: Als Basis eine generische Technologie, auf die kompatible kontextspezifische Systeme ergänzt werden können, und eine Ebene für personalisierte Systeme. Zusätzlich lassen sich die Herausforderungen Funktionalität, einfache Nutzung, Leistbarkeit, Zuverlässigkeit, Wartungsfreundlich, Flexibilität, Adaptierbarkeit, Upgrade-Fähigkeit, sowie Replizierbarkeit und einfache Installation nennen. (Harper, 2003)

Für die finanziellen Herausforderungen, wobei ein Smart Home leistbar für die Konsumenten sein soll, ist festzuhalten, dass die initiale Anschaffung für Konsumenten hoch ist. Dadurch wird der Marktfokus bereits auf die mittlere und obere Einkommensschicht beschränkt (Balta-Ozkan et al., 2014; Gann et al., 1999). Zusätzlich werden Smart Homes teilweise nur von technikaffinen Menschen wahrgenommen (Balta-Ozkan et al., 2014). Generell müssen die potenziellen Kund/innen erst von den Vorteilen überzeugt werden, um eine Kaufbereitschaft zu schaffen (Harper, 2003). Neben der Argumentation der Anschaffungskosten für ein Smart Home ist auch die Relevanzbegründung ein wichtiger Faktor, den Unternehmen überwinden müssen. Hierzu können Energie- und Kosteneinsparungen durch Smart Home Technologien ein wichtiges Potential zur Überzeugung der potenziellen Kund/innen sein. (Balta-Ozkan et al., 2014)

In diesem Kontext kann auch der Umweltschutz gesehen werden, welcher durch eine effizientere Nutzung bestehender Ressourcen entsteht. Dies zeigt sich mitunter durch Verhaltensänderungen von Haushalten, indem unter anderem Wäschetrockner seltener genutzt werden und der Energieverbrauch im Haushalt gerne genauer über In-Home-Display-Geräte überwacht wird. Somit zeigt sich bereits im Verhalten von potenziellen Kund/innen, dass der Vorteil im Monitoring des Energiebedarfs durch Smart Home gedeckt werden kann. Jedoch ist hierbei essenziell, dass die Smart Home Technologie eine Vertrauensbasis schafft, denn Personen stehen dem Smart Home teilweise noch skeptisch gegenüber. Eine Einstiegsbarriere für die Anschaffung von Smart Home Technologie kann auch die fehlende Bereitschaft, die bestehende Haushaltsroutine zu ändern, sein. Trotz Umweltbewusstsein, ist dennoch der Preis ein größeres Entscheidungskriterium für die Anschaffung eines Smart Homes. (Balta-Ozkan et al., 2014) Generell lässt sich die Wichtigkeit des finanziellen Faktors einer Smart Home Technologie hervorheben, da deren Gewichtung oft höher liegt, als der Komfort und die Kompatibilität der Technologien. Je nach Argumentation kann jedoch der finanzielle Aspekt eine untergeordnete Rolle einnehmen. (Park et al., 2018) Die Kaufbereitschaft kann durch die transparente Informationsbereitstellung von verursachtem versus eingespartem Energieverbrauch, welcher als Echtzeitüberwachung bereitgestellt wird, gefördert werden. Ebenfalls ist das Wesen der Kund/innen relevant, denn Personen, die nicht genügend Wert auf die Zukunft legen, werden die Anschaffungskosten eines Smart Homes nicht mit späterer Einsparung durch Energieeffizienz akzeptieren können. (Balta-Ozkan et al., 2014) Die Kommunikation des Vorteils im Energiemanagement zeigt sich als eine kurzfristige Perspektive, die kommuniziert werden kann. Hingegen ist die ökologische Nachhaltigkeit die langfristige Perspektive. (Marikyan et al., 2019) Neben Energiekosten können sich auch die Aufwände für Installation und Wartung von Smart

Home Technologien als Einstiegsbarrieren für potenzielle Kund/innen erweisen (Balta-Ozkan et al., 2014).

Für die genannten Haushalte zeichnet sich vermehrt ab, dass Smart Home als Anschaffung für zukünftige Hauseigentümer in die Planung einbezogen wird. Denn die Kosten für die Integration von Smart Home Technologien sind bei bereits bestehenden Haushalten eine wahrgenommene Barriere. Der Aufwand und die Kosten für diese Änderungen müssen erst mit den Vorteilen durch die dadurch ermöglichte Nutzung von Smart Home Technologien argumentiert werden. Andererseits sehen Personen, die eine Immobilie lediglich mieten eine Barriere darin, dass sie keine baulichen Veränderungen vornehmen können, um Smart Home Technologien zu nutzen. (Balta-Ozkan et al., 2014) Es zeichnete sich dazu bereits ab, dass ein modulares Smart Home System eine Option liefert, wodurch Kund/innen diese Technologien stufenweise erwerben können (Harper, 2003). Dieser modulare Ansatz zeigte sich exemplarisch beim Apple iPhone als erfolgsbringend, indem durch diesen modularen Ansatz jeder Aspekt vom Gerät sicherstellt, dass jede Funktion des Smartphones am neuesten Stand war und in der Nutzung klar strukturiert ist (Holroyd et al., 2010).

Somit lässt sich bei den finanziellen Herausforderungen sagen, dass diese sich aus den Anschaffungs-, Installations-, Wartungs- und gegebenenfalls Umbaukosten zusammensetzen. Das Wesen der finanziellen Herausforderungen selbst wird initial als zu hoch empfunden, und es müssen für die potenziellen Kund/innen Vorteile aufgezeigt werden, um den Kostenfaktor zu begründen. Erst, wenn die Vorteile und eine damit verbundene Kosten-Nutzen-Rechnung für die Kund/innen sich als positiv erweist, kann diese Herausforderung überwunden werden. Dabei sind Vorteile, wie Energiemanagement, Nachhaltigkeit und Nutzungsvorteile sowohl für nicht technikaffine Personen, als auch für bereits bestehende Haushalte, hervorzuheben. Das Verständnis für die Vorteile selbst ist auch ein essenzielles Element, um die finanziellen Aufwände begründen zu können, denn wenn die Vorteile nur für technikaffine Personen nachvollziehbar sind, schränkt sich die Kaufbereitschaft auf einen kleinen Markt ein. Es ist auch davon auszugehen, dass die potenziellen Kund/innen ausreichend Liquidität innehaben müssen, dies gilt lediglich für die mittlere und obere Einkommensschicht. (Balta-Ozkan et al., 2014; Gann et al., 1999)

## **2.3 Smart Services**

Einleitend ist der Unterschied zwischen Dienstleistungen, und wie diese in der Vergangenheit verstanden und genutzt wurden, und dem Smart Service, welches streng genommen ebenfalls eine Dienstleistung ist, festzuhalten (Allmendinger & Lombreglia, 2005). Die Entstehung von Smart Services beruht auch auf einem der Megatrends der heutigen Zeit, der Digitalen Transformation. Im Zuge der Digitalen Transformation werden auch Konzepte zum Internet-of-Things und der Industrie 4.0 genannt. Diese beschäftigen sich mit der Ausstattung von Sensorik bei Produkten und Geräten, wodurch mittels Datenverarbeitung Smarte Services entwickelt werden können. (Bullinger, 2015) Acatech - Deutsche Akademie für Technikwissenschaften (2015) haben für die Entwicklung der Smart Services mit einem Smart Space, der rein auf einer

technischen Infrastruktur basiert, begonnen, und dadurch den Weg hin zu einem Smart Service, wie es mittlerweile als präemptive Handlungsform genutzt wird, aufgezeigt (siehe Abbildung 9).

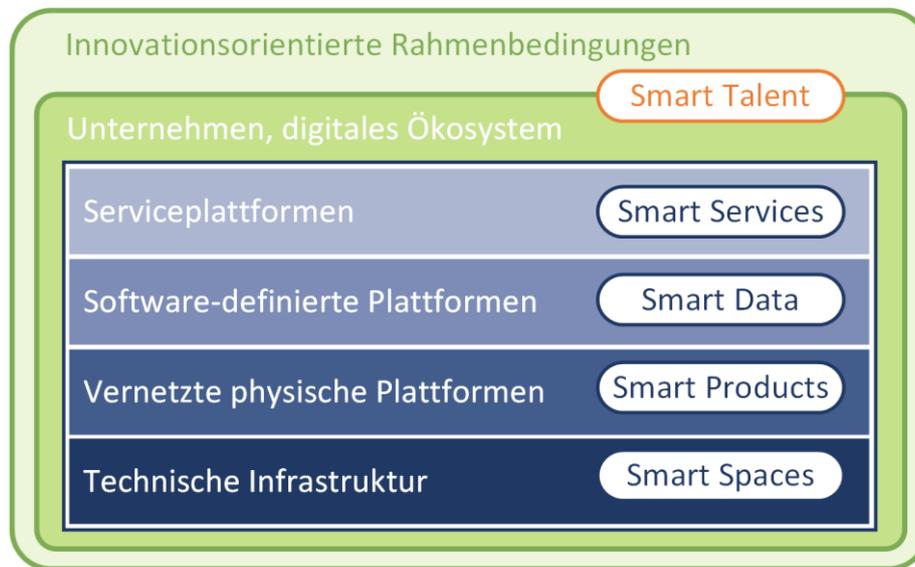


Abbildung 9: Ebenen integrierter Smart Service-Plattformen (vgl. Acatech - Deutsche Akademie für Technikwissenschaften, 2015, S. 17)

Smart Services zählen zu den präemptiven als auch zu reaktiven oder gar proaktiven Handlungsformen. Die präemptive Handlungsform beruht auf der Tatsache, dass Daten als Feldforschung gesammelt werden, und anhand dieser Informationen über zukünftig eintretende Ereignisse gewonnen werden können. Somit kann eine Tätigkeit bereits vor Eintreten des Ereignisses ausgeführt werden, um das Ereignis selbst zu vermeiden. Beispielhaft kann ein Smart Service für Druckerpatronen und -papier genannt werden. Das Unternehmen, das den Drucker an seine Kund/innen verleiht, hat auch ein Service inkludiert, durch das der Verbrauchsverlauf von Patronen und Papier aufgezeichnet wird, um individuell für jede/n Kund/innen eine Nachlieferung rechtzeitig erfolgen zu lassen. Damit haben Kund/innen nie zu wenig oder zu viel Verbrauchsartikel vor Ort, wodurch nur für den tatsächlichen Bedarf Kosten anfallen. Weitere Beispiele auf deren Basis ein Smart Service beruht sind: eine Maschine, die kurz vor dem Wartungsmodus steht; Vorräte von Materialien bei Kund/innen, die zur Neige gehen; eine Maschine, die kurz vor dem Ausfall steht; eine Lieferung, die sich verzögert hat; und viele weitere Beispiele. Damit schaffen Smart Services als präemptive Handlungen eine neue Art von Wert. Dieser Wert liegt darin, unerwünschte Ereignisse zu vermeiden und somit negative Überraschungen aus dem Alltag zu entfernen. (Allmendinger & Lombreglia, 2005) Somit beschreiben Smart Services ein Leistungsangebot, das auf Daten basiert und individuell konfiguriert werden kann (Bullinger et al., 2017).

Solch ein Leistungsangebot kann von Unternehmen mittels Dienstleistung oder einem digitalen Dienst und Produkten erbracht werden (Acatech - Deutsche Akademie für Technikwissenschaften, 2015). Mittels der präemptiven Handlungsform können Unternehmen auf eine neue Art Forschung und Entwicklung betreiben, beziehungsweise das implizite Feedback, anhand gesammelter Daten, nutzen. Smart Services tragen Sorge, dass Leistungen und

Nutzungsverhaltensweisen so transparent wie möglich werden und Bedürfnisse der Kund/innen greifbarer machen, um die optimale Lösung entwickeln und zusätzlichen Mehrwert schaffen zu können. (Allmendinger & Lombreglia, 2005) Das Einsetzen von Sensoren und das Vernetzen von Geräten, Maschinen und Produktionen an sich ist der ursprüngliche Mehrwert im Vergleich zu vorangegangenen Vorgehen und somit die treibende Kraft der Digitalisierung (Bullinger et al., 2017). Beim Smart Service liegt der Fokus jedoch auf dem Sammeln von Informationen und diese zielführend auszuwerten, zu kombinieren und in einen Mehrwert für Kund/innen zu verwandeln (Ganz et al., 2016).

Das Sammeln der Daten mittels Smart Services erfolgt grundlegend über Machine Learning, denn Menschen könnten diese Masse an Information nicht in Echtzeit verarbeiten. Dabei werden zuverlässige und auf Geschwindigkeit optimierte Mikroprozessoren eingesetzt, welche Daten erfassen und verarbeiten. (Allmendinger & Lombreglia, 2005) Diese Sensoren liefern, durch Vernetzung von einzelnen Geräten und Maschinen, bis hin zur Vernetzung ganzer Produktionen, erhebliche Potentiale im Dienstleistungssektor, indem aus der gesammelten Information Smart Services entwickelt werden (Bullinger, 2015). Mittels dieser Mikroprozessoren können in Nanosekunden Milliarden an Informationen erfasst, ausgetauscht und miteinander vernetzt werden, um sich gegenseitig anhand des Datenzustand zu steuern. Diese Abläufe werden als Hintergrundaktivitäten ausgeführt, denn im alltäglichen Leben sollten diese nicht direkt für Menschen einzusehen sein. (Allmendinger & Lombreglia, 2005) In diesen Hintergrundaktivitäten werden in integrierten Plattformen unstrukturierte Datenmassen in strukturierte Datensätze transformiert. Durch diese Strukturierung können Leistungsangebote spezifisch auf die Bedürfnisse zugeschnitten erstellt werden. Diese speziellen Plattformen weisen den Charakter der Digitalen Transformation selbst auf, denn hier erfolgt die Betrachtung nicht mehr in der Hersteller- sondern in der Kundenperspektive. (Bullinger et al., 2017) Diese Hintergrundaktivitäten liefern auch im Bedarfsfall zusätzliche Information für Entscheidungen des Managements (Allmendinger & Lombreglia, 2005).

In vielen Unternehmen werden bereits seit Jahrzehnten Sensoren, Controller und Mikroprozessoren kontinuierlich aufgebaut, um eine elektronische Intelligenz in den hergestellten Objekten zu integrieren. Dieses Vorgehen führt dazu, dass ein Großteil der elektronischen und elektromechanischen Geräte bereits über Datenverarbeitungsfunktionen verfügt. Beispielfähig können Autos, Smartphones, aber auch Kaffeemaschinen, Spielzeug und weitere Geräte genannt werden. Viele dieser datenverarbeitenden Funktionen beruhen auf Informationen über den Status des Geräts, eine Historie der Nutzung und der Leistungsauslastung über den Nutzungszeitraum. Selbst wenn ein Gerät aktuell noch nicht jedes Detail überwacht, kann diese Fähigkeit einfach mittels Hardware und/oder Software, je nach Anforderung des Herstellers, ergänzt werden. (Allmendinger & Lombreglia, 2005)

Es zeigt sich, dass Smart Services folgende Anwendungen erfüllen können:

- Status
- Diagnose
- Aufrüstungen

- Kontrolle und Automatisierung
- Profil erstellen und Verhaltenstracking
- Nachschub und Handel
- Standort lokalisieren und Logistik

(Allmendinger & Lombreglia, 2005)

Beim Status können der bereits genannte Betrieb, die Leistungsauslastung und die Nutzung überwacht werden. Diagnoseanwendungen sorgen dafür, dass Geräte sich selbst optimieren können, oder dem Menschen ermöglichen, einen Service, im Sinne der Überwachung, Fehlerbehebung oder Reparatur auszuführen. Das Aufrüsten dient einer Upgrade-Anwendung, um die Leistung zu verbessern. Dabei können Versionskontrollen, veraltete Technologien und das Ausfallen von Geräten verhindert werden. Bei der Kontrolle und Automatisierung tragen Anwendungen zum Steuern und Automatisieren dafür Sorge, dass Aktivitäten koordiniert und ausgeführt werden. Die Anwendungen für Profilerstellung und Verhaltenstracking überwachen Standorte und deren individuelle Aspekte, eine Kultur, die Leistung, das Nutzen und den Verkauf von Geräten. Dadurch können individueller und vorausschauender Ergebnisse für einzelne Kund/innen bereitgestellt werden. Die Anwendung für Nachschub und Handeln befasst sich mit der Verbrauchsüberwachung und dem Kaufverhalten. Diese Anwendungen sind spezialisiert auf das Ausführen von Transaktionen, beispielsweise dem Nachbestellen. Bei der Anwendung zum Lokalisieren eines Standortes und der Logistik, dienen Smart Services als Unterstützung für das Service-Support-System eines Geräts. Bei dieser Anwendung werden ebenfalls Logistikvorgänge und Vertriebsaktivitäten mit erfasst. (Allmendinger & Lombreglia, 2005)

Dennoch sehen viele Unternehmen noch immer die einzelnen Objekte als individuelle Informationsträger und weniger deren Vernetzungspotential (Allmendinger & Lombreglia, 2005). Mittels dem Synergieeffekt, der aus mehreren Geräten in Kombination miteinander gewonnen werden kann, könnten Unternehmen zusätzliche Informationen gewinnen, um den Mehrwert durch Smart Services weiter zu steigern. Denn über Unternehmensgrenzen hinweg könnten Leistungsangebote stärker auf die Kund/innen zugeschnitten werden. Diese Tatsache sorgt auch für einen höheren Bedarf an Agilität der Lösungen. (Bullinger et al., 2017)

Im Wesentlichen zeigt sich, dass die präemptive Handlungsform als eine weitere Stufe vom ursprünglichen Smart Space entstanden ist, und durch Weiterentwicklung dieser Domäne durch Unternehmen und Wissenschaft den heutigen Stand des Smart Services erreicht hat. Smart Services sind somit erweiterte Möglichkeiten, um die Bedürfnisse der Kund/innen zu decken. Hierbei werden große Mengen an Daten genutzt, um einen Mehrwert aus den Informationen zu sammeln und sich des eigentlichen Ansatzes der präemptiven Handlungsform zu Nutze zu machen, um Lösungen bereitzustellen, bevor sich die Kund/innen über deren Bedarf gewahr ist.

## 2.4 Machine Learning

*„Artificial intelligence is the study of the computations that make it possible to perceive, reason, and act.“*

*(Winston, 1993, S. 5)*

Machine Learning ist ein Gebiet der Informatik (Chowdhury et al., 2017). Im Speziellen kann es der künstlichen Intelligenz, oder auch Artificial Intelligence (AI) genannt, zugeordnet werden. Wobei Data Science, vor allem Data Mining und Machine Learning als Verfahren, zum Auswerten großer Datenmengen gesehen werden. Des Weiteren wird Data Science in Kombination mit statistischen Methoden auch für Business Analytics genutzt. (Haneke et al., 2019)

Statistik unterscheidet sich von Machine Learning durch beispielsweise das Datenvolumen, da durch Machine Learning tendenziell Millionen von Daten verarbeitet werden können (Chollet, 2018). Machine Learning ist somit selbst, vereinfacht gesagt, eine Form der künstlichen Intelligenz, die es den Software-Entwicklern ermöglicht, Programme auf einfachere Weise zu schreiben. Sie konzentriert sich auf die Entwicklung von Programmen, die es ermöglichen neuartige Daten zu klassifizieren und zu interpretieren. (Chowdhury et al., 2017) Daher befasst sich Machine Learning mit einer gesammelten Datenmenge und versucht anhand dieser ein Modell zu entwickeln (Burkov, 2019). Das Ziel des Programmes ist es, die Methoden zu verstehen und ihnen zu folgen, indem Methoden und Algorithmen verwendet werden, die diese Aufgabe automatisch und ohne menschliche Hilfe erledigen (Chowdhury et al., 2017). Im Grunde kann Machine Learning als Mechanismus zur Mustersuche und zum Intelligenzaufbau von Maschinen gesehen werden (Gollapudi & Laxmikanth, 2016). Durch Machine Learning wird also eher trainiert als explizit eine Lösung programmiert (Chollet, 2018). Samuel (1959) definierte, dass Computer die Fähigkeit besitzen, von sich selbst ohne explizite Programmierung zu lernen. Dies ermöglichte immer komplexere Muster und auch versteckte Datenmuster zu erkennen (Chowdhury et al., 2017). Dennoch ist es wesentlich, dass im Lernprozess auch der Bedarf nach Daten, Zeit und dem erforderlichen Platz beachtet wird (Gollapudi & Laxmikanth, 2016). Mittlerweile können sich Modelle automatisch an größere und komplexe Datensätze anpassen, um unterschiedliche Szenarien in Echtzeit abzubilden und so die datengestützte Entscheidungsfindung zu stärken. Machine Learning befindet sich weiterhin in einer Verbesserung und Innovation durch die Forschung und der Wirtschaft, um dessen Effizienz zu steigern. (Chowdhury et al., 2017) Daher sollte ein Ziel von Machine Learning Algorithmen sein, dass diese auf eine möglichst breite Lernproblematik angewendet werden können (Gollapudi & Laxmikanth, 2016).

Machine Learning wird derzeit bereits in vielen Domänen eingesetzt. Netflix setzt Machine Learning ein, um die maximal mögliche Qualität des Streams der Benutzer/innen während des laufenden Streamings vorherzusagen und entsprechend die Datenraten anzupassen (Chaitanya Ekanadham, 2018). Ebenfalls profitiert das Transportsystem von Machine Learning, da Muster von Verkehrssituationen besser in Echtzeit analysiert werden, um die Effizienz der Verkehrssystemabläufe zu verbessern oder zukünftige Trends zu erkennen, und des Weiteren auch, um Rettungseinsätze optimierter gestalten zu können. Machine Learning findet auch bei autonomen Fahrzeugen Einsatz, da über die Sensorik des Fortbewegungsmittels Daten

gesammelt und durch Machine Learning analysiert werden, um so Steuerungsentscheidungen zu treffen. (Chowdhury et al., 2017)

Eine erste Definition für künstliche Intelligenz wurde bereits 1955 auf einer Konferenz in Dartmouth genannt (McCarthy et al., 1955). Es wurden bereits komplexe Algorithmen zur Mustererkennung, mitunter für künstliche Intelligenz, erfunden, bevor sich der Begriff Machine Learning selbst als eigenes Gebiet geformt hat (Chowdhury et al., 2017). Hierbei wurde 1959 von Arthur Samuel das Machine Learning als ein Forschungsgebiet definiert, das Computern die Fähigkeit verleiht, zu lernen, ohne explizit programmiert zu werden. Dabei stellte er Programmierungsmethoden vor, bei denen sich das Programm verhält wie ein Lernprozess, vergleichbar mit denen von Menschen oder Tieren. (Knuth, 1990) Aufgrund von, zur damaligen Zeit, nicht überwindbaren Problemen, beispielsweise die Hardwarelimitierung, wurde das Thema der künstlichen Intelligenz im sogenannten „AI-Winter“ der 1980er-Jahre pausiert. Derzeit ist die künstliche Intelligenz wieder ein aktuelles Thema, mitunter durch stärkere Rechenleistung, neuer Ansätze in der Forschung und das Phänomen Big-Data, welches durch die Datenflut des 21. Jahrhunderts erforderlich wurde. (Haneke et al., 2019) Daher umfasst Machine Learning ein breites Spektrum, wobei deren Überblick, Elemente, komplementäre Felder, Subfelder, Problemkategorien sowie die dafür erforderlichen Technologien, Werkzeuge und Frameworks beachtet werden müssen (Gollapudi & Laxmikanth, 2016).

