



TECHNISCHE  
UNIVERSITÄT  
WIEN

D I P L O M A R B E I T

# Simulation eines Heizungssystems mit nichtinvasivem Energiemonitoring

ausgeführt am

Institut für  
Analysis und Scientific Computing  
TU Wien

unter der Anleitung von

**Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Felix Breitenecker**

durch

**Daniel Aufhauser**

Matrikelnummer: 1126021

Feldgasse 1

3130 Herzogenburg

# Kurzfassung

Die vorliegende Diplomarbeit beschäftigt sich mit der Entwicklung eines nichtinvasiven Energiemonitoring-Systems für den thermischen Energiebedarf von Gebäuden. Im Verlauf dieser Arbeit wird analysiert und beschrieben welche Informationen über ein Heizungssystem aus verschiedenen Sensoren herausgelesen werden. Neben der Aufspaltung des Gesamtenergieverbrauchs auf einzelne Verbraucher wird ein Augenmerk auf das automatische Erkennen fehlerhafter Komponenten des Heizungssystems gelegt. Der nichtinvasive Zugang ermöglicht eine preiswerte Integration eines Energiemonitoring-Systems in bereits bestehende Gebäude und deren Heizungssysteme. Speziell soll neben dem Wärmehzähler nur kostengünstige Hardware, beispielsweise Thermometer, verwendet werden.

Am Beginn des Prozesses steht ein Simulationsmodell für ein Heizungssystem eines Gebäudes, mit dem ausreichend Testdaten zu Raumtemperatur, Heizleistung und anderen Werten generiert werden sollen. Dieses Modell umfasst mehrere Stockwerke und beschreibt neben dem Heizungssystem mit Heizkörpern und deren Regelung auch den Wärmeaustausch zwischen den Räumen sowie Umgebungsbedingungen. Zur Analyse der Datensätze im Rahmen des Energiemonitorings werden anschließend vier Ansätze verfolgt: ein Hidden Markov Models, ein linearisiertes Modell zur Schätzung von Systemparametern, Clustering mittels Support Vector Machines und ein Autoregressions-Modell. Die erste Herangehensweise versucht mehrere Hidden Markov Models miteinander verschränkt zu trainieren und so Aussagen über den Einzelverbrauch zu treffen. Hierbei dient ausschließlich der Gesamtenergieverbrauch als Grundlage für die Berechnung. Die größten Erfolge werden mit einem Algorithmus erreicht, der den Energieeintrag eines Heizkörpers aus der Gesamtenergie und einem linearisierten Modell der Wärmeübertragung in den Raum schätzt. Jeder neue Heizzyklus fließt in die existierenden Berechnungen ein und verbessert die jeweilige Schätzung. Das Clustering mittels Support Vector Machines untersucht die Unterschiede im Heizverhalten der einzelnen Räume. Ein Algorithmus basierend auf einem Autoregressions-Modell erlaubt es, atypisches Heizverhalten zu erkennen. Die Methode stellt sicher, dass keine fehlerhaften Daten das Energiemonitoring des Einzelverbrauchs verfälschen.

Die in der Arbeit festgehaltenen Ergebnisse liefern einen Überblick über die Anwendungsgebiete von Verfahren für Energiemonitoring-Systeme. Zusätzlich zum Erreichen und Einhalten von Energiesparzielen sind solche Systeme hilfreich für Gebäudeverwalter, die auf Basis der Erkennung von atypischen Heizverhalten Wartungszyklen optimieren können.

# Abstract

This thesis deals with the development of an energy monitoring system for thermal energy consumption of buildings. The information that can be extracted from different sensors of a heating system are described and analysed in this work. Beside the disaggregation of the energy consumption, a focus lies on recognizing erroneous components of the heating system. Non-invasive sensors allow effortless and cheap installation in existing building and their heating. Just a heat meter and additional hardware is used for the energy monitoring system.

To start this process, a simulation model of a building is created, which generates test data for room temperature, heating power and other values for later research. The model contains three floors and describes the heat flow between the rooms and as well as with the environment. To analyse the data in the context of energy monitoring, four attempts are formulated: a Hidden Markov Model, a linearized model for parameter estimation, clustering based on a Support Vector Machine, and an auto-regression model. The first approach involves several Hidden Markov Models which are trained in parallel. The overall energy consumption is the only data used for this training. The biggest success are reached with an algorithm that estimates the energy consumption of one heater from the overall consumption and a linearized model of the heat flow into the room. The clustering with a Support Vector Machine investigates the heating behavior of each room. The last approach allows to recognise atypical heating behavior, based on an auto-regression model. This method is used to remove erroneous data from the energy monitoring.

The listed results offer an overview of methods for energy monitoring systems. These systems help reaching energy saving goals and allow facility manager to schedule maintenance cycles.