Aufgrund der breiten Möglichkeiten der künstlichen Intelligenz ist eine konkrete Definition herausfordernd, wobei der Ansatz von Winston (1993) betreffend der Wahrnehmung, Schlussfolgerung und Handlungsermöglichung durch Berechnungsverfahren eine Basiserklärung dazu sein kann (Winston, 1993). Gollapudi und Laxmikanth (2016) haben sich ebenso mit den möglichen Definitionen für Machine Learning befasst. Dabei stellten sie fest, dass diese Definitionen algorithmisch, statistisch oder mathematisch betrachtet werden. Machine Learning kann also vereinfacht als Lösung für praktische Aufgaben betrachtet werden (Burkov, 2019). Zum Verständnis der Komplexität, die mit Machine Learning einhergeht, kann Abbildung 10 einen Überblick verschaffen.

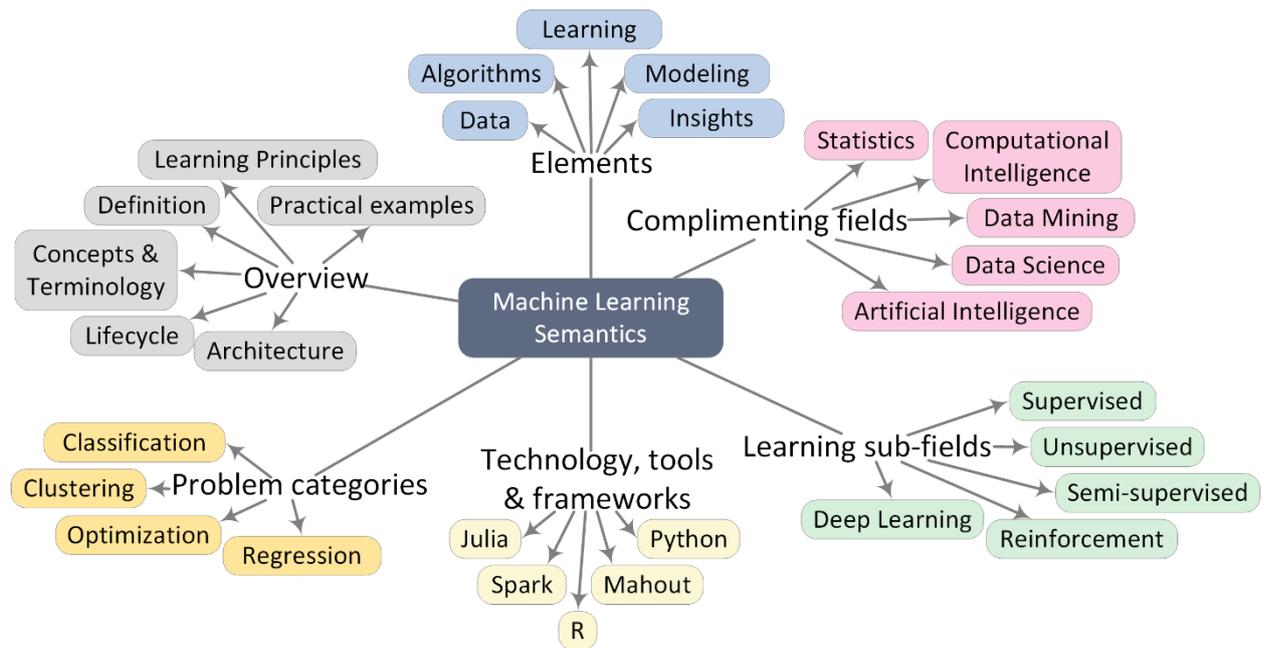


Abbildung 10: Machine Learning Schlüsselaspekte und Semantik (vgl. Gollapudi & Laxmikanth, 2016, S. 2)

Machine Learning versucht im allgemeinen Rohdaten zu betrachten, und diese in einer nützlicheren Repräsentation darzustellen (Chollet, 2018). Dabei werden Rohdaten bereits unterschieden nach gelabelt und ungelabelt (Gollapudi & Laxmikanth, 2016). Diese Kategorisierung, oder auch Clustering genannt, versucht Daten in eine logische Gruppierung aufzuteilen (Russell & Norvig, 2012). Beispielhaft können Fotos nach deren Darstellung gruppiert werden, ob ein Tier, Baum oder Hochschule darauf abgebildet sind (Gollapudi & Laxmikanth, 2016). Häufig werden gelabelte Daten als binäre Klassifizierung erkannt. Dafür kann als Beispiel genannt werden, ob die Fahrgeschwindigkeit eines Autos über oder nicht über dem Verkehrslimit lag. (Chowdhury et al., 2017) Für solche Klassifizierungen wurden in den letzten 50 Jahren bereits Erfahrungen gesammelt, um zu erkennen, wieviele Trainingspunkte ein Algorithmus benötigt, um einen möglichst genauen Klassifikator zu lernen (Sammut & Webb, 2017). In Abbildung 11 werden gelabelte Daten gezeigt. Diese werden für eine deutlichere Repräsentation in Abbildung 12 exemplarisch als neue Unterteilung der Daten dargestellt (Chollet, 2018). Anhand dieser Beispiele zeigt sich bereits der Zugang inklusiver Klassifizierter Daten, wobei die Rohdaten bereits klar einzelnen Klassen zugewiesen sind und als Ausgabe nach einer sinnvollen Aufteilung dazu gesucht wird (Russell & Norvig, 2012). Hingegen ist in Abbildung 13 eine Mischform aus gelabelten und ungelabelten Daten dargestellt (Gollapudi & Laxmikanth, 2016). Generell ist es einfacher, ungelabelte Daten zu beschaffen, da für die Klassifizierung von Daten im Vorhinein Aufwand betrieben werden muss (Sammut & Webb, 2017).

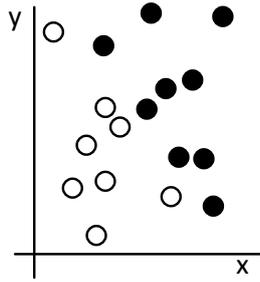


Abbildung 11: Exemplarische Rohdaten mit Labeling (vgl. Chollet, 2018, S. 7)

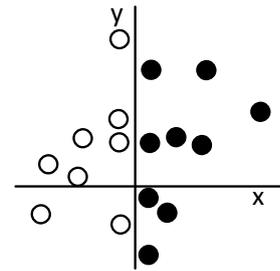


Abbildung 12: Exemplarisches Ergebnis zur besseren Repräsentation von gelabelten Daten (vgl. Chollet, 2018, S. 7)

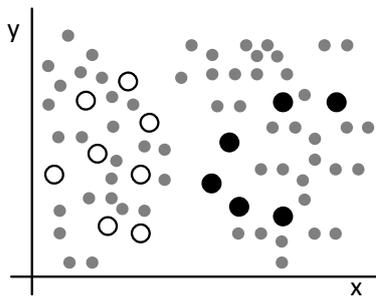


Abbildung 13: Darstellung einer Mischform aus gelabelten und ungelabelten Daten (in Anlehnung an Gollapudi & Laxmikanth, 2016, S. 8)

Für die nützlicheren Repräsentation von Daten werden verschiedene Algorithmen genutzt (Chollet, 2018). Algorithmen zielen darauf ab, dass sie durch Lernen eine Regel aufstellen können, die möglichst genau zutrifft (Gollapudi & Laxmikanth, 2016). Diese Algorithmen sind spezifisch für unterschiedliche Anwendungsgebiete zielführend (Russell & Norvig, 2012). Machine Learning kann in einzelne Subfelder unterteilt werden, die sich mit unterschiedlichen Typen von Algorithmen befassen, wie von Gollapudi und Laxmikanth (2016) genannt und in Abbildung 14 veranschaulicht.

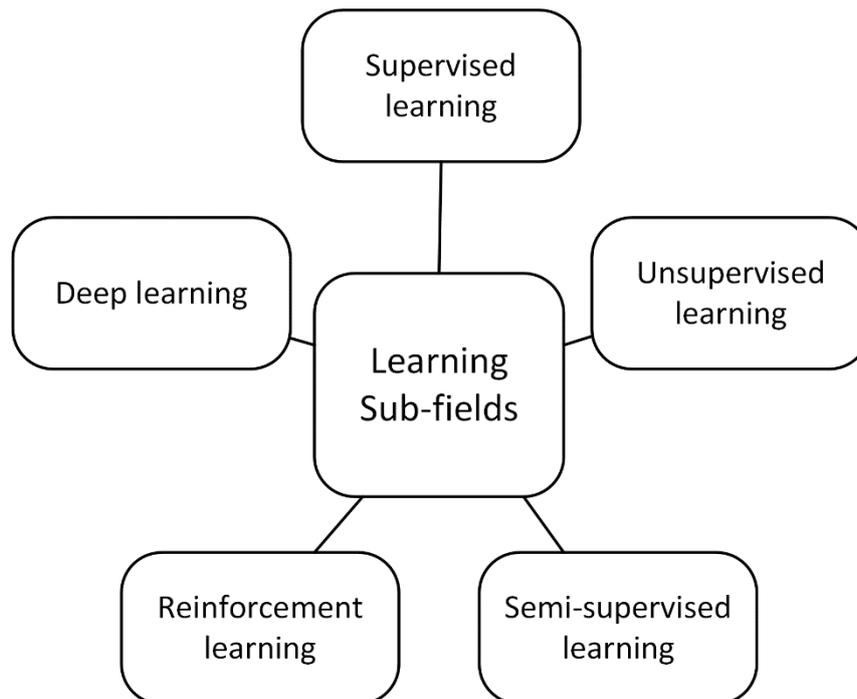


Abbildung 14: Subfelder des Machine Learning (vgl. Gollapudi & Laxmikanth, 2016, S. 21)

Beim Supervised Learning wird mit einer bekannten Erwartung an die Ausgabe gearbeitet (Chowdhury et al., 2017). Algorithmen dieses Subfeldes befassen sich mit der Beziehung zwischen Eingabe und Ausgabe. Beziehungen zwischen diesen beiden Attributen werden spekulativ vom Algorithmus verwendet, um für neue Eingabedaten eine Aussage zu deren Ausgabedaten zu liefern (Gollapudi & Laxmikanth, 2016). Beispielhaft kann genannt werden, dass Bremsaktionen für einen Bus anhand von erfassten Stoppsituationen erlernt werden, und dies auf zukünftige Szenarien, in denen ebenfalls Stoppsituationen erkannt werden, angewendet werden (Russell & Norvig, 2012). Der Schlüssel in diesen Algorithmen liegt somit im Mapping zwischen der Eingabe und der Ausgabe (Gollapudi & Laxmikanth, 2016). Aus diesem Grund werden für Algorithmen dieses Subfeldes Daten vorab klassifiziert, um eben stopprelevante Situationen vorab zu markieren, damit diese dann im Lernprozess für neue Datensätze identifiziert werden können (Chowdhury et al., 2017).

Beim Unsupervised Learning gibt es keine definierte Zielvorgabe für die Ausgabe (Gollapudi & Laxmikanth, 2016). Hierbei wird anhand der Eingabedaten geprüft, welche Logik draus erkannt werden kann (Russell & Norvig, 2012). Dies befasst sich mit Fragen, die generell für zielführende Cluster nötig sind (Burkov, 2019). Es sollen bei diesen Algorithmen versteckte Muster in den Daten erkannt werden, um daraus ein Modell für ein Clustering oder eine Assoziierung zu erfassen (Chowdhury et al., 2017). Es soll somit die Struktur in den Eingabedaten selbst geprüft werden, um deren Attribute zu entschlüsseln und mögliche Zusammenhänge oder Ähnliches zu erfassen. Daher werden für die Algorithmen in diesem Subfeld Daten ohne vorhergehende Klassifizierung genutzt (Gollapudi & Laxmikanth, 2016).

Beim Semi-Supervised Learning erfolgt eine Mischform der zuvor genannten beiden Subfelder (Gollapudi & Laxmikanth, 2016). Die Daten sind zum Teil klassifiziert, zum Teil ohne Cluster

(Burkov, 2019). Es kann auch angenommen werden, dass das vorhandene Clustering nicht zwingend korrekt ist (Russell & Norvig, 2012). Beispielhaft kann genannt werden, dass Personen nach ihrem Alter gefragt werden, und ein Foto von ihnen gemacht wird, um anhand der Bilder einen Algorithmus zu trainieren, und das Alter von Personen zu errechnen. Es kann sein, dass Personen in der Befragung ein korrektes oder ein falsches Alter genannt haben. Daher ist es wichtig, dass in solchen Fällen, in denen die Glaubwürdigkeit der vorhandenen Cluster nicht gegeben ist, auf eine Semi-Supervised Learning zurückgegriffen wird, um solche Fälle erfassen zu können (Russell & Norvig, 2012). Diese Art des Lernens bezieht seine Motivation bereits mehr aus der menschlichen Lernmethode (Gollapudi & Laxmikanth, 2016).

Reinforcement Learning nutzt den menschlichen Zugang, dass durch Belohnung und Bestrafung das Lernen optimiert werden kann (Gollapudi & Laxmikanth, 2016). In diesem Subfeld wird davon ausgegangen, dass eine Umgebung definiert wird, innerhalb der das maschinelle Lernen erfolgt (Burkov, 2019). Dies kann beispielhaft so verstanden werden, dass Taxifahrer/innen am Tagesende einen Verlust in ihrer Abrechnung haben, somit einen Fehler in ihren Berechnungen hatten (Russell & Norvig, 2012). Ähnliches zeigt sich, wenn Kleinkinder für positive Ereignisse gelobt werden, um dies zur Gewohnheit werden zu lassen (Gollapudi & Laxmikanth, 2016). Besonders Logistikfragen können mit Reinforcement Learning behandelt werden, da das langfristige Ziel wesentlich ist (Burkov, 2019). Somit werden Daten mit positiven und negativen Verstärkern versehen, wobei dem Algorithmus selbst überlassen ist, welche Kombination aus Verstärkern ihn am Ende zu bestmöglichen Ausgaben bringt (Gollapudi & Laxmikanth, 2016).

Das letzte Subfeld behandelt das Deep Learning. Hierbei wird maschinelles Lernen und AI vereint, wobei es sich um eine Weiterentwicklung von künstlichen neuronalen Netzwerken handelt, die mit großen Datenmengen umgehen können (Gollapudi & Laxmikanth, 2016). In diesen Algorithmen werden durch Aufeinanderichten von Daten eine repräsentative Aussagekraft gewonnen (Chollet, 2018). Es werden somit aus allgemeinen Daten praktische Ausgaben abgeleitet (Gollapudi & Laxmikanth, 2016). Durch dieses Vorgehen wird versucht herausragende Kennwerte aus den Daten zu erfassen und den Datensatz auf wesentliche Merkmale zu reduzieren (Chollet, 2018). Deep Learning befasst sich mit komplexen neuronalen Netzen, um daraus Lösungen für Probleme zu erarbeiten (Gollapudi & Laxmikanth, 2016).

## 2.5 Grundlegende Zusammenhänge zwischen Smart Home und Smart Service

*“Merely being convenient was not sufficient for automation to be considered “smart”.”*

*(Mennicken & Huang, 2012, S. 152)*

Anhand der Erläuterungen zum Smart Service (siehe Abschnitt 2.3) und dem Wesen des Smart Homes (siehe Abschnitt 2.2) zeigt sich, dass es in deren beiden Vorgehen Überschneidungen gibt. Der primäre Ansatz des Smart Homes ist, dass Hausautomatisierung ermöglicht wird (Mennicken et al., 2014). Dieser Ansatz deckt sich mit den Ursprüngen des Smart Services in dessen Form als Smart Spaces. Bei deren Vorgehen eine reaktive Handlungsform vorherrschend war (Acatech - Deutsche Akademie für Technikwissenschaften, 2015). Dennoch schien im Unternehmenskontext ein reines reaktives Vorgehen nicht ausreichend, wodurch mittels Sensorik einerseits begonnen wurde ein großes Volumen an Daten zu sammeln, sowie diese Daten zu nutzen. Mitunter durch den Einsatz von Machine Learning (Allmendinger & Lombreglia, 2005). Durch diese Umsetzung können mittlerweile präemptive Handlungsformen, sogenannte Smart Services, den Bedarf in Unternehmen besser decken (Acatech - Deutsche Akademie für Technikwissenschaften, 2015; Allmendinger & Lombreglia, 2005). Somit zeigt sich bereits im Kontext zu Unternehmen, dass eine präemptive Handlungsform existiert und auf den aktuellen Megatrend, der Digitalen Transformation aufbaut.

Gegenüber der weiterentwickelten Form für Smart Services im Unternehmenskontext zeigt sich bei Smart Home, dass dessen Verständnis gestreut ist. Viele Menschen verstehen unter einem Fernzugriff auf das Eigenheim bereits ein Smart Home, auch wenn nicht einmal der Automatisierungszustand darin involviert ist. Seitens der Forschung wird jedoch genannt, dass für die Definition „Smart“ ein System auch auf seine Benutzer/innen reagieren soll und autonom in hochentwickelter Weise anpasst sein soll. (Mennicken et al., 2014) In erweiterter Form sollte auch über die reine Automatisierung eines Smart Homes hinausgegangen werden. Dabei ist das Ziel des Systems, die ihm verfügbaren Daten, selbstlernen zu nutzen (Risteska Stojkoska & Trivodaliev, 2017) Dies kann als Prozess verstanden werden, indem ein System die Aktionen von Benutzer/innen überwacht und daraus Muster zur Vorhersage der Aktionen in der Umgebung sucht, und das Interagieren mit beziehungsweise anhand dieser (Harper, 2003). Diese Vorstellung stellt eine erweiterte Form zu der ursprünglichen reinen Automatisierung des Alltags dar, indem Aktionen für Benutzer/innen automatisch ausgeführt werden (Mennicken et al., 2014). Daraus können auch personalisierte Empfehlungen für Bewohner/innen erstellt werden (Risteska Stojkoska & Trivodaliev, 2017). Ursprünglich war somit eine Erkennung der Einstellungen je Benutzer/in ausreichend, zum Beispiel der Helligkeits- und Wärmegrad von Licht (Harper, 2003). Es entwickeln sich jedoch auch die Erwartungen der Benutzer/innen weiter. Diese Entwicklung entsteht dadurch, dass Technologien immer mehr Möglichkeiten zur Erleichterung im Alltag aufweisen. Daher wird auch die Vorstellung an das, was ein Smart Home ausmacht, kontinuierlich erweitert (Mennicken et al., 2014). Bereits Balta-Ozkan et al. (2014) benannten ein Smart Home als Kommunikationsnetzwerk, dass in einem Zuhause verschiedene Sensoren, Geräte und Haushaltsgegenstände verbindet, und neben der Überwachung und Steuerung auch Dienste zur

Deckung der Bedürfnisse der Bewohner/innen ermöglicht. Das Ziel ist also es zu ermöglichen, dass ein Smart Home Tätigkeiten besser ausführen kann als ein Mensch. Andernfalls wird die Beschreibung Smart nicht als passend empfunden (Mennicken & Huang, 2012). Um diesen Schritt zu ermöglichen, kann Machine Learning zur Verbesserung der reinen Automatisierung beitragen (Gram-Hanssen & Darby, 2018). Machine Learning Algorithmen selbst können auch als hochentwickelte autonome Lernmodelle angesehen werden (Gollapudi & Laxmikanth, 2016). Dabei sollen Machine Learning Algorithmen genutzt werden, um eine Vorhersage zu ermöglichen, die ein autonomes Steuern möglich machen (Mennicken & Huang, 2012). Ein Smart Home soll sich, wie auch in der Forschung genannt, zu einem hochentwickelten und autonomen System verändern. Diese Entwicklung soll über eine reine Automatisierung hinausgehen (Mennicken et al., 2014).

Es gilt somit, dass sich das Smart Home von einer reinen Automatisierung hin zu einem autonomen und hochentwickelten System erweitern soll. Daher scheint ein ähnliches Vorgehen, wie es bereits bei Smart Services genutzt wird, mittels präemptiver Steuerung im Unternehmensbezug, naheliegend zu sein. Aufgrund des Zusammenhangs von Smart Service und Smart Homes und dem weiteren Fortschritt des Smart Spaces zu einem Smart Service, scheint der Zugang naheliegend, die bereits im Smart Service eingesetzten Methoden ebenfalls für ein Smart Home zu nutzen. Im Konkreten geht es um das Einsetzen von Machine Learning Algorithmen zur präemptiven Steuerung auch im Kontext zum Smart Home.

## **3 UNTERSUCHUNGSDESIGN**

Im Nachfolgenden wird einerseits ein Ziel für empirische Softwareentwicklung in der Wirtschaftsinformatik aufgezeigt, andererseits wird das Vorgehen im Detail beschrieben. Es wird außerdem die Forschungsentscheidungsstruktur erarbeitet, wobei sich konkret das Prototyping als Methode für diese Arbeit zeigt. Hierzu wird auf den Cross Industry Standard Process for Data Mining (CRISP-DM) zurückgegriffen, der methodisch um weitere Einflüsse anderer Modelle, die teils spezifisch für Machine Learning sind, erweitert wird. Abschließend wird protokolliert, mit welcher Hardware, Tools und Software in dieser Arbeit vorgegangen wurde.

Ziel dieses Kapitels ist es, ein methodisches Verständnis für das Vorgehen zu entwickeln. Dabei wird auf eine empirische Herangehensweise in der Softwareentwicklung geachtet und auch die Phasen des definierten Vorgehens erläutert. Anhand dieses Vorgehensmodell werden im nachfolgenden Kapitel die Ergebnisse dazu aufgezeigt.

### **3.1 Ziel empirischer Softwareentwicklung**

Die Entscheidungsstruktur zur Wahl eines Untersuchungsdesign in der empirischen Softwareentwicklung ist besonders herausfordernd (Wohlin & Aurum, 2015).

Die Wirtschaftsinformatik findet sich in der Real-, Ingenieur- und Formalwissenschaft wieder (Schoder et al., 2009-2011). Aufgrund dieser breiten Fächerung der Möglichkeiten für Forschung in der Wirtschaftsinformatik, zeigt sich, dass es wesentlich ist, eine passende Wahl für die Methode seines Untersuchungsdesigns zu wählen (Wohlin & Aurum, 2015). Mithilfe eines Fokus auf die passende Methode, erscheint anhand dieser auch die Wahl der Werkzeuge zielführender (Wohlin & Aurum, 2015). Daher ist es wichtig, sich vorab mit den Optionen zu befassen. Hierbei zeigt sich nach Wilde und Hess (2006), dass die Methoden nach ihrem Formalisierungsgrad und Paradigma zugewiesen werden können, wie in Abbildung 15 veranschaulicht.

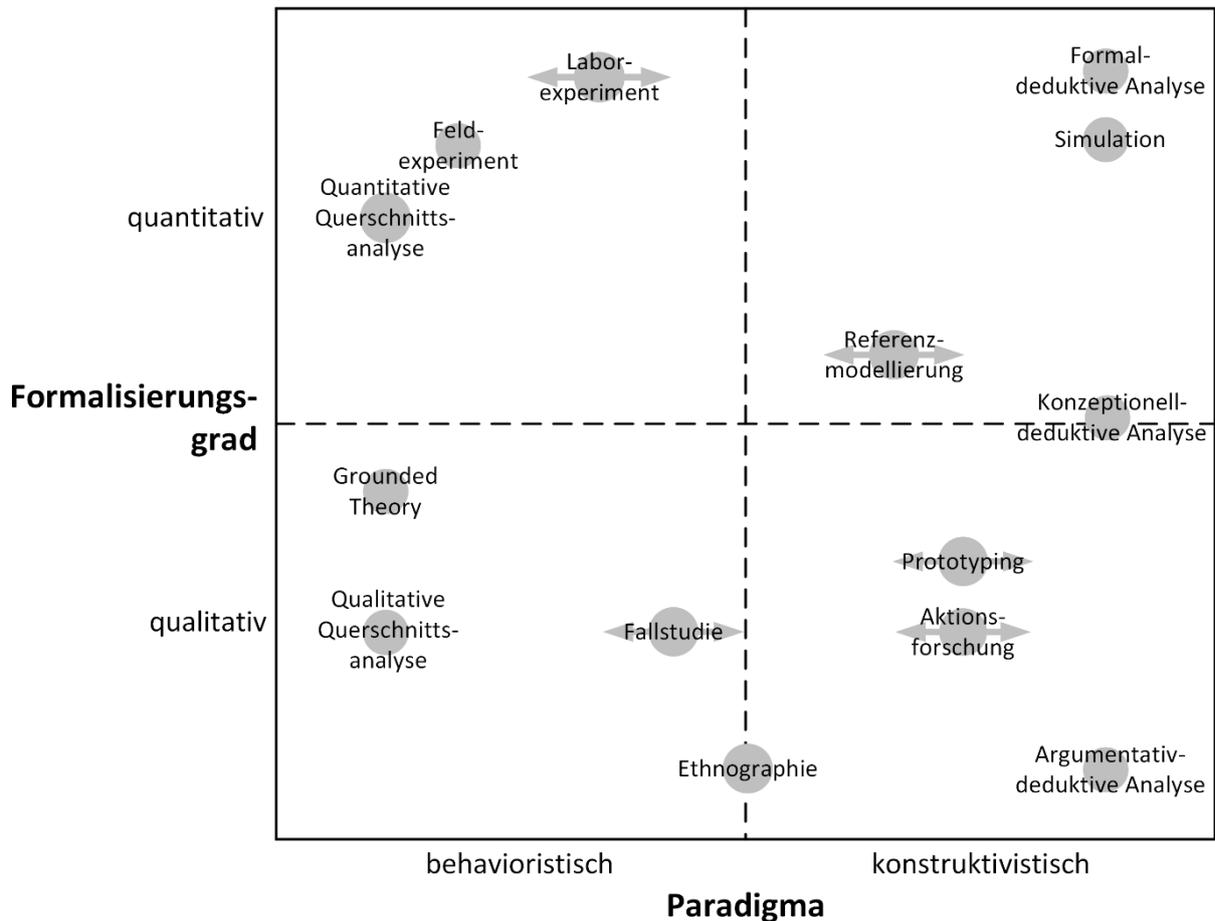


Abbildung 15: Einordnung der Methoden im Portfolio der Wirtschaftsinformatik (vgl. Wilde & Hess, 2006, S. 14)

Für die Methoden der Wirtschaftsinformatik kann übergeordnet gesagt werden, dass diese aus den Komponenten Prozess, Realitätsabbildung und der Analyse bestehen. Zum einen weisen Verhaltenswissenschaften eine Gliederung von Forschungsdesign, Operationalisierung/ Erhebung und Analyse auf, zum anderen bestehen die konstruktivistischen Methoden in der Wirtschaftsinformatik aus den folgenden drei typischen Komponenten:

- Vorgehensmodell
- Modellierungsregeln
- Analyse

(Wilde & Hess, 2006)

Nach der Untersuchung der Forschungsmethoden von Wilde und Hess (2007) zeigt sich, dass die Wirtschaftsinformatik einen konstruktiv-qualitativen Schwerpunkt hat. Aus diesem Grund wurde der Fokus für diese Arbeit auf den Quadranten rechts unten in der Abbildung 15 gelegt. Hierbei ist wesentlich zu nennen, dass bereits im Abschnitt 2.5 ein Zusammenhang zwischen Smart Homes und Smart Service, welcher auch mittels Machine Learning gelöst werden kann, genannt wurde. Diese präemptiven Handlungsformen des Smart Services gilt es daher zu prüfen, und somit auch, ob der Machine Learning Ansatz der Smart Services auch für Smart Homes

angewendet werden kann. Dabei wurde geprüft, welche vergleichbaren Vorgehen zur Modellierung bereits genutzt wurden, und deren Ansatz wurde auf das Machine Learning für Smart Home angewendet. Im Abschnitt 2.5 wurde die Annahme eines Zusammenhangs zwischen Smart Homes und Smart Services getroffen. Es galt daher diese getroffene Annahme zu prüfen, indem ein allgemeines Argument für die Integration einer präemptiven Steuerung in einem Smart Home getroffen werden kann. Somit fokussiert sich diese Arbeit auf die Beantwortung der folgenden Forschungsfrage:

„Wie und in welchem Umfang kann Machine Learning zur präemptiven Steuerung in ein Smart Home integriert werden?“

Aus dieser Forschungsfrage lassen sich folgende Arbeitshypothesen ableiten:

H<sub>0</sub>: Eine präemptive Smart Home Steuerung kann nicht mittels Machine Learning umgesetzt werden.

H<sub>1</sub>: Eine präemptive Smart Home Steuerung kann mittels Machine Learning umgesetzt werden.

Aufgrund dieser Fragestellung war es relevant für diese Arbeit ein Proof-Of-Concept zu erstellen. Anhand dieser Betrachtung zeigte sich, dass die eingehend definierte Forschungsfrage und die dafür definierten Arbeitshypothesen einen deduktiven Ansatz erfordern, und damit konkret die Entwicklung und den Test von Prototypen. Daher wurde geprüft, nach welchen Entscheidungskriterien in der Forschung vorzugehen ist (siehe Abbildung 16). Anhand des bereits definierten deduktiven Ansatzes wurde auch eine Basisrecherche, betreffend der Ergebnisse, zugrunde gelegt. Da es zu prüfen galt inwiefern Machine Learning zur präemptiven Steuerung in ein Smart Home integriert werden kann, wurde ein explorativer Zweck angestrebt.

Da für den Machine Learning Ansatz Daten erforderlich waren, wurden hierfür Archivdaten genutzt. Mit diesen Daten wurde ein Artefakt als neues Modell zur präemptiven Steuerung eines Smart Homes geschaffen, das anhand von Requirements in Form von Use Cases abgebildet wurde. Innerhalb des Machine Learning Algorithmus konnte von einer thematischen Analyse ausgegangen werden, da der Charakter von Machine Learning Algorithmen ein selbstlernender ist. Mithilfe des Modells, welches durch den Algorithmus erstellt werden konnte, wurde die statistische Analyse der Ergebnisse des Algorithmus betrieben. Für dieses wissenschaftliche Vorgehen wurde angestrebt, dass speziell für Data Science und Machine Learning spezifische Prozesselemente herangezogen wurden, um den Fokus auf die Beantwortung der Forschungsfrage zu sichern.

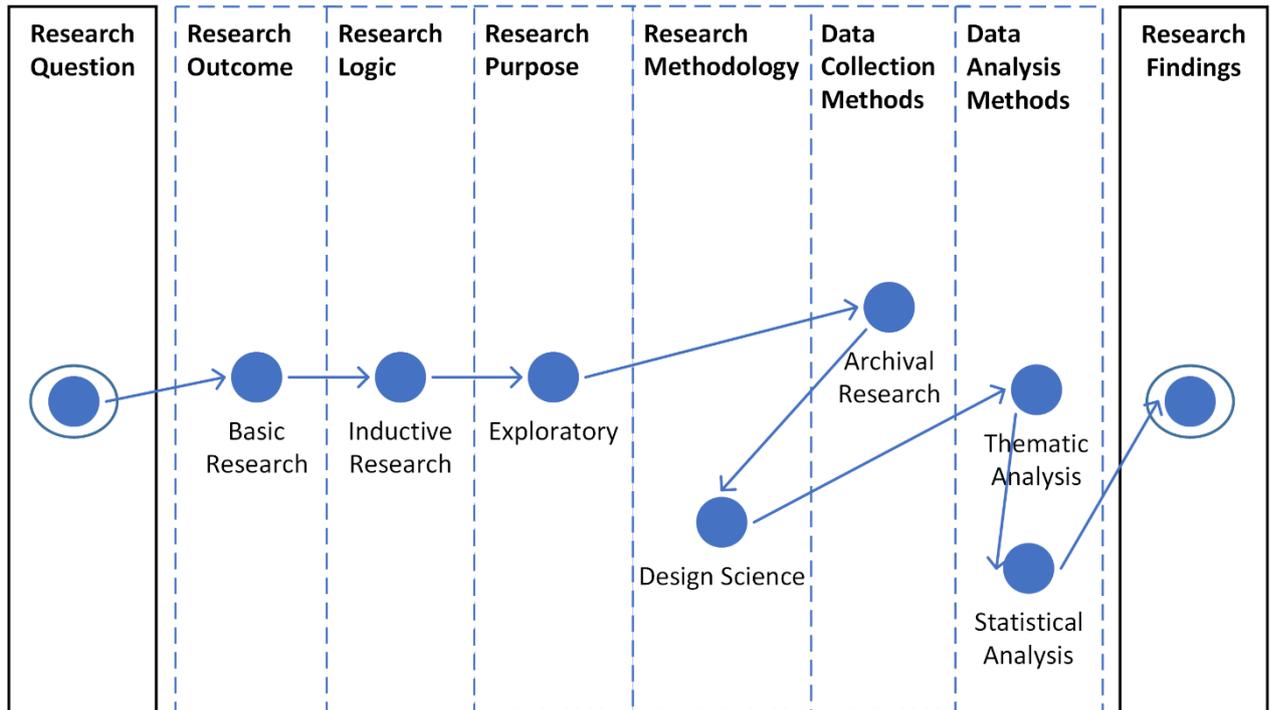


Abbildung 16: Forschungsentscheidungsstruktur (in Anlehnung an Wohlin & Aurum, 2015, S. 1431)

### 3.2 Methodik

Für das Vorgehen in dieser Arbeit wurde ein Prozess zur Entwicklung eines Machine Learning Algorithmus definiert. Als Ziel galt es, damit die Forschungsfrage zu beantworten. Es wurde darauf geachtet, dass die Methode des Design Science Ansatzes berücksichtigt wird. Daher wurde als Prozesskern auf die Phasen des CRISP-DM als Referenzmodell aufgebaut. Dessen Phasen befassen sich mit dem Geschäftsverständnis, Datenverständnis, Datenvorbereitung, Modellierung, Auswertung und dem Einsatz (Chapman et al., 2000).

In dieser Arbeit liegt der Fokus auf der Entwicklung eines Prototyps, wodurch die Phase "Einsatz" nicht umgesetzt wird. Dieses Grundmodell nach CIRSP-DM (siehe Abbildung 17) wurde um Einflüsse, die konkret auf Machine Learning und/oder Deep Learning zugeschnitten sind, erweitert (siehe Abbildung 18). Dabei wurden Methoden und Werkzeuge von Gollapudi und Laxmikanth (2016), Chowdhury et al. (2017), Zheng (2015) und spezifisch für die Daten auch von Fayyad et al. (1996) genutzt, da von Haneke et al. (2019) den Knowledge Discovery in Databases (KDD) Prozess von Fayyad et al. (1996) als Teilelement von AI sehen.

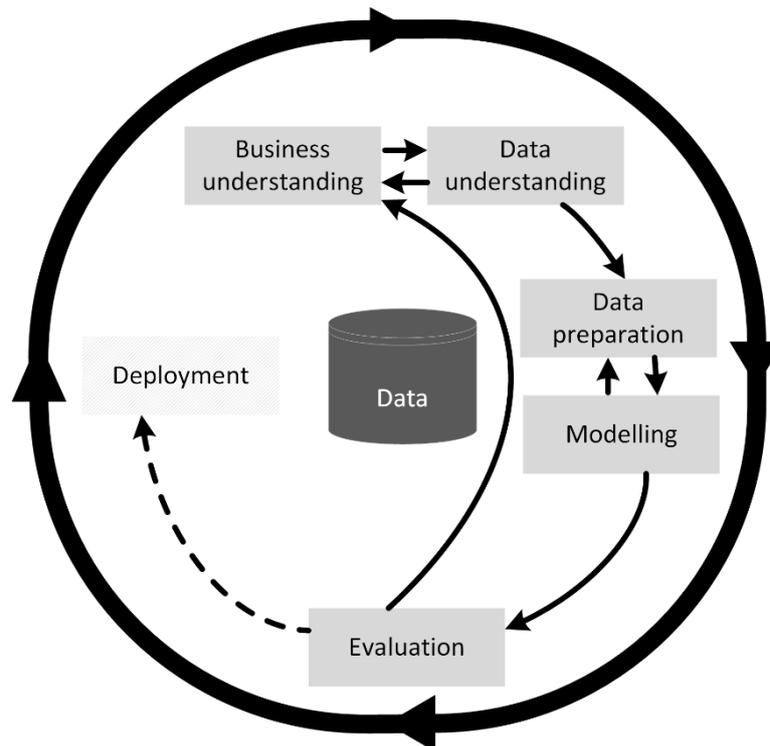


Abbildung 17: CRISP-DM Referenzmodell (in Anlehnung an Chapman et al., 2000, S. 13)

Im CRISP-DM Referenzmodell wurde für das Geschäftsverständnis genannt, dass die Projektobjekte und Requirements bekannt sein müssen (Chapman et al., 2000). Erweitert dazu wurde auch beachtet, dass die Problemstellung vorab definiert ist. Mit diesem Grundwissen wurde in der nächsten Phase, dem Datenverständnis, dann bereits eine Zielperspektive für Rohdaten genutzt. (Chowdhury et al., 2017) Diese Rohdaten waren historische Daten, also bestehende Informationen der Vergangenheit (Zheng, 2015) und damit Archivdaten nach Wohlin und Aurum (2015).

In dieser Phase sollte ein Verständnis für den Datensatz und dessen mögliche Qualitätsthemen geschaffen werden (Chapman et al., 2000). Dies wurde bereits im KDD-Prozess erfasst, da auch dort ein Verständnis für die Daten anhand der Anforderungen der Domäne zu prüfen ist, um mit dem Fokus auf passende Zieldaten zu gelangen (Fayyad et al., 1996). Die passenden Zieldaten sind eine Grundvoraussetzung für die weiteren Phasen (Chowdhury et al., 2017).

Diese Daten wurden in der Datenvorbereitung soweit aufbereitet, dass diese für eine weitere Modellierung bereit waren (Chapman et al., 2000). Besonders waren dabei mögliche fehlende Informationen, wie auch mögliches Rauschen in den Daten zu betrachten (Fayyad et al., 1996). Zudem wurden die Daten in erforderliche Segmente zerlegt, damit diese im Modell für unterschiedliche Anwendungen genutzt werden konnten (Zheng, 2015). Hierbei wurden die Daten konkret in Trainingsdaten und Validierungsdaten gesplittet (Chollet, 2018; Gollapudi & Laxmikanth, 2016; Zheng, 2015).

Die folgende Phase war bereits die Modellierung und die Wahl zum passenden Modell (Chapman et al., 2000). Dies war somit die eigentliche thematische Analyse, die auf quantitative Daten

angewendet wurde (Wohlin & Aurum, 2015). Hierbei erfolgte zuerst eine konkrete Auswahl für den anzuwendenden Machine Learning Algorithmus (Chowdhury et al., 2017). Dabei wurde, aufgrund der gewählten Daten und dem erforderlichen Ziel, anhand der Forschungsfrage Deep Learning Algorithmen genutzt (Chollet, 2018). Hierbei wurden die Algorithmen Long Short-Term Memory (LSTM) und Bidirectional Long Short-Term Memory (BLSTM) angewendet (Alahi et al., 2016; Chhetri et al., 2020; Géron, 2018; Hua et al., 2019; Wang et al., 2017; Zou et al., 2020).

Mit diesen Algorithmen wurde das eigentliche Training anhand der zugehörigen Daten durchgeführt, um ein Modell als Output zu erlangen (Gollapudi & Laxmikanth, 2016). Dabei wurden die Daten anhand des Machine Learning Modells von Zheng (2015) verarbeitet. Dazu wurde das erstellte Modell mit dem zuvor definierten Validierungsdatensatz getestet (Gollapudi & Laxmikanth, 2016). Konkret wurde hierbei die Hold-Out Validierung genutzt (Zheng, 2015). Ziel dieser Phase war es, ein möglichst valides Modell zu erlangen (Chapman et al., 2000; Fayyad et al., 1996).

Mit der nächsten Phase wurde das erstellte und auf Validität geprüfte Modell evaluiert (Chapman et al., 2000). In dieser Phase wurden statistische Analysen auf Mixed-Methods genutzt (Wohlin & Aurum, 2015). Hier wurde das finale Modell interpretiert, dabei konnte auch eine Visualisierung vom Modell und dessen Daten präsentiert werden (Fayyad et al., 1996).

Aus diesen Erweiterungen der Phasen nach CRISP-DM kann die folgende Mindmap (siehe Abbildung 18) gebildet werden.

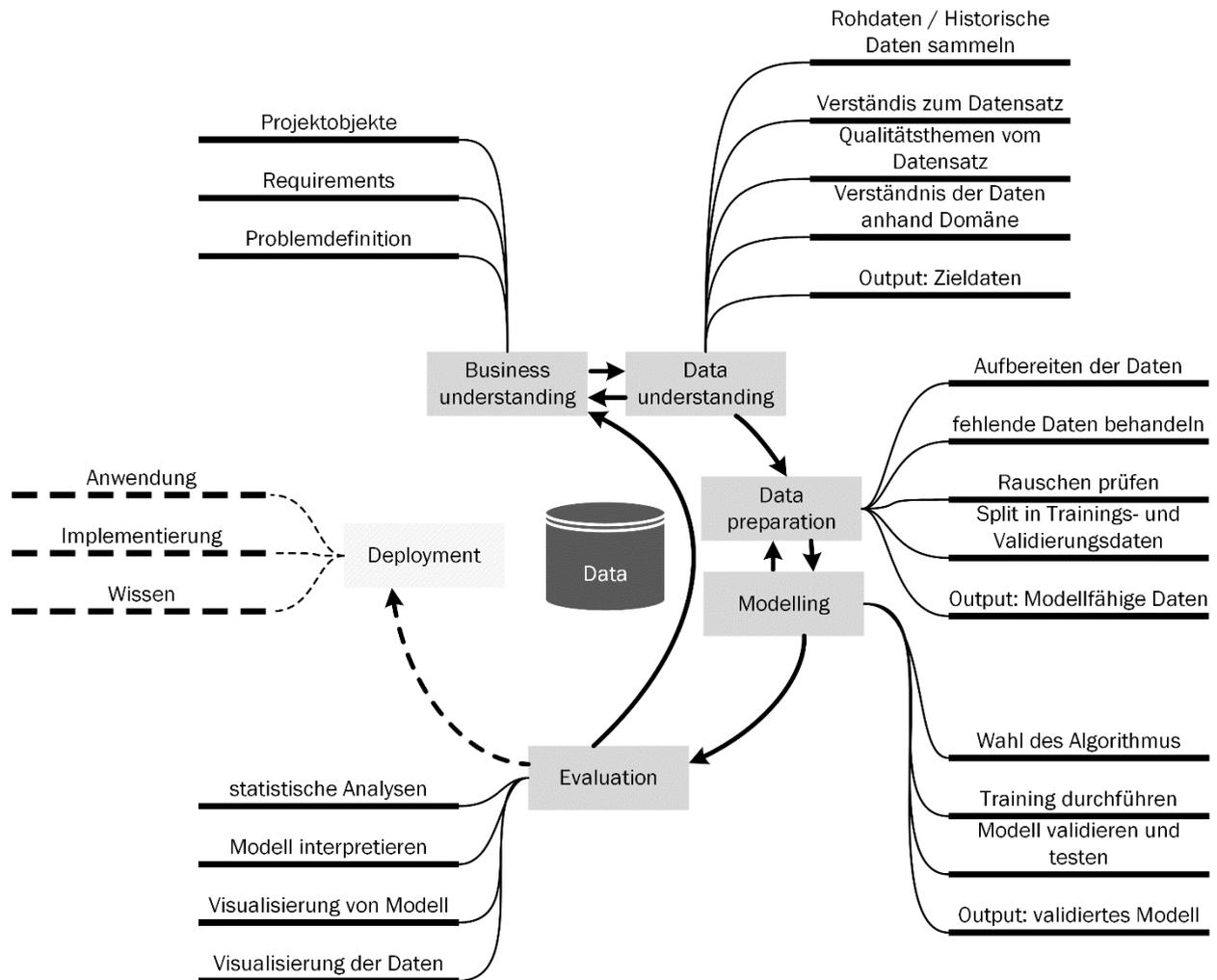


Abbildung 18: CRISP-DM Referenzmodell erweitert mit den zusätzlichen Methoden und Werkzeugen (in Anlehnung an Chapman et al., 2000, S. 13)

### 3.2.1 Geschäftsverständnis

Ein einleitendes Verständnis für das gestellte Problem ist ein wichtiger erster Schritt (Chowdhury et al., 2017). Nach CRISP-DM ist es erforderlich, dass als Output das Hintergrundwissen, die Projektobjekte und die Erfolgskriterien anhand von Requirements bekannt sind (Chapman et al., 2000). Ein ähnlicher Zugang wurde bereits beim klassischen KDD Prozess genannt. Hier wurde festgehalten, dass eingehend ein Verständnis für die Domäne des Anwendungsbereichs sichergestellt sein muss und das relevante Vorwissen zu entwickeln ist (Fayyad et al., 1996). Als Problemverständnis zeigte sich hierbei der Bedarf zur Klärung der nachfolgenden Forschungsfrage:

„Wie und in welchem Umfang kann Machine Learning zur präemptiven Steuerung in ein Smart Home integriert werden?“



Mit dem Vorwissen, dass der Use Case sich mit Planzeiten zum Einschalten befasste, war somit vorab von einem Inputdatensatz mit Zeitserien auszugehen, der für die weiteren Phasen spezifiziert wurde.

Betreffend der Requirements an ein Smart Home zeigt sich anhand von Studien, dass Bewohner/innen den Fokus auf Alltagserleichterung, steigenden Komfort und Energieeffizienz legen (Begleitforschung Mittelstand-Digital, 2016). Automatisierung des Alltags ist bereits bei Benutzer/innen vorhanden, hierbei zeigt sich in Deutschland, dass rund 18% angeben, ihre Schalter und Steckdosen, 10% ihre Heizung und 9% ihre Jalousien vernetzt haben (Deloitte Deutschland, 2018). Diese Themen entsprechen nach Marikyan et al. (2019) Smart Home Funktionen für die Kategorie Komfort.

Anhand dieser Recherche zeigt sich, dass die Möglichkeiten eines Smart Homes breit gefächert sind, die Frage zu präemptiver Steuerung allerdings noch zu klären ist.

Aus diesem Grund wurden im Rahmen dieser Arbeit Use Cases, aufbauend auf die Automatisierungsthemen, deren Umsetzung eine präemptive Steuerung erfordern, identifiziert. Hierbei handelt es sich um eine beispielhafte Liste. Tatsächliche Use Cases, welche im Feld angewendet werden sollen, hängen von Parametern, wie zum Beispiel der Art der im Smart Home verbauten Sensoren, den durch diese erfassten Daten, den verfügbaren Ressourcen für die Erstellung des Machine Learning Models, den persönlichen Anforderungen der Bewohner/innen, et cetera ab (Chowdhury et al., 2017).