# Danksagung

An dieser Stelle möchte ich all jenen danken, die durch ihre fachliche und persönliche Unterstützung zum Gelingen dieser Diplomarbeit beigetragen haben. Zuerst gebührt mein Dank Herrn Professor Breitenecker und Herrn Dipl.-Ing. Bernhard Heinzl, die meine Arbeit betreut und begutachtet haben. Die hilfreichen Anregungen und die konstruktive Kritik meiner Betreuer bei Siemens, Dr. Konrad Diwold und Dr. Lukas Krammer, erleichterten mir ebenfalls das Verfassen dieser Arbeit.

Ich bedanke mich bei dem Unternehmen Siemens AG Österreich, welches mir die notwendigen zeitlichen und wissenschaftlichen Ressourcen zur Verfügung gestellt hat. Die vorliegende Diplomarbeit ist im Zuge meiner Tätigkeit für die Siemens AG Österreich als Teil eines umfangreicheren Forschungsprojekts entstanden. Hier möchte ich auch die große Gruppe meiner Kollegen nicht unerwähnt lassen und mich bei ihnen für die zahlreichen interessanten Debatten und Ideen, die maßgeblich dazu beigetragen haben, dass die Diplomarbeit in dieser Form vorliegt, bedanken.

Ein besonderer Dank gilt auch meiner Tante Mag. Andrea Pinz für das Lektorat. Abschließend bedanke ich mich bei meinen Eltern und meiner Familie, die mir mein Studium durch ihre Unterstützung ermöglicht und mich mit Hilfestellungen und aufmunternden Worten durch diese Arbeit begleitet haben.

# Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 22. Oktober 2017

---

Daniel Aufhauser

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Vorgehensweise . . . . .	2
<b>2</b>	<b>Problemstellung</b>	<b>4</b>
2.1	Invasives versus Nichtinvasives Monitoring . . . . .	4
2.2	Monitoring-System für Heizungssysteme - State of the Art . . . . .	5
2.3	Non-Intrusive Load Monitoring NILM - State of the Art . . . . .	6
2.3.1	Ein Überblick . . . . .	6
2.3.2	Elektrische Energie . . . . .	7
<b>3</b>	<b>Simulationsmodell</b>	<b>10</b>
3.1	Modellierung der Wärmeflüsse . . . . .	10
3.1.1	Heizkörper . . . . .	11
3.1.2	Heizkörperthermostat . . . . .	13
3.1.3	Wohnraum . . . . .	14
3.1.4	Wand und Fenster . . . . .	14
3.2	Implementierung in Matlab/Simulink . . . . .	15
3.2.1	Modellaufteilung . . . . .	15
3.2.2	Konstanten . . . . .	17
3.3	Ergebnisse und besondere Auffälligkeiten . . . . .	20
<b>4</b>	<b>Energiemonitoring</b>	<b>23</b>
4.1	Zugang 1: Hidden Markov Modell HMM . . . . .	23
4.1.1	Vorbemerkungen . . . . .	23
4.1.2	Algorithmus . . . . .	24
4.1.3	Implementierung . . . . .	32
4.1.4	Ergebnisse und Probleme . . . . .	33
4.2	Zugang 2: Linearisiertes Modell . . . . .	35
4.2.1	Vorbemerkungen . . . . .	35
4.2.2	Algorithmus . . . . .	36

## *Inhaltsverzeichnis*

---

4.2.3	Ergebnisse und Probleme . . . . .	39
4.3	Zugang 3: Clustern . . . . .	43
4.3.1	Vorbemerkungen . . . . .	43
4.3.2	Support Vector Machine SVM . . . . .	46
4.3.3	Ergebnisse und Probleme . . . . .	52
4.4	Zugang 4: Atypisches Heizverhalten . . . . .	54
4.4.1	Vorbemerkungen . . . . .	54
4.4.2	Autoregressions-Modell AR . . . . .	55
4.4.3	Ergebnisse und Probleme . . . . .	59
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>61</b>
<b>6</b>	<b>Anhang</b>	<b>63</b>
	<b>Abbildungsverzeichnis</b>	<b>67</b>
	<b>Tabellenverzeichnis</b>	<b>68</b>
	<b>Literaturverzeichnis</b>	<b>69</b>