#### **Use Case 1**

Passend zum Terminplan, Verkehrslage und benötigter Zeit für Morgen-Routine, soll morgens die Verdunklung automatisch geöffnet werden. (Begleitforschung Mittelstand-Digital, 2016; BITKOM, 2008; Deloitte Deutschland, 2018; Marikyan et al., 2019)

#### **Use Case 2**

Die Begrünung und Hauspflanzen sollen, je nach deren individuellem Bedarf und der Vergangenheit, aktuell und zukünftig vorherrschenden Wetterlage bewässert werden. (Begleitforschung Mittelstand-Digital, 2016; BITKOM, 2008; Marikyan et al., 2019)

#### **Use Case 3**

Die Heizung soll so gesteuert werden, dass wenn benötigt ausreichend Wärme vorhanden ist, jedoch Wohnräume nicht durchgehend beheizt werden. (Begleitforschung Mittelstand-Digital, 2016; BITKOM, 2008; Marikyan et al., 2019)

#### **Use Case 4**

Die Kaffeemaschine soll zu einem Zeitpunkt eingeschaltet werden, so dass der Kaffee fertig ist, wenn das Frühstücksritual begonnen wird. (Begleitforschung Mittelstand-Digital, 2016; Marikyan et al., 2019)

Aufgrund der starken Abhängigkeit vom Prototyp und den dafür nötigen Daten, die wiederum von den Requirements abhängen, wurde im Zuge dieser Arbeit konkret der Use Case 4 behandelt.

Aufbauend auf diese Grundlage als Geschäftsverständnis konnte zielgerichtet in den nachfolgenden Phasen für diesen Use Case geforscht werden.

### 3.2.2 Datenverständnis

In dieser Phase soll das Vorwissen aus dem Geschäftsverständnis genutzt werden, um Zieldaten zu finden (Chowdhury et al., 2017). Hierfür gibt es unterschiedliche Möglichkeiten, wobei Daten zum Beispiel aus einer Umgebung mittels Sensoren initial gesammelt werden können, Interviews, Umfragen oder aus bereits bestehenden Daten aus Forschungsarchiven genutzt werden können (Wilde & Hess, 2006).

Jedenfalls ist der erste Zielwert dieser Phase, dass Rohdaten gesammelt werden (Chowdhury et al., 2017). Seitens Zheng (2015) wird für das Training von Machine Learning Algorithmen darauf verwiesen, dass historische Daten genutzt werden sollten. Auch schon im KDD Prozess wurde genannt, dass ein Zieldatensatz festgelegt werden muss (Fayyad et al., 1996).

Wenn dieser Zieldatensatz definiert wurde, ist es ein wichtiger Schritt sich mit diesem zu befassen (Chapman et al., 2000). Denn für Machine Learning Algorithmen können die Qualität, der Typ und die Größe von Datensätzen wichtige Faktoren sein, um effizient eine genaue Vorhersage treffen zu können (Chowdhury et al., 2017). Hierbei kann auch erforderlich sein, die möglichen Datensätze vorab zu sichten, gegebenenfalls auch unterstützt durch verfügbare Tools (Chapman et al., 2000).

Im Zuge dieser Vorabsichtung der Datensätze ist es wesentlich, dass für Machine Learning Algorithmen auch Verfügbarkeit, Ressourcen und Zeit beachtet werden (Chowdhury et al., 2017). Die Ressourcen sind im Besondern auch bezüglich der verfügbaren Hardware wesentlich. Neben diesen Anforderungen ist auch eine Betrachtung von möglichen statistischen Werten relevant, um den Aufbau des Zieldatensatzes zu erfassen, dabei können Werte wie Mittelwert, Median, Modus, Varianzen, Standardabweichungen, Histogramme, Streudiagramme und Boxplots aufgezeigt werden (Chowdhury et al., 2017).

Mit diesen theoretischen Informationen erfolgte dann die Auswahl und Schaffung eines Verständnisses für den Datensatz. In der vorherigen Phase wurde festgehalten, welche konkreten Use Cases mit Machine Learning zur präemptiven Steuerung in ein Smart Home integriert werden sollten. Aufgrund dessen wurde in dieser Phase geprüft, welche Daten hierfür herangezogen werden können. Es wurde mittels einer Recherche, anhand definierter Schlagwörter, nach einem passenden Datensatz für diese Use Cases gesucht.

Es wurden unterschiedliche Datensätze gefunden und jene, die im Zuge dieser Arbeit zugänglich waren, wurden auch gesichtet, wie nachfolgend gelistet:

- Multi-sensor dataset of human activities in a smart home environment (Chimamiwa, 2020)
- UCAMl Cup 2018 dataset (Espinilla, Medina & Nugent, 2018)
- Historie des Smart Homes vom Autor dieser Arbeit

Dazu wurden die Datensätze vorweg in Microsoft® Excel und RStudio® betrachtet, um innerhalb dieser den erforderlichen Zieldatensatz zu finden. Bei letzterem Datensatz zeigte sich, dass dieser aufgrund des kleineren Volumens an Daten weniger Aussagekraft geliefert hätte.

Nachfolgend wurden die verbleibenden beiden Datensätze verglichen. Hierbei wurde als Zieldatensatz der zweite gewählt. Dies erfolgte, da bei Machine Learning Algorithmen auch die Hardware für solche Unterfangen ausgelegt sein muss. Der erste Datensatz hat inhaltlich ähnliche Werte geliefert wie der zweite, jedoch war bereits in der ersten Sichtung des ersten Datensatzes das Laden dieses zeitlich herausfordernd. Daher wurde aufgrund der Größe des ersten Datensatzes davon abgesehen, diesen mit der verfügbaren Hardware in einem Machine Learning Algorithmus einzusetzen, da diese die anfallenden Datenmengen schlichtweg nicht bewältigen konnte.

Somit konnte innerhalb dieser Phase der gewünschte Zieldatensatz von UCAMl Cup 2018 dataset (Espinilla, Medina & Nugent, 2018) eruiert werden und wird im Nachfolgenden als Zieldatensatz benannt. Dieser stellt somit die Basis für den zu entwickelnden Machine Learning Algorithmus dar.

Anhand der Zusatzinformationen laut Espinilla, Medina und Nugent (2018) über den gewählten Zieldatensatz zeigt sich, dass es sich hierbei um eine Laborumgebung handelt, welche eine normale Wohnung repräsentieren soll. Diese Wohnung hat eine Größe von circa 25m<sup>2</sup>, mit 5,5m Länge und 4,6m Breite. Diese Wohnung wurde in fünf Regionen eingeteilt, welche aus Eingang, Küche, Arbeitsplatz, Wohnzimmer und Schlafzimmer bestehen.

Der Wohnraum wurde sensorisch ausgestattet, um umfangreiche Daten sammeln zu können. Generell wurde für dieses Unterfangen openHAB zur Datenerfassung eingesetzt. Für das Erfassen der Daten wurden 30 binäre Sensoren (siehe Abbildung 20 und Tabelle 4), 15 Annäherungssensoren (siehe Abbildung 21 und Tabelle 5), intelligenter Boden (siehe Abbildung 22) und Beschleunigungssensoren eingesetzt. Der intelligente Boden bestand gesamt aus 40 Teilelementen in der Wohnung, wobei jedes Element jeweils aus 8 Sensoren bestand. Für die Erfassung der Beschleunigungsdaten und der Annäherungsdaten trug der Proband "Mario" eine Smartwatch.

Im Zuge der vorangehenden Betrachtung des gewählten Datensatzes zeigte sich, dass die Daten bereits in Trainings- und Validierungsdaten geteilt waren. Innerhalb dieser beiden Teile lagen die Daten in einzelnen Dateien verteilt vor. Dabei wurden die Dateien einerseits je Tag erstellt. Die Tage waren zusätzlich in die drei Zeitsegmente Vormittag, Mittag und Nachmittag unterteilt. Da hier fortlaufende Messdaten von mehreren unterschiedlichen Sensoren vorliegen, handelte es sich um multivariate Zeitserien. Andererseits waren die Dateien je Kategorie aufgeteilt. Diese Kategorien setzten sich zusammen aus: Beschleunigung, Aktivität, Bodensensor, Annäherungssensoren und Sensoren (zum Beispiel: Tür auf/zu). Somit konnte festgestellt werden, dass die Daten, wie nachfolgend gelistet, aufgesplittet vorliegen:

- Geteilt in Trainings- und Validierungsdaten
- Je Tag
- Zeitsegment

- Vormittag
- Mittag
- Nachmittag
- Kategorie
  - Beschleunigung
  - Aktivität
  - Bodensensor
  - Annäherungssensoren
  - Sensoren

Innerhalb dieser Dateien variierte die Struktur der vorliegenden Samples. Beispielsweise lässt sich nennen, dass für die Kategorie Beschleunigung ein Sample je Zeile aus dem Zeitstempel und der Beschleunigung auf x-Achse, y-Achse und z-Achse bestand. Als zweites Beispiel kann die Kategorie Bodensensor betrachtet werden, wobei ein Sample je Zeile aus dem Zeitstempel, dem jeweiligen Quellsensor und dessen Messwerten bestand.

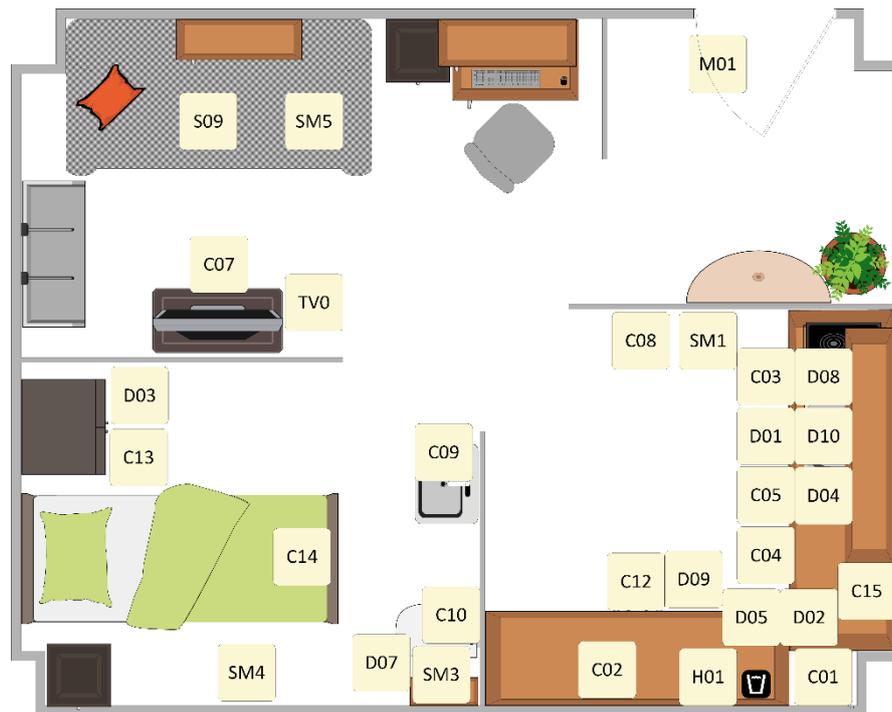


Abbildung 20: Binäre Sensoren im UJAml Smart Lab (in Anlehnung an Martínez-Santiago et al., 2020, S. 230)

Sensor-Codes	Beschreibung
M01	Tür
TV0	Fernseher
SM1	Sensor Küche Bewegung
SM3	Bewegungssensor Bad
SM4	Bewegungssensor Bett
SM5	Bewegungssensor Sofa
D01	Kühlschrank
D02	Mikrowelle
D03	Kleiderschrank
D05	Geschirrspüler
D07	Top WC
D08	Schrank
D09	Waschmaschine
D10	Speisekammer
H01	Wasserkocher

Sensor-Codes	Beschreibung
C01	Medizinbox
C02	Fruchtplatte
C03	Geschirr
C04	Töpfe
C05	Wasserflasche
C07	Remote Xbox
C08	Müll
C09	Wasserhahn
C10	Tank
C12	Wäschekorb
C13	Pyjamalade
C14	Bett
C15	Küchenabfluss
S09	Drucksensor Sofa

Tabelle 4: Codes der binären Sensoren im UJAml Smart Lab

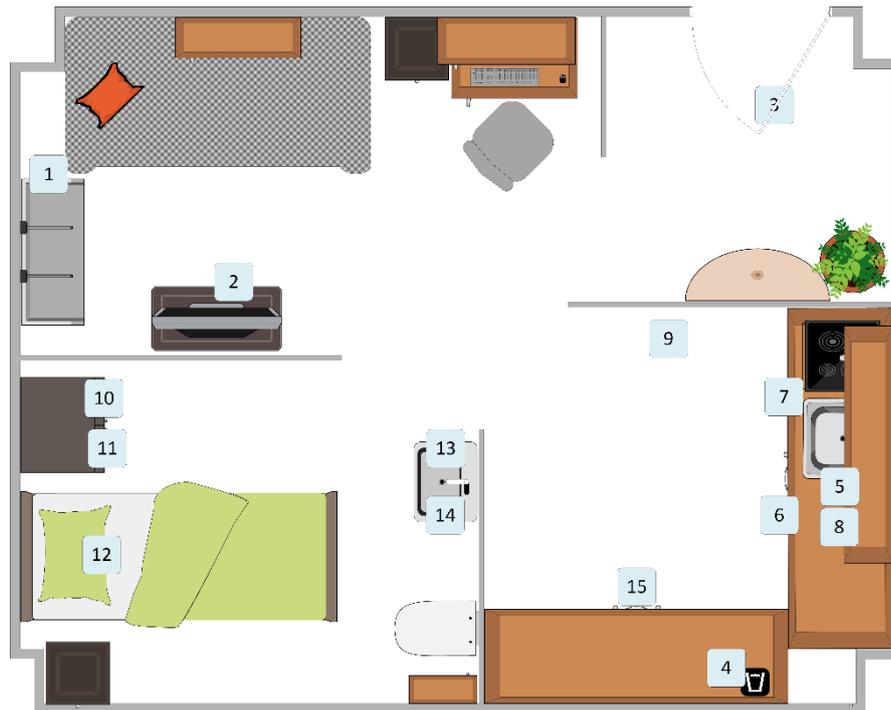


Abbildung 21: Annäherungssensoren im UJAml Smart Lab (in Anlehnung an Espinilla, Martinez, et al., 2018, S. 34640)

Annäherungs-sensor-Codes	Beschreibung
1	Buch
2	Fernbedienung
3	Eingangstür
4	Medizinbox
5	Essensregal
6	Kühlschrank
7	Topfschrank
8	Wasserflasche

Annäherungs-sensor-Codes	Beschreibung
9	Mülleimer
10	Schranktür
11	Pyjamalade
12	Bett
13	Wasserhahn
14	Zahnbürste
15	Wäschekorb

Tabelle 5: Codes der Annäherungssensoren im UJAml Smart Lab

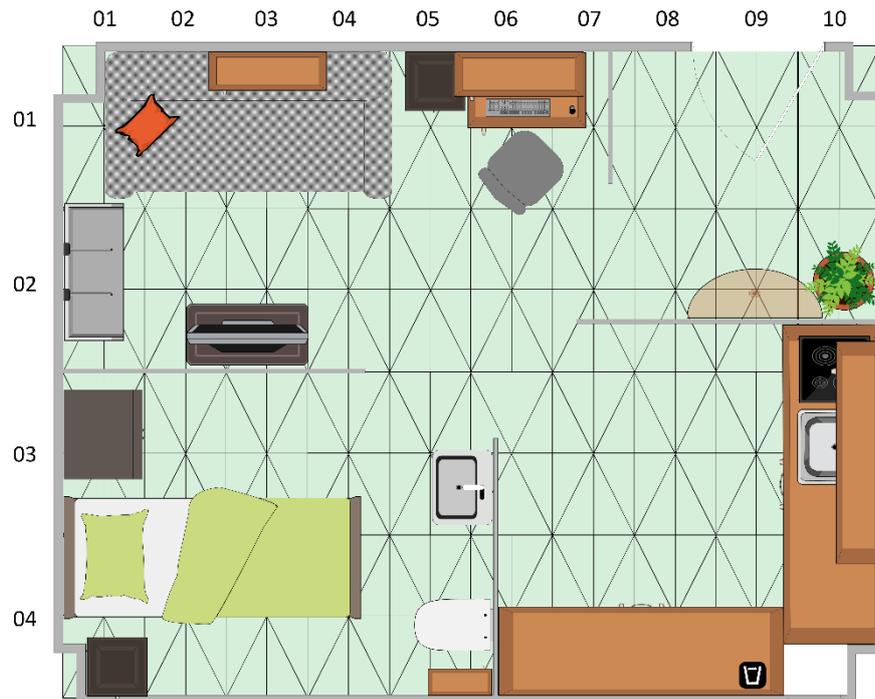


Abbildung 22: Intelligenter Boden im UJAmI Smart Lab (in Anlehnung an Espinilla, Medina & Nugent, 2018, S. 11)

### 3.2.3 Datenvorbereitung

Diese Phase stellt die Vorbereitung der Daten für die Modellierung dar (Chapman et al., 2000). Da Datensätze die Hauptlernquelle für Machine Learning sind, ist dieses Vorbereiten ein wesentlicher Schritt (Gollapudi & Laxmikanth, 2016). Dabei sollen die Daten gegebenenfalls bereinigt, selektiert und transformiert werden (Chapman et al., 2000).

Speziell zu behandeln sind mögliche unvollständige Dateneinträge, oder auch ein Rauschen innerhalb der Daten ist zu überprüfen (Fayyad et al., 1996). Für eine Datenvorbereitung kann das Konstruieren durch Transformieren von Daten, das Beziehen neuer Attribute aus bestehenden Daten und ein mögliches Kombinieren von Daten erforderlich sein (Chapman et al., 2000). Der angestrebte Output dieser Phase ist ein für das Machine Learning vorbereiteter Datensatz (Chollet, 2018). Daher ist ein Verständnis der Datentypen, ob kategorial, ordinal, numerisch oder spezielle Formen, erforderlich (Gollapudi & Laxmikanth, 2016). Besonders bei Zeitdaten kann ein Formatieren wichtig sein (Chapman et al., 2000).

Bei der Datenvorbereitung wird auch spezifisch für Machine Learning darauf geachtet, dass die vorhandenen Daten in zwei Datensätze geteilt werden (siehe Abbildung 23), die als Trainings- und Validierungsdaten benannt werden (Zheng, 2015). Bei den Trainingsdaten handelt es sich um jene, die konkret im Machine Learning Algorithmus verarbeitet werden, um damit die Basis eines Modells zu bilden (Gollapudi & Laxmikanth, 2016). Die Validierungsdaten dienen einem Testing des erstellten Modells (Gollapudi & Laxmikanth, 2016; Zheng, 2015). Dazu wird geprüft, wie stark die Vorhersage anhand des erstellten Modells auf die Validierungsdaten zutrifft (Gollapudi & Laxmikanth, 2016).

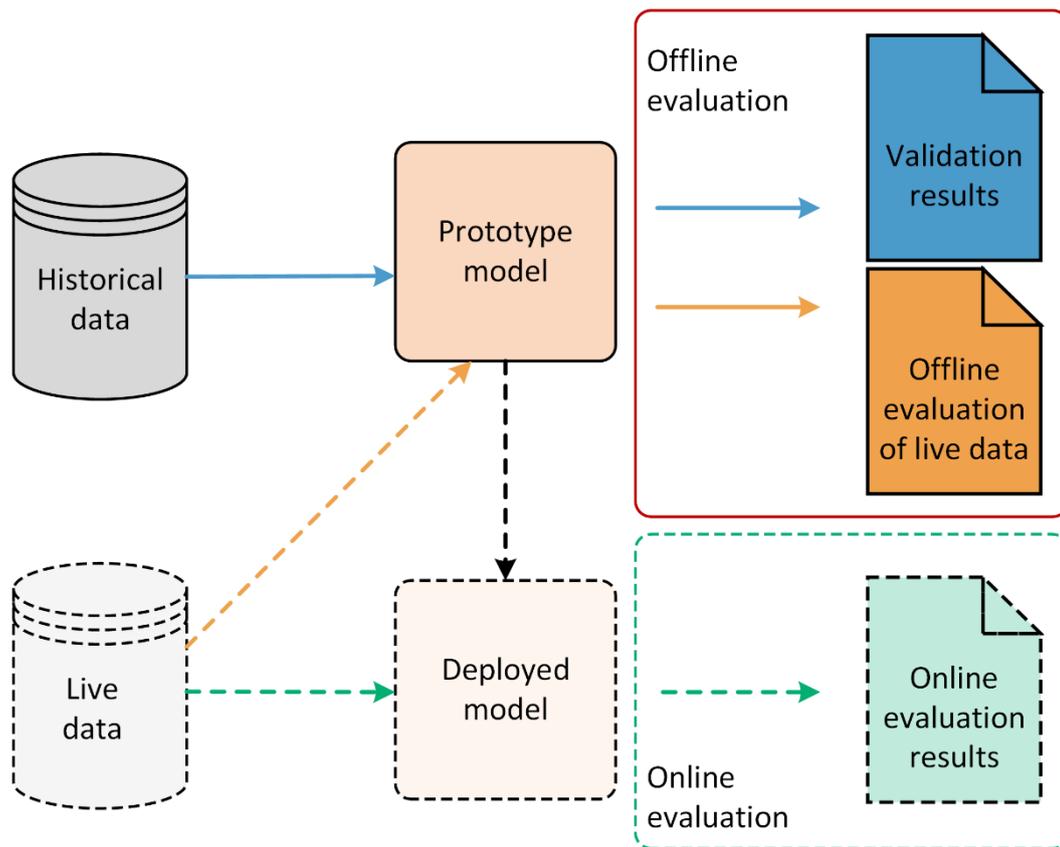


Abbildung 23: Evaluierungsworkflow für die Daten eines Machine Learning Models (in Anlehnung an Zheng, 2015, S. 2)

Bereits beim Datenverständnis (Abschnitt 3.2.2) zeigte sich, dass die Daten vorweg in Trainings- und Validierungsdaten geteilt waren. Innerhalb dieser Aufteilung lagen die Daten in einzelnen Dateien verteilt vor. Daher war es in der Vorbereitung der Daten erforderlich, diese in eine Datei für Trainingsdaten und eine zweite Datei für Validierungsdaten zusammenzufassen. Hierzu wurden die einzelnen Dateien innerhalb der Splittung zwischen Trainings- und Validierungsdaten im Text Editor Microsoft® Visual Studio Code in einem Jupyter Notebook mittels der Programmiersprache python™ und der dafür nötigen Library pandas geladen.

Bereits bei diesem ersten Importschritt wurden die Datensätze nach der Kategorie gruppiert. Somit konnte die Verteilung nach Tag und Zeitsegment bereits beim Importieren in die Kategorien integriert werden. Daher lagen nach diesem Schritt die Datensätze nach den Kategorien Beschleunigung, Aktivität, Bodensensor, Annäherungssensoren und Sensoren gruppiert vor.

Innerhalb der Kategorien wurden die Daten transformiert. Dabei wurde die Datensätze, welche je Messwerte eine eigene Zeile für den jeweiligen Sensor enthalten haben, so pivottiert, dass jeder Sensor als eigene Spalte enthalten war. Mögliche fehlende Daten wurden in diesem Schritt mit Not a Number (NaN) befüllt. Speziell für die Kategorie Annäherungssensoren wurden die Messwerte für die Received Signal Strength Indication (RSSI) für Bluetooth Low Energy (BLE) Sensoren durch Distanzklassen ersetzt. Hierbei wurde nach Bertuletti et al. (2016) vorgegangen (siehe Tabelle 6).

BLE RSSI	Distanz
≥ -49,4 dBm	0,5 m
≥ -61,3 dBm	1 m
≥ -64,9 dBm	2 m
≥ -67,5 dBm	3 m
< -67,5 dBm	5,5 m
NaN	15 m

Tabelle 6: Korrelation BLE RSSI Werte zu Distanz

Auch die Datensätze der Bodensensoren erforderten eine spezielle Transformation. Hierbei wurden die Messwerte durch boolesche Werte ersetzt, indem geprüft wurde, ob sich der Messwert im Vergleich zum, zeitlich gesehen, vorhergehenden Messwert verändert hat. Nachdem diese Transformation der Daten abgeschlossen war, musste für die einzelnen Samples je Kategorie ein einheitliches Zeitintervall erstellt werden. Dazu wurden alle Datensätze innerhalb der jeweiligen Kategorie durch resampling auf ein einheitliches Intervall von fünf Sekunden gebracht.

Speziell für die booleschen Variablen innerhalb der einzelnen Samples wurde geprüft, ob mindestens ein True-Wert vorkam, um dann das Sample gegebenenfalls auf True zu setzen. Bei den quantitativen Messdaten wurden diese pro Intervallsample auf deren Mittelwert gesetzt. Nachdem diese Vorbereitungen abgeschlossen waren, konnte das finale Ziel angestrebt werden, die Datensätze gesamt in eine Datei zu speichern. In diesem Schritt wurden sämtliche verbleibende NaN Werte auf 0 gesetzt. Damit war das abschließende Ziel dieser Phase erreicht, indem dieser gesamte Datenvorbereitungsprozess in einem csv-File für Trainingsdaten und einem zweiten csv-File für Validierungsdaten gespeichert wurden, wobei sich zeigte, dass sämtliche Daten nun numerisch interpretiert werden konnten.

### 3.2.4 Modellierung

In dieser Phase erfolgt die Modellierung, und es wird das passende Modell anhand der Daten und den Requirements gewählt (Chapman et al., 2000). Die passende Wahl eines Machine Learning Algorithmus ist für die Modellierung ein wesentlicher Schritt (Chowdhury et al., 2017). Bereits Tarassenko (1998, zitiert nach Chowdhury et al., 2017, S. 287) nannte folgende Kriterien als Basisentscheidung für einen Machine Learning Algorithmus:

- Es gibt eine logische Verbindung zwischen einer Input- und einer Output-Variable.
- Die konkrete Funktion  $f(x) = y$  dieser Verbindung ist unbekannt, beziehungsweise kann nicht mittels einfacher mathematischer Operatoren errechnet werden.
- Es gibt Daten, um Input und Output Variablen zu trainieren und um den Machine Learning Algorithmus zu testen.

Bei Machine Learning Algorithmen werden häufig für das Problem mehreren Lösungen bereitgestellt und es gilt dann, diesen einen Vertrauensgrad zuzuordnen. Somit ist die

Lösungsrelevanz von der Vertrauensbewertung abhängig. Damit zeigt dieser Data-Science-Ansatz dass Machine Learning Algorithmen menschenähnliche Problemlösungsansätze haben, indem Lösungen in ihrer Relevanz charakterisiert werden (Chowdhury et al., 2017). Aus diesem Ansatz heraus ist eine Wahl von zwei Machine Learning Algorithmen, deren Ergebnisse miteinander verglichen werden können, zielführend.

Neben der Wahl der Machine Learning Algorithmen, ist auch eine konkrete Gestaltung der Modellierung wichtig (Chapman et al., 2000). Denn anhand dieser Algorithmen erfolgt das eigentliche Training, basierend auf dem gewählten Zieldatensatz, der bereits für das Training vorab vorbereitet wurde, mit dem das Modell einen Output erzeugen kann (Gollapudi & Laxmikanth, 2016). Zheng (2015) beschreibt hierfür einen Modellierungsprozess für Machine Learning Algorithmen. Durch den Modellierungsprozess von Zheng (2015) soll ein Prototyp erstellt werden, der ein Proof-Of-Concept ermöglicht nach Wilde und Hess (2006).

Die Trainingsdaten und Validierungsdaten werden hierbei für das Training der Machine Learning Algorithmen genutzt (Gollapudi & Laxmikanth, 2016; Zheng, 2015). Um einen validen Prototypen zu erreichen, ist innerhalb des Prototypentrainings (siehe Abbildung 24) darauf zu achten, dass einerseits ein Hyperparameter Tuning durchzuführen ist und andererseits eine Validierungen des jeweiligen Modells erfolgt (Zheng, 2015).

Die konkreten Hyperparameter sind von der Wahl des jeweiligen Machine Learning Algorithmus abhängig und haben einen direkten Einfluss auf das Ergebnis des Modells (Zheng, 2015). Die Validierung dient als Sicherheit, dass es sich bei dem Modell um einen validen Output handelt (Gollapudi & Laxmikanth, 2016). Für die Validierung im Konkreten können unterschiedliche Vorgehen, abhängig vom Datensatz und dem Modell, genutzt werden (Chollet, 2018; Gollapudi & Laxmikanth, 2016; Zheng, 2015). Beispielhaft kann dazu eine Hold-Out-Validierung, K-Cross-Validierung oder Bootstrap und Jackknife durchgeführt werden (Zheng, 2015). Mit diesem Vorgehen, kann in dieser Phase auf ein möglichst valides Modell abgezielt werden (Chapman et al., 2000; Fayyad et al., 1996).

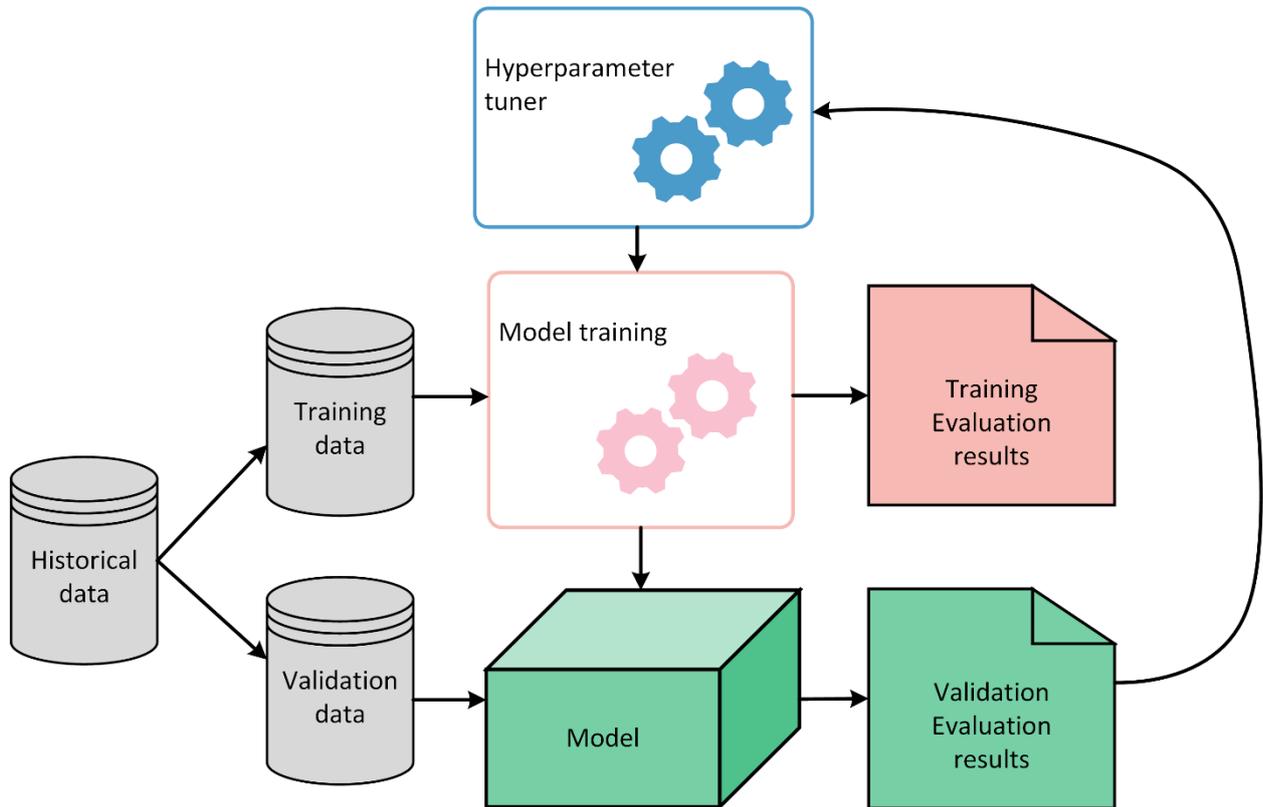


Abbildung 24: Prototypentraining zur Bildung eines Machine Learning Modells (vgl. Zheng, 2015, S. 4)

In dieser Phase wurde, anhand der erstellten Requirements, laut der Use Cases aus Abschnitt 3.2.1 und dem gewählten Datensatz, der passende Machine Learning Algorithmus gewählt. Für eine Vertrauenswürdigkeitskontrolle ist entschieden worden, dass zwei Machine Learning Algorithmen durchgeführt werden sollten, und deren Ergebnisse verglichen werden sollten. Bereits im Datenverständnis (siehe Abschnitt 3.2.2) wurde erkannt, dass es sich bei den Messdaten um Zeitserien handelt. Da es sich bei diesem Datentyp um eine spezielle Form handelt, waren diese ausschlaggebend für die Wahl der Machine Learning Algorithmen. Hierbei zeigte sich, dass in der Domäne des Machine Learnings auf die Algorithmen innerhalb des Deep Learning (siehe Abbildung 25) zurückgegriffen werden muss (Chollet, 2018).

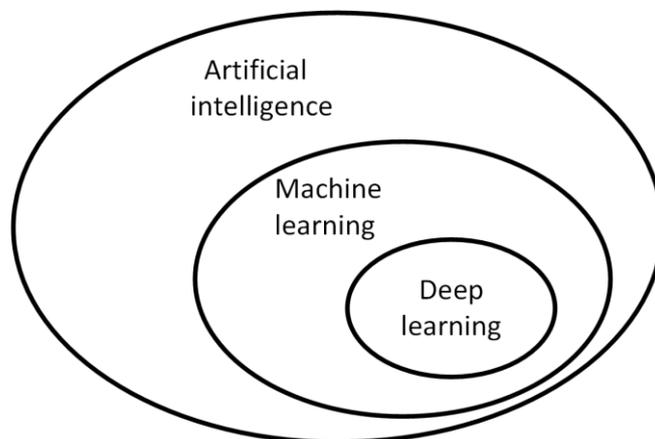


Abbildung 25: Deep Learning innerhalb von ML und AI (vgl. Chollet, 2018, S. 18)

Betreffend der Zeitserien wurde bereits der LSTM als populärster Machine Learning Algorithmus identifiziert (Alahi et al., 2016; Chhetri et al., 2020; Géron, 2018; Hua et al., 2019; Wang et al., 2017; Zou et al., 2020). Bei LSTM handelt es sich um ein Recurrent Neural Network (RNN) (Géron, 2018). Hierbei werden im Algorithmus anstelle von eindimensionalen Vektoren zweidimensionale Matrizen verrechnet. Dies bietet den besonderen Vorteil, dass auch vergangene Daten in der Berechnung berücksichtigt werden können (Géron, 2018). Eine solche Berechnung wird als Forward Layer bezeichnet.

Für einen Vergleich mit einem zweiten Algorithmus wurde hierzu der BLSTM als erweiterte Form des LSTM genannt (Chhetri et al., 2020; Wartala, 2018; Zou et al., 2020). BLSTM wurde um einen Backward Layer erweitert. In diesem befinden sich auch LSTM Zellen, allerdings mit umgekehrter Verarbeitungsrichtung. Dies hat zum Effekt, dass im BLSTM Algorithmus auch zukünftige Daten einen Einfluss auf das Ergebnis haben (Wartala, 2018).

Seitens der Theorie ist für LSTM und BLSTM bei den Hyperparametern auf die Anzahl der Epochen, Größe von Batches und die Anzahl der Neuronen einzugehen (Zou et al., 2020). Eine Epoche stellt einen gesamten Durchlauf eines Datensatzes dar. Nach jedem Durchlauf einer Epoche passt der Algorithmus automatisch seine Variablen an, um  $f(x) = y$  zu optimieren (Géron, 2018). Innerhalb jeder Epoche werden diese Parameter bereits auf Basis der Batch-Größe angepasst.

Die BatchGröße gibt hierbei an, wie viele Inputvektoren des Datensatzes verarbeitet werden müssen, bevor die Variablen angepasst werden (Géron, 2018; Wartala, 2018). Bei der Anzahl der Neuronen wiederum, handelt es sich um den Parameter für die Konfiguration der zu verwendenden Neuronen in dem jeweiligen Layer. Bei BLSTM bezieht sich die Anzahl der Neuronen sowohl auf den Forward Layer als auch auf den Backward Layer (Rashid, 2017; Wartala, 2018).

Basierend auf diesem Wissen wurde für beide Algorithmen für die genannten Hyperparameter eine Gittersuche mit Kreuzvalidierung eingesetzt. Dieses Vorgehen bestand aus 15 Iterationen, je Ausprägung der einzelnen Parameter (Géron, 2018).

In dem Hyperparameter Tuning der beiden Machine Learning Algorithmen ist die Prüfung der Parameter mittels des Root Mean Square Error (RMSE) erfolgt. Der RMSE ist die Standardabweichung der Residuen, wobei die Residuen die Differenz zwischen dem vorhergesagten Wert und dem tatsächlichen Wert sind. Beim RMSE handelt es sich somit um den durchschnittlichen Fehler des Modells.

Die Auswahl des optimalen Wertes für den Hyperparameter erfolgte über den RMSE-Wert mit der geringsten Streuung (Wartala, 2018). Das Modell, das bereits durch das Hyperparameter Tuning optimiert wurde, ist dann mittels der Hold-Out-Validierung überprüft worden. Die Wahl der Hold-Out-Validierung konnte bereits anhand der Angaben von Espinilla, Medina und Nugent (2018) getroffen werden. Dabei wurde schon genannt, dass aufgrund des vorliegenden Datensatzes eine Hold-Out-Validierung zu wählen ist. Anhand dieses Modellprozesse konnte als Ziel ein valides Vorhersagemodell für beide Algorithmen angestrebt werden.

### 3.2.5 Evaluierung

*“How can I measure success for this project?”*

*(Zheng, 2015)*

Diese Phase befasst sich mit der Evaluierung des erstellten und validierten Modells (Chapman et al., 2000). Dazu ist es erforderlich, mögliche statistische Werte eines Modells zu erfassen und zu analysieren (Chapman et al., 2000; Fayyad et al., 1996).

Abschließend ist es das Ziel, die Ergebnisse eines Modells und dessen Daten zu präsentieren (Fayyad et al., 1996). Hierzu können beispielsweise Boxplots eine gute Vergleichsmöglichkeit bieten (Chowdhury et al., 2017). Für die Machine Learning Algorithmen LSTM und BLSTM wird geprüft, in welchem Verhältnis deren Vorhersagen zu den tatsächlichen Werten stehen. Anhand dieser Werte erfolgt ein Vergleich der jeweiligen Machine Learning Algorithmen, basierend auf den Metriken Mean Squared Error (MSE), Pearson Korrelationskoeffizient (PCC), RMSE und R<sup>2</sup> (siehe Tabelle 7). Ziel hierbei ist eine Interpretation zur jeweiligen Vertrauensbewertung der Modelle.

Bezeichnung	Formel
MSE	$\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2$
PCC	$\frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$
RMSE	$\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2}$
R <sup>2</sup>	$1 - \frac{\sum_{i=0}^{n_{samples}-1} (x_i - y_i)^2}{\sum_{i=0}^{n_{samples}-1} (x_i - \bar{y})^2}$

Tabelle 7: Berechnung der eingesetzten Metriken

### 3.3 Informationen zur Hardware, Tools und Software

Zur Vollständigkeit der Informationen des Vorgehens wurden auch die Details über die Hardware, Tools und Software, die für diese Arbeit genutzt wurden, aufgelistet. Besonders für Tools und Software, die zum Einsatz kamen, wurden auch relevante Informationen zu den Versionen ergänzt.

#### Hardware

- Betriebssystem: Microsoft® Windows 10 Pro 20H2 Build 19042
- CPU: Intel® Core™ i7-7700k
- GPU: Nvidia® GeForce® GTX 1080

- RAM: 32 GB

### **Tools und Software**

- python™ Version 3.8.5
- pip Version 20.1.1
- Libraries
  - pandas Version 1.2.3
  - keras Version 2.4.3
  - matplotlib Version 3.3.4
  - h5py Version 2.10.0
  - scipy Version 1.4.1
  - six Version 1.15.0
  - sklearn Version 0.0
  - tensorflow Version 2.2.0
  - absl-py Version 0.10.0
  - tf-slim Version 1.1.0
  - ml-metadata Version 0.26.0
  - keras-tuner Version 1.0.2
  - mock Version 4.0.3
  - numpy Version 1.19.5

## 4 IMPLEMENTIERUNG UND VALIDIERUNG

Im nachfolgenden Kapitel wird einerseits auf die Aufbereitung und Analyse der Daten eingegangen, andererseits auf die Analyse der erstellten Modelle. Hierzu wird auf die Erstellung der Source Code Komponenten für den Import der Daten eingegangen (Phasenerklärung dazu in Abschnitt 3.2.2). Anhand dieser Daten wird die eingehende Bereinigung und Auswertung der Rohdaten erläutert (wie im Abschnitt 3.2.3 theoretisch behandelt). Für diese Rohdaten wird zusätzlich eine statistische Interpretation präsentiert.

Aufbauend auf diese Phasen wird in der Phase zur Modellierung auf Source Code Komponenten, der Erstellung und dem Vergleich der Maschine Learning Modelle eingegangen. Dieses wird zur Evaluierung der Modelle durchgeführt, um zu prüfen, inwiefern Machine Learning zur präemptiven Steuerung in ein Smart Home integriert werden kann.

Für dieses Vorgehen wurde die Implementierung des Source Codes mit Jahresbeginn 2021 begonnen. Der gesamte Source Code wurde im Zeitraum von etwa 2,5 Monaten erstellt. Dazu wurde auch die Validierung modellseitig und die Evaluierung der Ergebnisse geprüft. In den folgenden Abschnitten werden Teile des Source Codes näher beschrieben und die Interpretation dieser Ergebnisse behandelt.

### 4.1 Datenaufbereitung und -analyse

In diesem Abschnitt werden die Ergebnisse zu den Phasen Datenverständnis und -vorbereitung erläutert. Zu diesem Zweck wurden bereits für den Import der Rohdaten für die Trainings- und Validierungsdaten zwei separate Jupyter Notebooks erstellt. Zwar ist ein großer Teil des Source Codes für beide Datensätze ident, jedoch war es eine zielorientierte Entscheidung, diese in zwei dedizierten Dateien zu implementieren. Diese Entscheidung wurde getroffen, da hierdurch die Interpretation der Daten erleichtert werden sollte. Denn es wurden Graphen direkt im Source Code erstellt. Hierbei wurde auf den positiven Aspekt von Jupyter Notebooks zurückgegriffen, dass dies inline angezeigt werden konnte.

Aufgrund dieser Aufteilung in zwei dedizierte Dateien herrscht im gesamten Vorgehen ein Bewusstsein für den jeweils bearbeiteten Datensatz. Der Source Code je Datei umfasst jeweils in etwa 300 Zeilen Source Code. Dabei ist zu nennen, dass der Source Code zu gleichen Teilen in Funktionen betreffend dem Datenverständnis, zur statistischen Interpretation der Daten, und den Funktionen zur Datenvorbereitung, zum Import und der Aufbereitung der Daten, aufgeteilt werden kann.

Die statistische und graphische Auswertung der Datensätze zeigte, dass sich Trainings- und Validierungsdaten sehr ähnlich sind. Aufgefallen ist hierbei, dass fehlerhafte oder fehlende Daten ausschließlich im Validierungsdatensatz vorhanden waren. Hier können zum Beispiel die Sensoren C02 und D09 genannt werden. Diese waren im Trainingsdatensatz enthalten, im Validierungsdatensatz fehlten diese jedoch vollständig. Weiters fehlten bei einigen Datensätzen der Annäherungssensoren die ID und die Objektbezeichnung. Bei den Annäherungssensoren ist

der des Bettes hervorzuheben. Für diesen gab es, im Verhältnis zu den anderen Sensoren, sehr viele RSSI Daten größer als  $-67,5$  dBm. Hierdurch wurden viele der vorhandenen Daten einer Distanz von  $5,5$ m zugeordnet (siehe Abbildung 26 und Abbildung 27). Die Daten der verbleibenden Annäherungssensoren verteilten sich in etwa immer gleich. Es gab sehr viele Daten kleiner als  $-67,5$  dBm, und nur sehr wenige in den anderen Distanz-Klassifizierungen. Beispielfhaft werden hierfür in Abbildung 28, Abbildung 29, Abbildung 30 und Abbildung 31 die Daten für die Sensoren Schranktür und Kühlschrank dargestellt.

Die Analyse der Bodensensoren erfolgte als Heatmap. Hierbei wurde die Anzahl der Aktivierungen je Sensorelement als prozentualer Anteil der gesamten Daten der Bodensensoren des jeweiligen Datensatzes dargestellt. Markant waren hier die Daten des Sensorelements 01,07. Dieses eine Element verursachte in beiden Fällen nahezu 20% der Aktivierungen (siehe Abbildung 32 und Abbildung 33). Im Vergleich mit Abbildung 22 zeigt sich, dass es sich hierbei um das Sensorelement beim Eingang der Laborumgebung handelte.

Die Beschleunigungssensoren erfassten in beiden Datensätzen Werte von  $20$  und  $-20$   $m/s^2$ . Bei der Prüfung der Boxplot-Diagramme fiel auf, dass die Z-Achse lediglich Ausreißer mit negativer Beschleunigung aufweist, während die X- und Y-Achse auch Ausreißer im positiven Bereich aufweist (siehe Abbildung 34 und Abbildung 35). Daher wurde die Verteilung der Daten geprüft. In Abbildung 36 und Abbildung 37 ist zu erkennen, dass die Daten primär im positiven Bereich angesiedelt sind. Die konkrete physische Struktur der Smartwatch wurde im Zuge dieser Arbeit nicht näher überprüft, daher wurde die Verteilung in dieser Form zur Kenntnis genommen und nicht weiter transformiert.

In der Aufbereitung der Daten wurden die erkannten Probleme des Datensatzes bereinigt und für die Verarbeitung durch das Machine Learning Modell gespeichert. Als Ursache dafür, dass primär der Validierungsdatensatz fehlerbehaftet war, wurde die Tatsache festgestellt, dass der Datensatz im Rahmen einer Konferenz verwendet wurde, damit Wissenschaftler/innen und Experten/innen auf dem Gebiet des Machine Learning betreffend Aktivitätsklassifizierung ihre Tools, Algorithmen und Wissen erproben und miteinander vergleichen konnten (Espinilla, Medina & Nugent, 2018). Aus den aufbereiteten Daten ergaben sich 85 Features. Der Trainingsdatensatz besteht aus 6326 Inputvektoren, der Validierungsdatensatz besteht aus 3087 Vektoren.