# 1 Einleitung

Auf der Grundlage des Pariser Klimaabkommens 2015 hat sich die Europäische Kommission verpflichtet, den Energieverbrauch bis 2020 um 20% zu reduzieren. In privaten Haushalten wird mit 24.8%<sup>1</sup> annähernd ein Viertel der Gesamtenergie verbraucht. Somit sind Strategien in Wohngebäuden Energie einzusparen, zu grundlegenden Fragestellungen geworden. Energiemonitoring-Systeme erweisen sich dabei als zielführende Unterstützung für Bewohner von Häusern. Das Anwendungsgebiet von Monitoring-Systemen im thermischen Bereich beschränkt sich ausschließlich auf die Abrechnung der Heizkosten beim Energieversorger im Falle einer Fernwärmeheizung und auf die Aufteilung der Heizkosten in Mehrparteienhäusern. Im Gegensatz dazu geben ähnliche Systeme, die auf Basis der elektrischen Energie arbeiten, eine genaue Aufschlüsselung des Einzelverbrauchs jedes elektrischen Geräts im überwachten Bereich an. Da üblicherweise aufgrund des schlechten Wirkungsgrades einer Elektroheizung in den mittel- und nordeuropäischen Staaten vorwiegend fossile Brennstoffe oder Holz zum Einsatz kommen, kann die beim Heizen verbrauchte Wärmeenergie von herkömmlichen Monitoring-Systemen nur erfasst und nicht analysiert beziehungsweise aufgeschlüsselt werden. In der vorliegenden Arbeit suchen wir nach Möglichkeiten analog, zu einem elektrischen ein thermisches Monitoring System zu entwickeln.

Zur Messung von Wärmeenergie benötigt man einen Wärmehähler, wie beispielsweise jenen der Firma *Techem* in Abbildung 1.1<sup>2</sup>. Dieser misst den Volumenstrom des zirkulierenden Mediums im System sowie die Differenz zwischen Vor- und Rücklauftemperatur. Anschließend errechnet der Zähler aus dem Produkt der beiden Faktoren und aufsummieren die Wärmeenergie. Für diese Messung muss ein massiver Eingriff in das Heizungssystem durchgeführt werden, falls die Sensoren nicht bereits verbaut sind. Zusätzlich betragen die Kosten für einen einzelnen Wärmehähler mehr als 100€. Möchte man nun in einem Haus den exakten Wärmeenergieverbrauch jedes Raumes beziehungsweise Heizkörpers auswerten und optimieren, stehen die anfallenden Kosten für die Installation in keiner Relation zu den möglichen Einsparungen im Energieverbrauch. In vielen Fällen ist eine exakte Messung des

---

<sup>1</sup>Quelle: [http://ec.europa.eu/eurostat/statistics-explained/index.php/Consumption\\_of\\_energy/de](http://ec.europa.eu/eurostat/statistics-explained/index.php/Consumption_of_energy/de)

<sup>2</sup>Quelle: <https://www.techem.de/bautraeger-und-fachplaner/erfassungs-und-ablesungsgeraete/kaelte-waermezaehler/techem-messkapsel-kaelte-waermezaehler.html>



Abbildung 1.1: Wärmehähler der Firma *Techem*

Energieverbrauchs nicht von Nöten, um die Sparziele der Europäischen Union zu erreichen. Eine Schätzung des thermischen Energieverbrauchs reicht aus Räume mit erhöhten Verbrauch zu lokalisieren und deren Verhalten zu optimieren. Für dieses Einsatzgebiet bietet sich ein Energiemonitoring-System, welches im Gegensatz zu im Einsatz befindlichen Systemen den Verbrauch nicht misst sondern aus bestimmten Messwerten einfach zu verbauender Sensoren ableitet. Zusätzlich zum Erreichen und Einhalten von Energiesparzielen sind solche Systeme hilfreich für Gebäudeverwalter, die auf Basis der Erkennung von atypischen Heizverhalten Wartungszyklen optimieren können.

Das Analysieren und Aufschlüsseln von Daten, die aus einem Heizungssystem stammen, ist das übergeordnete Ziel der vorliegenden Arbeit. Neben der Aufspaltung des Gesamtenergieverbrauchs auf einzelne Verbraucher wird ein Augenmerk auf das automatische Erkennen fehlerhafter Komponenten des Heizungssystems gelegt. Der nichtinvasive Zugang spiegelt sich in der Verwendung kostengünstiger Hardware, beispielsweise Thermometer, neben einem einzigen Wärmehähler wider.

### 1.1 Vorgehensweise

Die nachfolgende Arbeit gliedert sich in zwei Hauptteile. Der erste beschäftigt sich mit dem Erzeugen von Versuchsdaten. Hierfür erstellen wir ein Modell eines Hauses in *Simulink*, welches uns alle Wärmeflüsse, Temperaturwerte und Thermostatstellungen aufzeichnet. Dieses

Modell umfasst mehrere Stockwerke mit insgesamt 39 Räumen und beschreibt neben dem Heizungssystem mit Heizkörpern und deren Regelung auch den Wärmeaustausch zwischen den Räumen sowie Umgebungsbedingungen.