Annäherungssensor - Bett

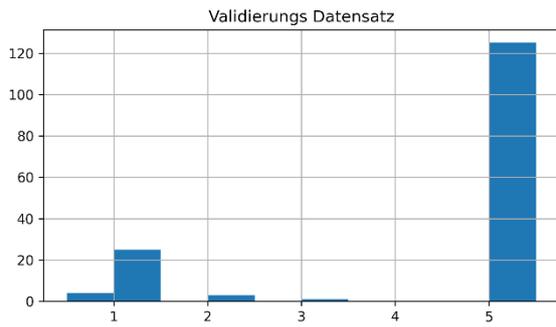


Abbildung 26: Distanzklassen für Annäherungssensor Bett im Validierungs Datensatz

Annäherungssensor - Bett

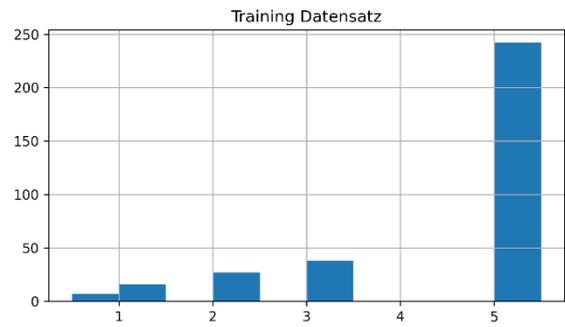


Abbildung 27: Distanzklassen für Annäherungssensor Bett im Training Datensatz

Annäherungssensor - Schranktür

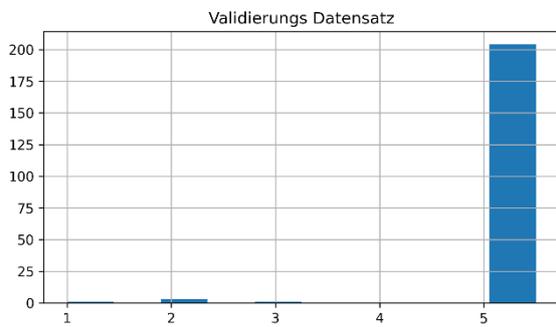


Abbildung 28: Distanzklassen für Annäherungssensor Schranktür im Validierungs Datensatz

Annäherungssensor - Schranktür

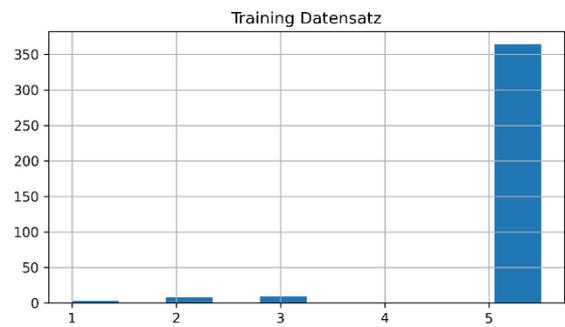


Abbildung 29: Distanzklassen für Annäherungssensor Schranktür im Training Datensatz

Annäherungssensor - Kühlschrank

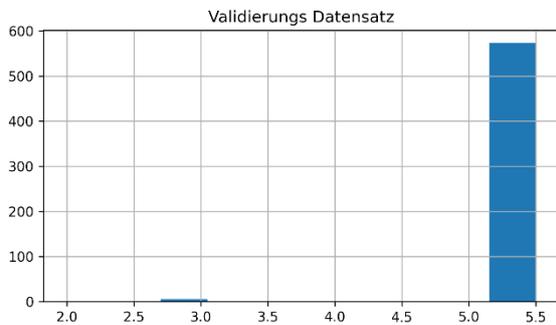


Abbildung 30: Distanzklassen für Annäherungssensor Kühlschrank im Validierungs Datensatz

Annäherungssensor - Kühlschrank

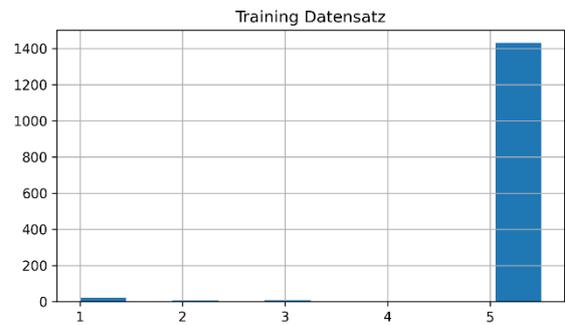


Abbildung 31: Distanzklassen für Annäherungssensor Kühlschrank im Training Datensatz

Aktivierungen der Bodensensoren

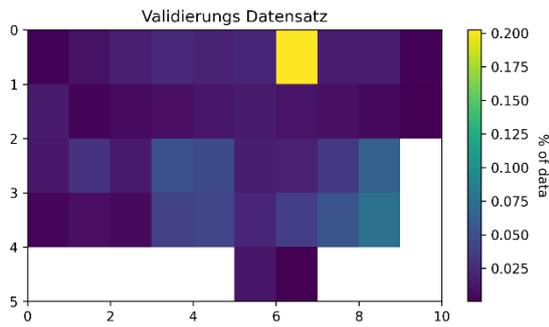


Abbildung 32: Heatmap Aktivierungen der Bodensensoren im Validierungs Datensatz

Aktivierungen der Bodensensoren



Abbildung 33: Heatmap Aktivierungen der Bodensensoren im Training Datensatz

Beschleunigung

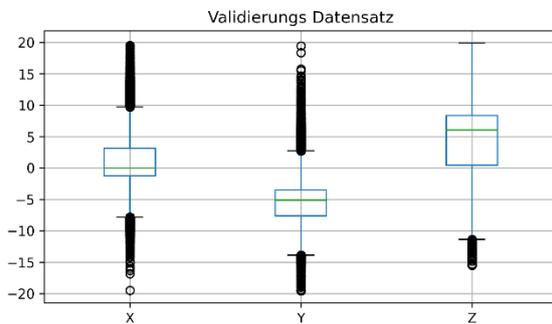


Abbildung 34: Boxplot der Beschleunigungswerte im Validierungs Datensatz

Beschleunigung

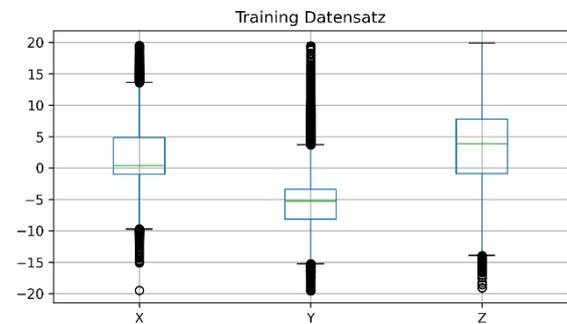


Abbildung 35: Boxplot der Beschleunigungswerte im Training Datensatz

Z-Beschleunigung Verteilung

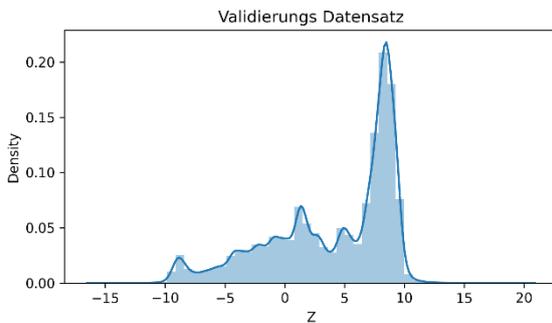


Abbildung 36: Verteilung der Z-Achsen Beschleunigungswerte im Validierungs Datensatz

Z-Beschleunigung Verteilung

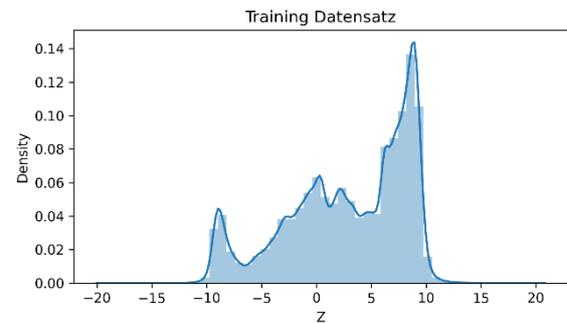


Abbildung 37: Verteilung der Z-Achsen Beschleunigungswerte im Training Datensatz

## 4.2 Modellanalyse

Bei diesem Abschnitt werden die Ergebnisse aus dem Vorgehen der Phasen Modellierung und Evaluierung erläutert. Ebenfalls im Zuge des eigentlichen Trainings der Machine Learning Modelle wurden mehrere Jupyter Notebooks implementiert. Dabei wurde auf eine klare Abgrenzung im Source Code zwischen den überschneidenden Funktionen und den, je gewähltem

Modell, abweichenden, LSTM und BLSTM, geachtet. Dies resultiert auch aus der Tatsache, dass die jeweilige Implementierung zwar in der Basis die gleichen Funktionen teilt, jedoch aufgrund des jeweiligen Algorithmus des Modells im Verlauf des Source Codes abweicht. Beispielfähig können hier Funktionen für die Erstellung der Vorhersagevektoren und die Berechnung des PCC anhand der Realdaten und den Vorhersagevektoren genannt werden.

#### 4.2.1 Hyperparameter Tuning

Das Hyperparameter Tuning wurde, wie bereits in Abschnitt 3.2.4 erwähnt, als Kreuzvalidierung, speziell dem einmaligen Ablauf, durchgeführt. Für jeden Parameter wurden mehrere Ausprägungen festgelegt und anschließend je 15 Modelle erstellt und validiert. Die verwendeten Parameter und Wiederholungen basieren auf Erfahrungswerten des Autors.

Um den Vergleich zwischen den einzelnen Parametersets zu erleichtern, wurden diese graphisch als Boxplot Diagramme und die maximal und minimal Werte als Tabelle dargestellt. Bei der Darstellung als Boxplot wurden vorhandene Ausreißer ausgeschlossen, um die Lesbarkeit der Grafik zu erleichtern. Während des Epochen Tunings wurde für beide Algorithmen die Batch-Größe mit 4 und die Anzahl der Neuronen mit 1 festgelegt. Die nachfolgenden Werte wurden in die bereits zuvor ermittelten Parameter eingesetzt.

Bei LSTM zeigt sich, dass eine Epoche mit einem RMSE von 0,061 bis 0,133 zur geringsten Abweichung zwischen den tatsächlichen Werten und den vorhergesagten Werten führt. Die Streuung ist jedoch relative hoch. 20 Epochen wiederum führen zu einem RMSE zwischen 0,226 und 0,268 (siehe Abbildung 38 und Tabelle 8). Die Wiederholbarkeit ist hier wesentlich höher, daher wurde mit 20 Epochen weiter getunt. Die Ergebnisse für die Batch-Größen sind eindeutiger. Eine Batch-Größe von 8 führt mit einer Differenz von 0,074 zwischen den höchsten und niedrigsten berücksichtigten RMSE zum besten Ergebnis (siehe Abbildung 39 und Tabelle 9). Abschließend wurde, mit den zuvor genannten Parametern, die Anzahl der Neuronen für das LSTM Model geprüft. Eine Anzahl von 10 Neuronen führte zwar zu einer Differenz der aufgezeichneten RMSE von 0,089, jedoch wurde aufgrund der sehr geringen Abweichung und der lediglich geringfügig schlechteren Wiederholbarkeit die Neuronen die Anzahl auf 7 festgelegt (siehe Abbildung 40 und Tabelle 10).

Die Ergebnisse des Tunings des Epochen Parameters für den BLSTM Algorithmus zeigen 50 Epochen als den optimalen Wert. Dieser führte lediglich zu einer maximalen Differenz von 0,034 zwischen den RMSE (siehe Abbildung 41 und Tabelle 11). Die Batch-Größe wurde basierend auf den Ergebnissen des Tunings auf 8 festgelegt. Die anderen Ausprägungen wiesen hier mindestens die fünffache Differenz auf (siehe Abbildung 42 und Tabelle 12). Die Ermittlung der optimalen Anzahl der Neuronen zeigt, dass 7 Neuronen außerordentlich gut abschnitten. Die Wiederholbarkeit lag hier bei einer maximalen Abweichung von 0,020. Die nächstbeste Ausprägung von 1 Neuron wies mit einer Differenz von 0,056 eine wesentlich höhere Ungenauigkeit auf (siehe Abbildung 43 und Tabelle 13).

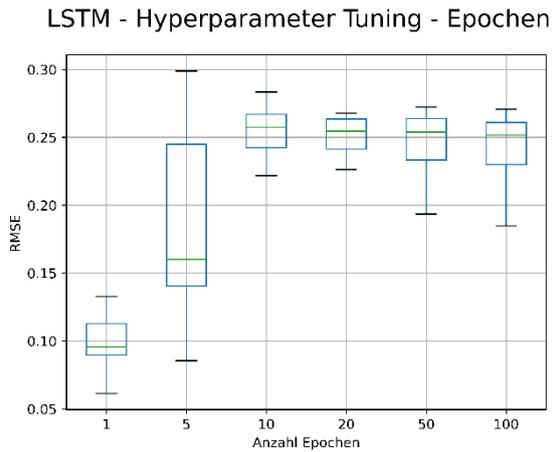


Abbildung 38: Ergebnis des LSTM Epochen Tunings

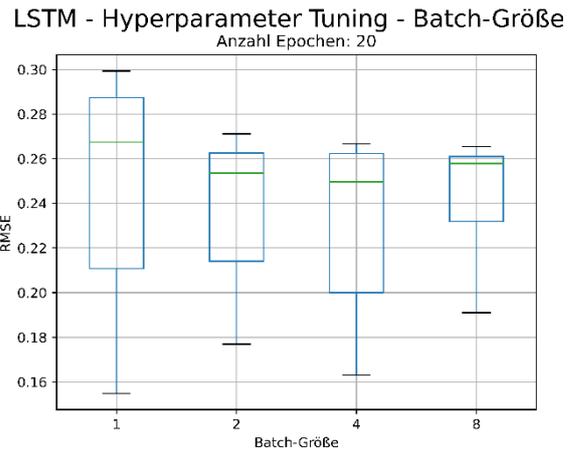


Abbildung 39: Ergebnis des LSTM Batch-Größe Tunings

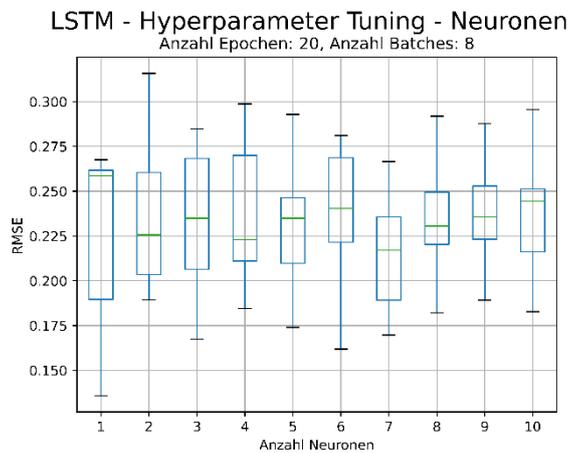


Abbildung 40: Ergebnis des LSTM Neuronen Tunings

	1	5	10	20	50	100
<b>min. RMSE</b>	0,061	0,086	0,222	0,226	0,194	0,203
<b>max. RMSE</b>	0,133	0,299	0,283	0,268	0,272	0,271
<b>Differenz</b>	<b>0,071</b>	<b>0,213</b>	<b>0,061</b>	<b>0,041</b>	<b>0,079</b>	<b>0,068</b>

Tabelle 8: Minimaler und maximaler RMSE für LSTM Epochen Hyperparameter Tuning

	1	2	4	8
<b>min. RMSE</b>	0,155	0,177	0,163	0,191
<b>max. RMSE</b>	0,299	0,271	0,267	0,265
<b>Differenz</b>	<b>0,145</b>	<b>0,094</b>	<b>0,104</b>	<b>0,074</b>

Tabelle 9: Minimaler und maximaler RMSE für LSTM Batch-Größe Hyperparameter Tuning

	1	2	3	4	5	6	7	8	9	10
<b>min. RMSE</b>	0,136	0,189	0,167	0,184	0,174	0,162	0,170	0,182	0,189	0,183
<b>max. RMSE</b>	0,268	0,316	0,285	0,299	0,302	0,281	0,267	0,292	0,288	0,272
<b>Differenz</b>	<b>0,132</b>	<b>0,127</b>	<b>0,117</b>	<b>0,114</b>	<b>0,128</b>	<b>0,119</b>	<b>0,097</b>	<b>0,110</b>	<b>0,099</b>	<b>0,089</b>

Tabelle 10: Minimaler und maximaler RMSE für LSTM Neuronen Anzahl Hyperparameter Tuning

BLSTM - Hyperparameter Tuning - Epochen

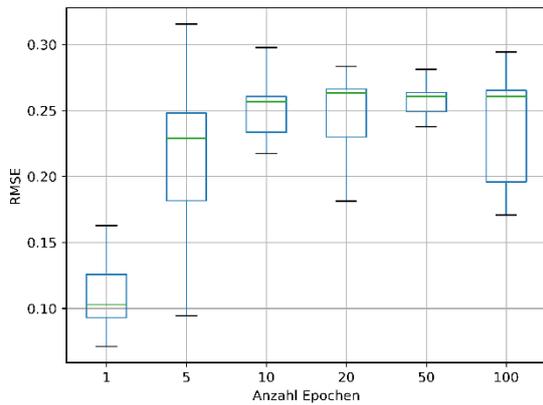


Abbildung 41: Ergebnis des BLSTM Epochen Tunings

BLSTM - Hyperparameter Tuning - Batch-Größe  
Anzahl Epochen: 50

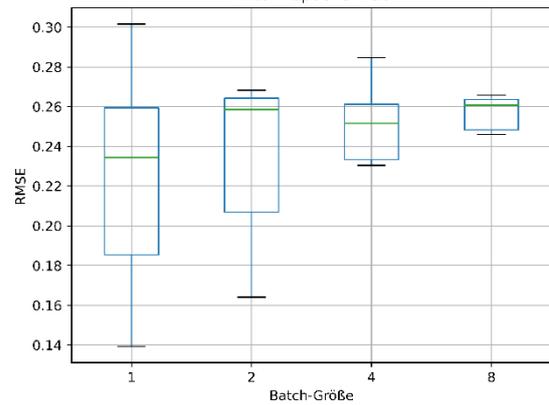


Abbildung 42: Ergebnis des BLSTM Batch-Größe Tunings

BLSTM - Hyperparameter Tuning - Neuronen  
Anzahl Epochen: 50, Anzahl Batches: 8

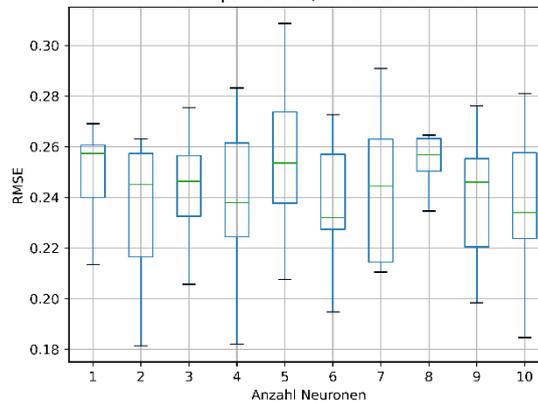


Abbildung 43: Ergebnis des BLSTM Neuronen Tunings

	1	5	10	20	50	100
<b>min. RMSE</b>	0,071	0,095	0,217	0,181	0,238	0,171
<b>max. RMSE</b>	0,163	0,316	0,298	0,284	0,272	0,294
<b>Differenz</b>	<b>0,092</b>	<b>0,221</b>	<b>0,080</b>	<b>0,102</b>	<b>0,034</b>	<b>0,123</b>

Tabelle 11 Minimaler und maximaler RMSE für BLSTM Epochen Hyperparameter Tuning

	1	2	4	8
<b>min. RMSE</b>	0,139	0,164	0,188	0,246
<b>max. RMSE</b>	0,302	0,268	0,285	0,266
<b>Differenz</b>	<b>0,162</b>	<b>0,104</b>	<b>0,097</b>	<b>0,020</b>

Tabelle 12 Minimaler und maximaler RMSE für BLSTM Batch-Größe Hyperparameter Tuning

	1	2	3	4	5	6	7	8	9	10
<b>min. RMSE</b>	0,214	0,181	0,190	0,182	0,208	0,195	0,211	0,244	0,198	0,185
<b>max. RMSE</b>	0,269	0,263	0,275	0,283	0,309	0,273	0,291	0,265	0,276	0,281
<b>Differenz</b>	<b>0,056</b>	<b>0,082</b>	<b>0,085</b>	<b>0,101</b>	<b>0,101</b>	<b>0,078</b>	<b>0,080</b>	<b>0,020</b>	<b>0,078</b>	<b>0,096</b>

Tabelle 13 Minimaler und maximaler RMSE für BLSTM Neuronen Anzahl Hyperparameter Tuning

Somit konnten die erforderlichen Werte je Machine Learning Modell ermittelt werden, um das eigentliche Training des jeweiligen Modells durchführen zu können. Diese Werte werden beim Training an den jeweiligen Algorithmus übergeben. Dadurch sollte ermöglicht werden, dass die Modelle beim Training möglichst optimale Ergebnisse liefern. Optimal bedeutet in diesem Fall ein hohes Zusammenhangsmaß zwischen tatsächlichen und vorhergesagten Werten, und geringe Fehlerquoten. Die ausgewählten Werte je Parameter werden in Tabelle 14 zusammengefasst.

Parameter	LSTM	BLSTM
Epochen	20	50
Batch-Größe	8	8
Neuronen	7	8

Tabelle 14: Ergebnisse des Hyperparameter Tunings

## 4.2.2 Evaluierung der Modelle

Die Modelle wurden, wie in Listing 1 und Listing 2 dargestellt, implementiert. Die Hyperparameter wurden hierbei als Variablen umgesetzt, damit diese im Bedarfsfall einfach geändert werden können.

```

model = Sequential()
model.add(LSTM(neurons, activation='relu', input_shape=(
    trainX.shape[1], trainX.shape[2]), return_sequences=False))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse', metrics=['acc'])
history = model.fit(trainX, trainY, epochs=epochs, batch_size=batch_size,
                    validation_split=0.1, verbose=1, shuffle=False)

```

Listing 1: Implementierung des LSTM Modells

```

model = Sequential()
model.add(Bidirectional(LSTM(neurons, activation='relu', input_shape=(
    trainX.shape[1], trainX.shape[2]), return_sequences=False)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse', metrics=['acc'])
history = model.fit(trainX, trainY, epochs=epochs, batch_size=batch_size,
                    validation_split=0.1, verbose=1, shuffle=False)

```

Listing 2: Implementierung des BLSTM Modells

Nach dem Trainieren der Modelle wurde für diese eine Gegenüberstellung des sogenannten Validation Loss und des Training Loss als Liniendiagramm je Algorithmus erstellt, um zu prüfen, ob ein Over- oder Underfitting vorliegt (siehe Abbildung 44 und Abbildung 45). Da anhand der gesamten Liniendiagramme keine eindeutige Aussage getroffen werden konnte, wurde die

Skalierung der Y-Achse so angepasst, dass die Ergebnisse eindeutig interpretiert werden konnten (siehe Abbildung 46 und Abbildung 47).

Die Daten des BLSTM-Algorithmus zeigen, dass weder Over- noch Underfitting vorliegt. Die Daten für LSTM legen dies auch nahe, jedoch gibt es hierbei einen Ausreißer. Hierdurch wurde die Auswahl der Hyperparameter und die erstellten Modelle als repräsentativ bestätigt. Diese Prüfung war entscheidend, um weitere Evaluierungen der Modelle zur Kontrolle der Arbeitshypothese durchführen zu können. Im Falle, dass diese Ergebnisse keine Repräsentativität bestätigt hätten, wäre ein weiterer Durchlauf des Hyperparameter Tuning erforderlich gewesen. Wenn sich in weiterer Folge hierdurch keine Möglichkeit gezeigt hätte, wäre dies somit die Bestätigung der Arbeitshypothese  $H_0$  gewesen.

Da jedoch dieser Schritt die erstellten Modelle in ihrer Repräsentativität bestätigt, konnte mit den folgenden Analysen fortgefahren werden. Wie in Abschnitt 3.2.5 erläutert wurden die folgenden Metriken für den Vergleich der Algorithmen herangezogen:

- MSE
- PCC
- RMSE
- $R^2$

Beide Algorithmen konnten die Problemstellung außerordentlich gut bewältigen. LSTM erreichte sehr geringe MSE Werte von maximal 0,014 und minimal 0,002, die Standardabweichung lag hierbei bei 0,003. Mit einem BLSTM Modell sinken diese Werte wesentlich. Das BLSTM Modell wies hier einen maximalen Fehler von 0,004 und einer Standardabweichung von 0,002 auf (siehe Abbildung 48).

Der Pearson Korrelationskoeffizient zeigt bei beiden Modellen eine starke lineare Abhängigkeit der vorhergesagten Daten von den tatsächlich erfassten Werten. Die Mittelwerte der beiden lagen hier sehr eng beieinander. Das BLSTM Modell erreichte einen Mittelwert von 0,989, während das LSTM Modell 0,982 erreichte. Die Standardabweichung des LSTM Modells war mit 0,011 im Vergleich zu 0,003 des BLSTM Modells hoch. Absolut betrachtet, ist der Wert des LSTM Algorithmus jedoch noch immer sehr gut (siehe Abbildung 49).

Wie die sehr kleinen MSE Ergebnisse bereits nahelegen, sind die RMSE Werte etwas höher als diese. Auch hier zeigt sich eine deutliche Überlegenheit des BLSTM Modells. Mit einem maximalen Fehler von 0,066 war dieser geringer als der Mittelwert des LSTM Algorithmus mit 0,073 bei dieser Metrik. Der LSTM Algorithmus erzielte hier einen maximalen Fehler von 0,117 (siehe Abbildung 50).

Auch die Daten des  $R^2$  zeigen, dass die erstellten Modelle sehr gut zu den Daten passen. Beide weisen im Mittel ein Bestimmtheitsmaß von 95,5% oder mehr auf. Die Schwankung ist bei den LSTM-Modellen mit einer Standardabweichung von 3,2% jedoch mehr als viermal so hoch als die 0,7% der BLSTM-Modelle (siehe Abbildung 51).

BLSTM learning curve

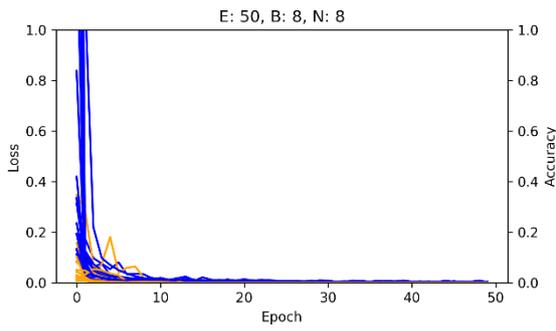


Abbildung 44: BLSTM Lernkurve

LSTM learning curve

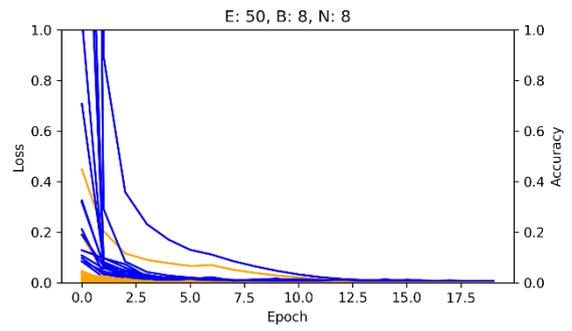


Abbildung 45: LSTM Lernkurve

BLSTM learning curve

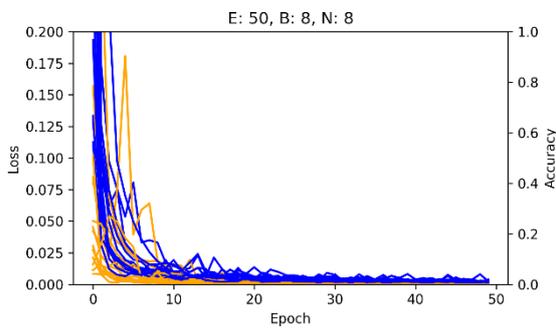


Abbildung 46: Ausschnitt aus BLSTM Lernkurve

LSTM learning curve

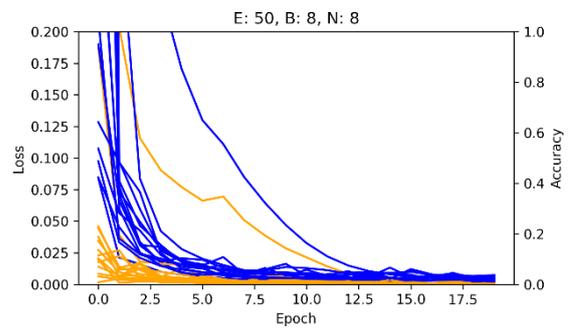


Abbildung 47: Ausschnitt aus LSTM Lernkurve

MSE

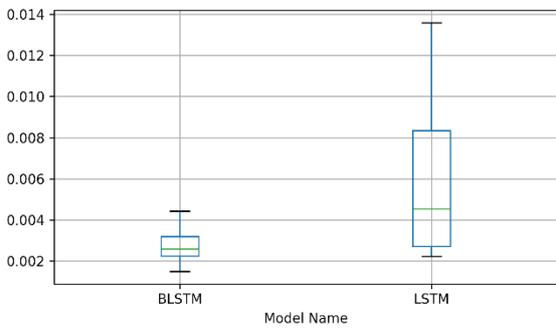


Abbildung 48: MSE der geprüften Modelle

PCC

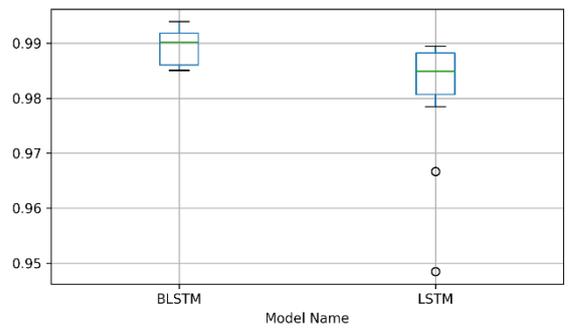


Abbildung 49: PCC der geprüften Modelle

RMSE

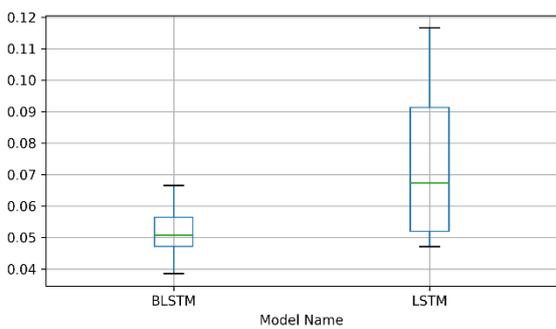


Abbildung 50: RMSE der geprüften Modelle

R<sup>2</sup>

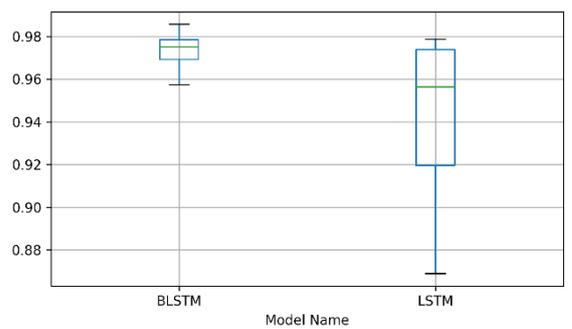


Abbildung 51: R<sup>2</sup> der geprüften Modelle

Für alle erhobenen Metriken hat sich gezeigt, dass der BLSTM Algorithmus bei dieser Problemstellung dem LSTM Algorithmus teils deutlich überlegen ist. MSE, RMSE und  $R^2$  weisen eine wesentlich geringere Streuung auf, und sind außerdem oftmals quantitativ besser. Beim Trainieren der Modelle zeigte sich jedoch, dass das BLSTM Modell bezüglich der Ressourcen anspruchsvoller ist. Wie bereits durch Chowdhury et al. (2017) genannt, ist auch Zeit eine relevante Ressource. Aus diesem Grund gilt es zu erwähnen, dass eben das LSTM, gegenüber dem BLSTM, zeiteffizienter im Training war (siehe Tabelle 15 und Tabelle 16).

	<b>Laufzeit</b>	<b>MSE</b>	<b><math>R^2</math></b>	<b>RMSE</b>	<b>PCC</b>
<b>Mittelwert</b>	69,53 s	0,006	94,5 %	0,073	0,982
<b>Standardabweichung</b>	1,36 s	0,003	3,2 %	0,022	0,011
<b>Minimum</b>	66,98 s	0,002	86,9 %	0,047	0,948
<b>25%</b>	68,73 s	0,003	92,0 %	0,052	0,981
<b>50%</b>	69,75 s	0,005	95,6 %	0,067	0,985
<b>75%</b>	70,02 s	0,008	97,4 %	0,091	0,988
<b>Maximum</b>	73,01 s	0,014	97,9 %	0,117	0,989

Tabelle 15: Metriken des LSTM Models

	<b>Laufzeit</b>	<b>MSE</b>	<b><math>R^2</math></b>	<b>RMSE</b>	<b>PCC</b>
<b>Mittelwert</b>	218,92 s	0,003	97,4 %	0,052	0,989
<b>Standardabweichung</b>	4,59 s	0,001	0,7 %	0,007	0,003
<b>Minimum</b>	213,82 s	0,001	95,7 %	0,039	0,985
<b>25%</b>	216,01 s	0,002	96,9 %	0,047	0,986
<b>50%</b>	217,72 s	0,003	97,5 %	0,051	0,990
<b>75%</b>	221,29 s	0,003	97,8 %	0,056	0,992
<b>Maximum</b>	231,14 s	0,004	98,6 %	0,066	0,994

Tabelle 16: Metriken des BLSTM Models

## 5 CONCLUSIO

Wie in Abschnitt 2.5 aufgezeigt, weisen Smart Services und Smart Homes wesentliche Überschneidungen auf. Der Begriff Smart Homes wird häufig bei Gedanken an Heimautomatisierung verwendet, wobei es sich um eine reine proaktive oder reaktive Handlungsform handelt. Das Konzept der proaktiven und reaktiven Handlung ist auch Teil der Smart Spaces, welche als Ursprung von Smart Services gelten (Acatech - Deutsche Akademie für Technikwissenschaften, 2015).

Smart Services haben sich jedoch im Vergleich zum Smart Home schneller entwickelt. Daher wurden bei diesen proaktive und reaktive Handlungsformen bereits durch präemptive Mechanismen abgelöst (Acatech - Deutsche Akademie für Technikwissenschaften, 2015; Allmendinger & Lombreglia, 2005). Ein zentraler Punkt hierfür ist die allumfassende Datenerfassung. Die Fortschritte im Machine Learning Bereich ermöglichten es Unternehmen, auf Basis der erfassten Daten, ihren Kund/innen präemptive Dienstleistungen anzubieten.

Risteska Stojkoska und Trivodaliev (2017) sagen, dass in einem Smart Home die erfassten Daten zum selbstständigen Lernen genutzt werden sollen. Auch wird in der Forschung festgelegt, dass ein Smart Home sich autonom in hochentwickelter Form an dessen Bewohner/innen anpasst (Mennicken et al., 2014). Hierdurch wurde die folgende Forschungsfrage erkannt:

„Wie und in welchem Umfang kann Machine Learning zur präemptiven Steuerung in ein Smart Home integriert werden?“

Um das Vorgehen in dieser Arbeit besser festlegen zu können wurden außerdem nachfolgende Arbeitshypothesen aufgestellt.

H<sub>0</sub>: Eine präemptive Smart Home Steuerung kann nicht mittels Machine Learning umgesetzt werden.

H<sub>1</sub>: Eine präemptive Smart Home Steuerung kann mittels Machine Learning umgesetzt werden.

In der Festlegung der Vorgehensweise wurde dazu die empirische Softwareentwicklung betrachtet, wobei als Vorgehensweise Prototyping identifiziert wurde. Hierbei wurde die Umsetzung der Modelle auf ein reines Proof-of-Concept eingeschränkt.

Eine applikative Anwendung, im Sinne der Live-Evaluierung der Modelle, fand daher nicht statt. Im weiteren Verlauf wurde das CRISP-DM Konzept um weitere Elemente von anderen Umsetzungsmodellen für Machine Learning Ansätze erweitert, und ein passender Datensatz zum Trainieren und Validieren des angestrebten Machine Learning Models ermittelt. Bei der Ermittlung des Datensatzes wurde eine erste Evaluierung der Datensätze durchgeführt. Hierbei hat sich gezeigt, dass die mögliche Größe des Datensatzes sehr stark durch die verfügbare Hardware eingeschränkt wird.

In der empirischen Umsetzung wurden die Daten des zuvor identifizierten Datensatzes im Detail evaluiert, bereinigt und transformiert. Dies dient dem Zweck der Aufbereitung des Datensatzes für die Verarbeitung durch den Machine Learning Algorithmus. Anschließend wurden erste

Ansätze der eruierten Algorithmen implementiert und die idealen Parameter für diese über Hyperparameter Tuning erarbeitet. Zur Ermittlung der Ergebnisse wurden mehrere Machine Learning Modelle erstellt und damit Vorhersagen auf Basis des Datensatzes erstellt. Mit diesen Vorhersagen wurden statistische Merkmale dieser errechnet, welche den Vergleich der Machine Learning Algorithmen ermöglichte.

Aufgrund dieser Vergleiche zeigte sich, dass die Modelle auf Basis des BLSTM Algorithmus jenen, welche mit dem LSTM Algorithmus erstellt wurden, zumeist wesentlich überlegen sind. Diese Qualität in den Vorhersagen wird jedoch auf Kosten der zu investierenden Zeit beim Training der Modelle erreicht. Im Vergleich der Modelle konnte weiters festgestellt werden, dass die Arbeitshypothese  $H_0$  aufgrund der Ergebnisse verworfen werden kann.

Aus den Erkenntnissen der Vorevaluierung, der möglichen Datensätze, und den Ergebnissen aus dem Vergleich der erstellten Modelle, lässt sich klar feststellen, dass sich mittels Machine Learning eine präemptive Steuerung eines Smart Homes erzielen lässt.

Der Umfang der Möglichkeiten einer solchen Steuerung wird jedoch durch die verfügbaren Daten und Ressourcen eingeschränkt. Eine genaue Festlegung des möglichen Umfanges muss daher im jeweiligen Fall individuell geprüft werden.

## ABKÜRZUNGSVERZEICHNIS

AI.....	Artificial Intelligence
BLE .....	Bluetooth Low Energy
BLSTM .....	Bidirectional Long Short-Term Memory
CRISP-DM.....	Cross Industry Standard Process for Data Mining
dBm.....	Dezibel Milliwatt
HED.....	Home Energy Display
IoT.....	Internet of Things
KDD.....	Knowledge Discovery in Databases
LSTM.....	Long Short-Term Memory
MSE .....	Mean Squared Error
NaN .....	Not a Number
PCC.....	Pearson Korrelationskoeffizient
RMSE.....	Root Mean Square Error
RNN .....	Recurrent Neural Network
RSSI.....	Received Signal Strength Indication

## ABBILDUNGSVERZEICHNIS

Abbildung 1: Zusammenhänge des Ubiquitous Computing laut Springer Gabler (vgl. Lackes & Siepermann)	5
Abbildung 2: Generelle Cloud-Architektur eines Smart Home (in Anlehnung an Mocrii et al., 2018, S. 84)	6
Abbildung 3: Typen eines Smart Home Services (vgl. Balta-Ozkan et al., 2013, S. 362)	7
Abbildung 4: Ein Beispiel für die kontrollierten Umweltdienste eines Smart Home (in Anlehnung an Ali & Awad, 2018, S. 3)	11
Abbildung 5: Die weitere Entwicklung führt schließlich zur Konvergenz der Dienste. (vgl. Strese et al., 2010, S. 9)	12
Abbildung 6: Umsatz mit Smart Homes weltweit (in Anlehnung an Statista, 2020)	13
Abbildung 7: Anzahl Smart Homes weltweit (in Anlehnung an Statista, 2020)	14
Abbildung 8: Zugriff auf Smart Home Geräte über ein globales Netzwerk (in Anlehnung an Espinilla, Medina & Nugent, 2018, S. 7287)	18
Abbildung 9: Ebenen integrierter Smart Service-Plattformen (vgl. Acatech - Deutsche Akademie für Technikwissenschaften, 2015, S. 17)	21
Abbildung 10: Machine Learning Schlüsselaspekte und Semantik (vgl. Gollapudi & Laxmikanth, 2016, S. 2)	26
Abbildung 11: Exemplarische Rohdaten mit Labeling (vgl. Chollet, 2018, S. 7)	27
Abbildung 12: Exemplarisches Ergebnis zur besseren Repräsentation von gelabelten Daten (vgl. Chollet, 2018, S. 7)	27
Abbildung 13: Darstellung einer Mischform aus gelabelten und ungelabelten Daten (in Anlehnung an Gollapudi & Laxmikanth, 2016, S. 8)	27
Abbildung 14: Subfelder des Machine Learning (vgl. Gollapudi & Laxmikanth, 2016, S. 21)	28
Abbildung 15: Einordnung der Methoden im Portfolio der Wirtschaftsinformatik (vgl. Wilde & Hess, 2006, S. 14)	33
Abbildung 16: Forschungsentscheidungsstruktur (in Anlehnung an Wohlin & Aurum, 2015, S. 1431)	35
Abbildung 17: CRISP-DM Referenzmodell (in Anlehnung an Chapman et al., 2000, S. 13)	36
Abbildung 18: CRISP-DM Referenzmodell erweitert mit den zusätzlichen Methoden und Werkzeugen (in Anlehnung an Chapman et al., 2000, S. 13)	38
Abbildung 19: Modellierung und mögliche Anwendungsfelder (in Anlehnung an BITKOM, 2008, S. 12)	39
Abbildung 20: Binäre Sensoren im UJAml Smart Lab (in Anlehnung an Martínez-Santiago et al., 2020, S. 230)	44
Abbildung 21: Annäherungssensoren im UJAml Smart Lab (in Anlehnung an Espinilla, Martinez, et al., 2018, S. 34640)	45
Abbildung 22: Intelligenter Boden im UJAml Smart Lab (in Anlehnung an Espinilla, Medina & Nugent, 2018, S. 11)	46
Abbildung 23: Evaluierungsworkflow für die Daten eines Machine Learning Models (in Anlehnung an Zheng, 2015, S. 2)	47
Abbildung 24: Prototypentraining zur Bildung eines Machine Learning Modells (vgl. Zheng, 2015, S. 4)	50

## Abbildungsverzeichnis

---

Abbildung 25: Deep Learning innerhalb von ML und AI (vgl. Chollet, 2018, S. 18) .....	50
Abbildung 26: Distanzklassen für Annäherungssensor Bett im Validierungs Datensatz .....	56
Abbildung 27: Distanzklassen für Annäherungssensor Bett im Training Datensatz .....	56
Abbildung 28: Distanzklassen für Annäherungssensor Schranktür im Validierungs Datensatz .....	56
Abbildung 29: Distanzklassen für Annäherungssensor Schranktür im Training Datensatz .....	56
Abbildung 30: Distanzklassen für Annäherungssensor Kühlschrank im Validierungs Datensatz .....	56
Abbildung 31: Distanzklassen für Annäherungssensor Kühlschrank im Training Datensatz .....	56
Abbildung 32: Heatmap Aktivierungen der Bodensensoren im Validierungs Datensatz .....	57
Abbildung 33: Heatmap Aktivierungen der Boden Sensoren im Training Datensatz .....	57
Abbildung 34: Boxplot der Beschleunigungswerte im Validierungs Datensatz .....	57
Abbildung 35: Boxplot der Beschleunigungswerte im Training Datensatz .....	57
Abbildung 36: Verteilung der Z-Achsen Beschleunigungswerte im Validierungs Datensatz .....	57
Abbildung 37: Verteilung der Z-Achsen Beschleunigungswerte im Training Datensatz .....	57
Abbildung 38: Ergebnis des LSTM Epochen Tunings .....	59
Abbildung 39: Ergebnis des LSTM Batch-Größe Tunings .....	59
Abbildung 40: Ergebnis des LSTM Neuronen Tunings .....	59
Abbildung 41: Ergebnis des BLSTM Epochen Tunings .....	60
Abbildung 42: Ergebnis des BLSTM Batch-Größe Tunings .....	60
Abbildung 43: Ergebnis des BLSTM Neuronen Tunings .....	60
Abbildung 44: BLSTM Lernkurve .....	63
Abbildung 45: LSTM Lernkurve .....	63
Abbildung 46: Ausschnitt aus BLSTM Lernkurve .....	63
Abbildung 47: Ausschnitt aus LSTM Lernkurve .....	63
<i>Abbildung 48: MSE der geprüften Modelle .....</i>	<i>63</i>
<i>Abbildung 49: PCC der geprüften Modelle .....</i>	<i>63</i>
<i>Abbildung 50: RMSE der geprüften Modelle .....</i>	<i>63</i>
<i>Abbildung 51: R<sup>2</sup> der geprüften Modelle .....</i>	<i>63</i>

## TABELLENVERZEICHNIS

Tabelle 1: Komponenten der Heimautomatisierung (in Anlehnung an Strese et al., 2010, S. 8).....	15
Tabelle 2: Direkt auf die Bewohner/innen bezogene Produkte (in Anlehnung an Strese et al., 2010, S. 9) .....	15
Tabelle 3: Beischreibung und Beispiele für Hausenergie-Management-Systeme (in Anlehnung an Lobaccaro et al., 2016, S. 13ff) .....	16
Tabelle 4: Codes der binären Sensoren im UJAml Smart Lab.....	44
Tabelle 5: Codes der Annäherungssensoren im UJAml Smart Lab .....	45
Tabelle 6: Korrelation BLE RSSI Werte zu Distanz .....	48
Tabelle 7: Berechnung der eingesetzten Metriken .....	52
Tabelle 8: Minimaler und maximaler RMSE für LSTM Epochen Hyperparameter Tuning .....	59
Tabelle 9: Minimaler und maximaler RMSE für LSTM Batch-Größe Hyperparameter Tuning .....	59
Tabelle 10: Minimaler und maximaler RMSE für LSTM Neuronen Anzahl Hyperparameter Tuning .....	59
Tabelle 11 Minimaler und maximaler RMSE für BLSTM Epochen Hyperparameter Tuning .....	60
Tabelle 12 Minimaler und maximaler RMSE für BLSTM Batch-Größe Hyperparameter Tuning .....	60
Tabelle 13 Minimaler und maximaler RMSE für BLSTM Neuronen Anzahl Hyperparameter Tuning .....	60
Tabelle 14: Ergebnisse des Hyperparameter Tunings.....	61
Tabelle 15: Metriken des LSTM Modells.....	64
Tabelle 16: Metriken des BLSTM Modells .....	64

## LISTINGS

Listing 1: Implementierung des LSTM Modells.....	61
Listing 2: Implementierung des BLSTM Modells .....	61

## LITERATURVERZEICHNIS

- Acatech - Deutsche Akademie für Technikwissenschaften. (2015). *Smart Service Welt: Umsetzungsempfehlungen für das Zukunftsprojekt Internetbasierte Dienste für die Wirtschaft* [White Paper]. Berlin. Zugriff am 16.10.2020, verfügbar unter <https://www.acatech.de/publikation/zwischenbericht-smart-service-welt-umsetzungsempfehlungen-fuer-das-zukunftsprojekt-internetbasierte-dienste-fuer-die-wirtschaft/download-pdf?lang=de>
- Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L. & Savarese, S. (2016). Social LSTM: Human Trajectory Prediction in Crowded Spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (S. 961–971). IEEE.  
<https://doi.org/10.1109/CVPR.2016.110>
- Alam, M. R., Reaz, M. B. I. & Ali, M. A. M. (2012). A Review of Smart Homes—Past, Present, and Future. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6), 1190–1203. <https://doi.org/10.1109/TSMCC.2012.2189204>
- Ali, B. & Awad, A. I. (2018). Cyber and Physical Security Vulnerability Assessment for IoT-Based Smart Homes. *Sensors (Basel, Switzerland)*, 18(3).  
<https://doi.org/10.3390/s18030817>
- Allmendinger, G. & Lombreglia, R. (2005). *Four Strategies for the Age of Smart Services*. Zugriff am 13.04.2020, verfügbar unter <https://hbr.org/2005/10/four-strategies-for-the-age-of-smart-services>
- Balta-Ozkan, N., Boteler, B. & Amerighi, O. (2014). European smart home market development: Public views on technical and economic aspects across the United Kingdom, Germany and Italy. *Energy Research & Social Science*, 3, 65–77.  
<https://doi.org/10.1016/j.erss.2014.07.007>
- Balta-Ozkan, N., Davidson, R., Bicket, M. & Whitmarsh, L. (2013). The development of smart homes market in the UK. *Energy*, 60, 361–372. <https://doi.org/10.1016/j.energy.2013.08.004>
- Barlow, J. & Gann, D. (1998). A changing sense of place: are integrated IT systems reshaping the home? In *Technological Futures, Urban Futures Conference*, Durham.
- Begleitforschung Mittelstand-Digital (2016). Wissenschaft trifft Praxis. *Mittelstand-Digital*, 4. Zugriff am 08.03.2021, verfügbar unter <https://www.mittelstand-digital.de/MD/Redaktion/DE/Publikationen/Wissenschaft-trifft-Praxis/magazin-wissenschaft-trifft-praxis-ausgabe4.html>
- Bertuletti, S., Cereatti, A., Caldara, M., Galizzi, M. & Della Croce, U. (2016). Indoor distance estimated from Bluetooth Low Energy signal strength: Comparison of regression models. In

- 2016 *IEEE Sensors Applications Symposium (SAS)*. IEEE.  
<https://doi.org/10.1109/SAS.2016.7479899>
- Briere, D. D. & Hurley, P. J. (2007). *Smart homes for dummies* (3. Aufl.). Wiley.
- Brumitt, B., Meyers, B., Krumm, J., Kern, A. & Shafer, S. (2000). EasyLiving: Technologies for Intelligent Environments. In G. Goos, J. Hartmanis, J. van Leeuwen, P. Thomas & H.-W. Gellersen (Hg.), *Lecture Notes in Computer Science. Handheld and Ubiquitous Computing* (Bd. 1927, S. 12–29). Springer Berlin Heidelberg. [https://doi.org/10.1007/3-540-39959-3\\_2](https://doi.org/10.1007/3-540-39959-3_2)
- Bullinger, H.-J. (30. Juli 2015). Industrie 4.0? Service 4.0!: Wem gehört künftig der Kunde? *manager magazin*. Zugriff am 18.10.2020, verfügbar unter <https://www.manager-magazin.de/digitales/it/industrie-4-0-wem-gehört-künftig-der-kunde-a-1045769.html>
- Bullinger, H.-J., Ganz, W. & Neuhüttler, J. (2017). Smart Services – Chancen und Herausforderungen digitalisierter Dienstleistungssysteme für Unternehmen. In M. Bruhn & K. Hadwich (Hg.), *Dienstleistungen 4.0* (S. 97–120). Springer Fachmedien Wiesbaden. [https://doi.org/10.1007/978-3-658-17550-4\\_4](https://doi.org/10.1007/978-3-658-17550-4_4)
- Bundesverband Informationswirtschaft (Hg.). (2008). *Studienreihe zur Heimvernetzung: Konsumentennutzen und persönlicher Komfort* [White Paper]. Zugriff am 14.03.2021, verfügbar unter <https://www.bitkom.org/sites/default/files/file/import/Studie-Konsumentennutzen.pdf>
- Burkov, A. (2019). *Machine Learning kompakt: Alles, was Sie wissen müssen* (1. Auflage). MITP Verlags GmbH & Co. KG.
- Cambra Baseca, C., Sendra, S., Lloret, J. & Tomas, J. (2019). A Smart Decision System for Digital Farming. *Agronomy*, 9(5), Artikel 216. <https://doi.org/10.3390/agronomy9050216>
- Carvalho, R. M., Andrade, R. M., Oliveira, K. M [Kathia M.] & Kolski, C. (2018). Catalog of Invisibility Requirements for UbiComp and IoT Applications. In *2018 IEEE 26th International Requirements Engineering Conference (RE)* (S. 88–99). IEEE. <https://doi.org/10.1109/RE.2018.00019>
- Carvalho, R. M., Castro Andrade, R. M. de, Oliveira, K. M. de [Káthia Marçal], Sousa Santos, I. de & Bezerra, C. I. M. (2017). Quality characteristics and measures for human–computer interaction evaluation in ubiquitous systems. *Software Quality Journal*, 25(3), 743–795. <https://doi.org/10.1007/s11219-016-9320-z>
- Chaitanya Ekanadham. (2018). *Using Machine Learning to Improve Streaming Quality at Netflix*. Zugriff am 15.03.2020, verfügbar unter <https://netflixtechblog.com/using-machine-learning-to-improve-streaming-quality-at-netflix-9651263ef09f>
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. R. & Wirth, R. (2000). *CRISP-DM 1.0: Step-by-step data mining guide*. CRISP-DM consortium.

- Chhetri, M., Kumar, S., Pratim Roy, P. & Kim, B.-G. (2020). Deep BLSTM-GRU Model for Monthly Rainfall Prediction: A Case Study of Simtokha, Bhutan. *Remote Sensing*, 12(19), Artikel 3174. <https://doi.org/10.3390/rs12193174>
- Chimamiwa, G. (2020). *Multi-sensor dataset of human activities in a smart home environment*. Mendeley Data, V1. <https://doi.org/10.17632/t9n68yfk3.1>
- Chollet, F. (2018). *Deep learning with Python*. *Safari Tech Books Online*. Manning.
- Chowdhury, M. A., Apon, A. & Dey, K. (Hg.). (2017). *Data analytics for intelligent transportation systems*. Elsevier.
- Davies, N. & Clinch, S. (2017). Pervasive Data Science. *IEEE Pervasive Computing*, 16(3), 50–58. <https://doi.org/10.1109/MPRV.2017.2940956>
- Deloitte Deutschland. (2018). *Deloitte Smart Home Studie 2018: Ausgewählte Ergebnisse für den deutschen Markt* [White Paper]. Zugriff am 08.03.2021, verfügbar unter <https://www2.deloitte.com/de/de/pages/technology-media-and-telecommunications/articles/smart-home-studie-2018.html>
- Espinilla, M., Martinez, L., Medina, J. & Nugent, C. (2018). The Experience of Developing the UJAml Smart Lab. *IEEE Access*, 6, 34631–34642. <https://doi.org/10.1109/ACCESS.2018.2849226>
- Espinilla, M., Medina, J. & Nugent, C. (2018). UCaml Cup. Analyzing the UJA Human Activity Recognition Dataset of Activities of Daily Living. In *The 12th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAml 2018)*. Symposium im Rahmen der Tagung von Punta Cana, Dominican Republic: MDPI.
- Fabi, V., Spigliantini, G. & Corgnati, S. P. (2017). Insights on Smart Home Concept and Occupants' Interaction with Building Controls. *Energy Procedia*, 111, 759–769. <https://doi.org/10.1016/j.egypro.2017.03.238>
- Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3), 37–54. <https://doi.org/10.1609/aimag.v17i3.1230>
- Forman, G. H. & Zahorjan, J. (1994). The challenges of mobile computing. *Computer*, 27(4), 38–47. <https://doi.org/10.1109/2.274999>
- Gann, D., Barlow, J. & Venables, T. (1999). *Digital futures: Making homes smarter*. Chartered Institute of Housing.
- Ganz, W., Neuhüttler, J. & Tombeil, A.-S. (2016). Dienstleistungsproduktivität und -qualität: eine kritische Würdigung. In M. Gouthier (Hg.), *Kundenbindung durch kosteneffiziente Service Excellence: Strategien* (S. 51–68). NOMOS VERLAGSGESELLSCHAFT. <https://doi.org/10.5771/9783845274461-51>

- Géron, A. (2018). *Praxiseinstieg Machine Learning mit Scikit-Learn und TensorFlow: Konzepte, Tools und Techniken für intelligente Systeme* ((K. Rother, Übers.)) (1. Auflage). O'Reilly.
- Gibbs, W. W. (2000). As We May Live. *Scientific American*, 283(5), 36–40.  
<https://doi.org/10.1038/scientificamerican1100-36>
- Gollapudi, S. & Laxmikanth, V. (2016). *Practical Machine Learning. Community Experience Distilled*. Packt Publishing.
- Gram-Hanssen, K. & Darby, S. J. (2018). “Home is where the smart is”? Evaluating smart home research and approaches against the concept of home. *Energy Research & Social Science*, 37, 94–101. <https://doi.org/10.1016/j.erss.2017.09.037>
- Haneke, U., Trahasch, S., Zimmer, M. & Felden, C. (2019). *Data Science: Grundlagen, Architekturen und Anwendungen. Edition TDWI*. dpunkt.verlag.
- Harper, R. (2003). *Inside the Smart Home*. Springer-Verlag London Limited.  
<https://doi.org/10.1007/b97527>
- Holroyd, P., Watten, P. & Newbury, P. (2010). Why Is My Home Not Smart? In Y. Lee, Z. Z. Bien, M. Mokhtari, J. T. Kim, M. Park, J. Kim, H. Lee & I. Khalil (Hg.), *Lecture Notes in Computer Science. 0302-9743: Bd. 6159, Aging Friendly Technology for Health and Independence: 8th International Conference on Smart Homes and Health Telematics, ICOST 2010, Seoul, Korea, June 22-24, 2010. Proceedings* (S. 53–59). Springer-Verlag Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-13778-5\\_7](https://doi.org/10.1007/978-3-642-13778-5_7)
- Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z. & Zhang, H. (2019). Deep Learning with Long Short-Term Memory for Time Series Prediction. *IEEE Communications Magazine*, 57(6), 114–119.  
<https://doi.org/10.1109/MCOM.2019.1800155>
- King, N. (2003). *Smart home – a definition*.
- Knuth, D. (1990). Arthur Lee Samuel, 1901-1990. *TUGboat*, 11(4), 497–498.
- Lackes, R. & Siepermann, M. *Ubiquitous Computing*. Zugriff am 14.03.2021, verfügbar unter <https://wirtschaftslexikon.gabler.de/definition/ubiquitous-computing-48216>
- Lashkari, Chen & Musilek (2019). Energy Management for Smart Homes—State of the Art. *Applied Sciences*, 9(17), Artikel 3459. <https://doi.org/10.3390/app9173459>
- Li Jiang, Liu, D.-Y. & Yang, B. (2004, August). Smart home research. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)* (S. 659–663). IEEE. <https://doi.org/10.1109/ICMLC.2004.1382266>
- Lobaccaro, G., Carlucci, S. & Löfström, E. (2016). A Review of Systems and Technologies for Smart Homes and Smart Grids. *Energies*, 9(5), Artikel 348.  
<https://doi.org/10.3390/en9050348>

- Lutolf, R. (1992). Smart Home concept and the integration of energy meters into a home based system. In *Seventh International Conference on Metering Apparatus and Tariffs for Electricity Supply 1992*. Symposium im Rahmen der Tagung von Institution of Electrical Engineers. Power Division.
- Marikyan, D., Papagiannidis, S. & Alamanos, E. (2019). A systematic review of the smart home literature: A user perspective. *Technological Forecasting and Social Change*, 138, 139–154. <https://doi.org/10.1016/j.techfore.2018.08.015>
- Martínez-Santiago, F., García-Viedma, M. R., Williams, J. A., Slater, L. T. & Gkoutos, G. V. (2020). Aging Neuro-Behavior Ontology. *Applied Ontology*, 15(2), 219–239. <https://doi.org/10.3233/AO-200229>
- McCarthy, J., Minsky, M. L., Rochester, N. & Shannon, C. (1955). *A PROPOSAL FOR THE DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE*. DARTMOUTH COLLEGE. Zugriff am 14.03.2021, verfügbar unter <http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html>
- Mendes, T., Godina, R., Rodrigues, E., Matias, J. & Catalão, J. (2015). Smart Home Communication Technologies and Applications: Wireless Protocol Assessment for Home Area Network Resources. *Energies*, 8(7), 7279–7311. <https://doi.org/10.3390/en8077279>
- Mennicken, S. & Huang, E. M. (2012). Hacking the Natural Habitat: An In-the-Wild Study of Smart Homes, Their Development, and the People Who Live in Them. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, J. Kay, P. Lukowicz, H. Tokuda, P. Olivier & A. Krüger (Hg.), *Lecture Notes in Computer Science. Pervasive Computing* (Bd. 7319, S. 143–160). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-31205-2\\_10](https://doi.org/10.1007/978-3-642-31205-2_10)
- Mennicken, S., Vermeulen, J. & Huang, E. M. (2014). From today's augmented houses to tomorrow's smart homes. In A. J. Brush, A. Friday, J. Kientz, J. Scott & J. Song (Hg.), *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '14 Adjunct* (S. 105–115). ACM Press. <https://doi.org/10.1145/2632048.2636076>
- Meyer, H., Sancho, J. C., Quiroga, J. V., Zyulkyarov, F., Roca, D. & Nemirovsky, M. (2017). Disaggregated Computing. An Evaluation of Current Trends for Datacentres. *Procedia Computer Science*, 108, 685–694. <https://doi.org/10.1016/j.procs.2017.05.129>
- Mocrii, D., Chen, Y. & Musilek, P. (2018). IoT-based smart homes: A review of system architecture, software, communications, privacy and security. *Internet of Things*, 1-2, 81–98. <https://doi.org/10.1016/j.iot.2018.08.009>

- Ocegueda-Miramontes, V., Sanchez, M. A. & Aguilar, L. (2019). Towards Intelligent Systems for Ubiquitous Computing: Tacit Knowledge-Inspired UbiComp. In M. A. Sanchez, L. Aguilar, M. Castañón-Puga & A. Rodríguez (Hg.), *Studies in Systems, Decision and Control. Applied Decision-Making* (Bd. 209, S. 65–94). Springer International Publishing.  
[https://doi.org/10.1007/978-3-030-17985-4\\_4](https://doi.org/10.1007/978-3-030-17985-4_4)
- Papaioannou, A. D., Nejabati, R. & Simeonidou, D. (2016). The Benefits of a Disaggregated Data Centre: A Resource Allocation Approach. In *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE. <https://doi.org/10.1109/GLOCOM.2016.7842314>
- Park, E., Kim, S., Kim, Y. & Kwon, S. J. (2018). Smart home services as the next mainstream of the ICT industry: determinants of the adoption of smart home services. *Universal Access in the Information Society*, 17(1), 175–190. <https://doi.org/10.1007/s10209-017-0533-0>
- Rashid, T. (2017). *Neuronale Netze selbst programmieren: Ein verständlicher Einstieg mit Python* ((F. Langenau, Übers.)) (1. Auflage). O'Reilly.
- Risteska Stojkoska, B. L. & Trivodaliev, K. V. (2017). A review of Internet of Things for smart home: Challenges and solutions. *Journal of Cleaner Production*, 140, 1454–1464.  
<https://doi.org/10.1016/j.jclepro.2016.10.006>
- Robles, R. J. & Kim, T. (2010). Applications, Systems and Methods in Smart Home Technology: A Review. *International Journal of Advanced Science and Technology*, 15.
- Roe, P. R. W. (2007). *Towards an inclusive future: Impact and wider potential of information and communication technologies*. COST 219ter.
- Russell, S. J. & Norvig, P. (2012). *Künstliche Intelligenz: Ein moderner Ansatz* (3. Aufl.). *Always learning*. Pearson.
- Sammut, C. & Webb, G. I. (Hg.). (2017). *Springer reference. Encyclopedia of machine learning and data mining*. Springer.
- Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), 210–229. <https://doi.org/10.1147/RD.33.0210>
- Satpathy, L. (2006). *Smart Housing: Technology to Aid Aging in Place: New Opportunities and Challenges*. Mississippi State University. Zugriff am 17.10.2020, verfügbar unter [http://sun.library.msstate.edu/ETD-db/theses/available/etd-06082006-012243/unrestricted/Lalatendu\\_Satpathy\\_Architecture\\_Final.pdf](http://sun.library.msstate.edu/ETD-db/theses/available/etd-06082006-012243/unrestricted/Lalatendu_Satpathy_Architecture_Final.pdf)
- Satyanarayanan, M. (2001). Pervasive computing: vision and challenges. *IEEE Personal Communications*, 8(4), 10–17. <https://doi.org/10.1109/98.943998>
- Schoder, D., Bichler, M., Buhl, U., Hess, T., Krcmar, H. & Sinz, E. (2009-2011). *Ergebnis der Arbeitsgruppe "Profil der Wirtschaftsinformatik"*. WKWI und GI FB WI.

- Scott, F. (2007). *Teaching homes to be green: Smart homes and the environment*. Green Alliance.
- Silva, L. C. de, Morikawa, C. & Petra, I. M. (2012). State of the art of smart homes. *Engineering Applications of Artificial Intelligence*, 25(7), 1313–1321.  
<https://doi.org/10.1016/j.engappai.2012.05.002>
- Statista. (2020). *Smart Home - worldwide* [zitiert nach de.statista.com]. Zugriff am 17.10.2020, verfügbar unter <https://www.statista.com/outlook/283/100/smart-home/worldwide>
- Strengers, Y. (2014). Smart energy technologies in everyday life: smart Utopia? *Technology Analysis & Strategic Management*, 26(10), 1242–1247.  
<https://doi.org/10.1080/09537325.2014.975789>
- Strese, H., Seidel, U., Knappe, T. & Botthof, A. (2010). *Smart Home in Deutschland: Untersuchung im Rahmen der wissenschaftlichen Begleitung zum Programm Next Generation Media (NGM) des Bundesministeriums für Wirtschaft und Technologie*. iit in der VDI/VDE-IT.
- Suryadevara, N. K. & Mukhopadhyay, S. C. (Hg.). (2015). *Smart Sensors, Measurement and Instrumentation. Smart homes: Design, implementation and issues*. Springer.  
<https://doi.org/10.1007/978-3-319-13557-1>
- Taylor, A. S., Harper, R., Swan, L., Izadi, S., Sellen, A. & Perry, M. (2007). Homes that make us smart. *Personal and Ubiquitous Computing*, 11(5), 383–393. <https://doi.org/10.1007/s00779-006-0076-5>
- Temdee, P. & Prasad, R. (2018). *Context-Aware Communication and Computing: Applications for Smart Environment*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-59035-6>
- van Berlo, A. (2002). Smart home technology: Have older people paved the way? *Gerontechnology*, 2(1). <https://doi.org/10.4017/gt.2002.02.01.010.00>
- van Berlo, A. (2005). *Developments in Smart Homes and Smart Living: From real needs to electronic lifestyle*. Smart Homes Association. International Cablemakers Federation, Tokyo. Zugriff am 17.10.2020, verfügbar unter [http://www.icf.at/fileadmin/user\\_upload/Congresses/2005/17.vanBerlo.pdf](http://www.icf.at/fileadmin/user_upload/Congresses/2005/17.vanBerlo.pdf)
- van Berlo, A. & Allen, B. (1999). *Design Guidelines on Smart Homes: A COST 219bis Guidebook*. COST, European Co-Operation in the Field of Scientific and Technical Research.
- Wang, J., Tang, J., Xu, Z., Wang, Y., Xue, G., Zhang, X. & Yang, D. (2017). Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach. In

- IEEE INFOCOM 2017 - IEEE Conference on Computer Communications* (S. 1–9). IEEE.  
<https://doi.org/10.1109/INFOCOM.2017.8057090>
- Wartala, R. (2018). *Praxiseinstieg Deep Learning: Mit Python, Caffe, TensorFlow und Spark eigene Deep-Learning-Anwendungen erstellen* (1. Auflage). O'Reilly.
- Weiser, M. (1991). The Computer for the 21 st Century. *Scientific American*, 265(3), 94–105.
- Wilde, T. & Hess, T. (2006). *Methodenspektrum der Wirtschaftsinformatik: Überblick und Portfoliobildung* [Working Paper]. München. Ludwig-Maximilians-Universität München, Institut für Wirtschaftsinformatik und Neue Medien (WIM).
- Wilde, T. & Hess, T. (2007). Forschungsmethoden der Wirtschaftsinformatik. *WIRTSCHAFTSINFORMATIK*, 49(4), 280–287. <https://doi.org/10.1007/s11576-007-0064-z>
- Winkler, B. (2002). An Implementation of an Ultrasonic Indoor Tracking System Supporting the OSGi Architecture of the ICTA Lab.
- Winston, P. H. (1993). *Artificial intelligence* (3., aktualisierte Aufl.). Addison-Wesley.
- Wohlin, C. & Aurum, A. (2015). Towards a decision-making structure for selecting a research design in empirical software engineering. *Empirical Software Engineering*, 20(6), 1427–1455.  
<https://doi.org/10.1007/s10664-014-9319-7>
- Zheng, A. (2015). *Evaluating Machine Learning Models* (1. Aufl.). O'Reilly Media, Inc.
- Zou, Q., Xiong, Q., Li, Q., Yi, H., Yu, Y. & Wu, C. (2020). A water quality prediction method based on the multi-time scale bidirectional long short-term memory network. *Environmental science and pollution research international*, 27(14), 16853–16864.  
<https://doi.org/10.1007/s11356-020-08087-7>