Im zweiten Teil wenden wir uns der Analyse der Simulationsdaten zu. Wir entwickeln und verfolgen vier verschiedene Zugänge, um das Energiemonitoring-System zu implementieren: Hidden Markov Models, ein linearisiertes Modell zur Schätzung von Systemparametern, Clustering mittels Support Vector Machines und ein Autoregressions-Modell. Da wir vor der Aufgabe stehen, große Datenmengen zu analysieren, wird auf Basis von selbst lernenden Algorithmen operiert. So sollen qualitativ möglichst hochwertige Ergebnisse erzielt werden. Dank dieser Vorgehensweise wird zeitgleich die Anfälligkeit für Störungen reduziert.

Die erste Herangehensweise versucht mehrere Hidden Markov Models miteinander verschränkt zu trainieren und so Aussagen über den Einzelverbrauch zu treffen. Hierbei dient ausschließlich der Gesamtenergieverbrauch als Grundlage für die Berechnung. Die größten Erfolge werden mit einem Algorithmus erreicht, er den Energieeintrag eines Heizkörpers aus der Gesamtenergie und einem linearisierten Modell der Wärmeübertragung in den Raum schätzt. Jeder neue Heizzyklus fließt in die existierenden Berechnungen ein und verbessert die jeweilige Schätzung. Das Clustering mittels Support Vector Machines untersucht die Unterschiede im Heizverhalten der einzelnen Räume. Ein Algorithmus, basierend auf einem Autoregressions-Modell, erlaubt es, atypisches Heizverhalten zu erkennen. Die Methode stellt sicher, dass keine fehlerhaften Daten das Energiemonitoring des Einzelverbrauchs verfälschen.

Die Ergebnisse, die den Algorithmen jeweils beigelegt sind, liefern einen Überblick über die Anwendungsgebiete von Verfahren für Energiemonitoring-System.

## 2 Problemstellung

### 2.1 Invasives versus Nichtinvasives Monitoring

Das Energiesparpotential ist in low-tech oder Bestandsgebäuden höher als in modernen high-tech Bauten, da sie aufgrund ihrer Langlebigkeit nur die Baustandards von zehn oder zwanzig Jahren zuvor erfüllen. Es gibt jedoch einige Konzepte, die sich mit *retrofitting*, dem Prozess, welcher low-tech Gebäude an ein State-of-the-Art Gebäude heranführen soll, beschäftigen [27, 4]. Ein besonderes Augenmerk wird dabei auf das Steigern der Energieeffizienz gelegt. In high-tech Gebäuden funken alle Sensoren und Messgeräte an einen Kommunikationsinterface und ein Monitoring-System, welches basierend auf diesen Daten den Energieverbrauch optimiert. Bedingt durch die hohen Kosten, die mit einer Adaption eines existierenden Altbaus auf die neuen Standards einhergehen, liegt der Fokus auf nichtinvasiven und somit kostengünstiger Mess- und Analysetechnik. Nichtinvasive Sensoren erlauben eine mühelose Installation im Gebäude und eine einfache Integration in ein Monitoring-System. In diesem Kontext werden Temperatursensoren und Thermostate, welche die Ventilstellung aufzeichnen, verwendet.

Neuerdings werden mehrere Paper über Ansätze, den Energieverbrauch eines Gebäudes zu optimieren, veröffentlicht [1, 31, 34]. Als konkretes Beispiel möchten wir hier die Entwicklungen der Forscher der *University of Southern California* in [24] vorstellen. Diese stellen zwei verschiedene Labore mit einem Sensor aus, welcher die Luftfeuchtigkeit, den Lärm, den Kohlenstoffdioxidgehalt in der Luft, den Türzustand (offen, geschlossen) aufzeichnet und mit einem Bewegungssensor Bewegungen wahrnimmt. Mit diesen minütlich ausgelesenen Daten wird eine Support Vector Machine SVM trainiert, welche die Anzahl der Personen im Raum angibt. Auf Basis dieser Belegung wird das Heating, Ventilation and Air Conditioning HVAC-System geregelt und optimiert. Die Wissenschaftler stützen sich hierbei auf den Zusammenhang zwischen dem Bewegungsfluss von Personen und dem Wärmefluss in einem Gebäude, welcher in [26] beschrieben ist. Bei diesem so wie bei der Mehrheit der Optimierungsansätzen müssen die Gebäude mit modernen HVAC-Systemen ausgestattet sein. Eine Vielzahl der Bestandsgebäude ist daher von diesen Verbesserungen

ausgeschlossen. Einen gänzlich anderen Ansatz, den Energieverbrauch eines Gebäudes zu senken beziehungsweise zu optimieren, verfolgen die Entwickler in [7]. Sie wählen eine SVM um den Energieverbrauch eines HVAC-Systems vorauszusagen. Zum Energiesparen können sie nun die Bausubstanz an bestimmten Stellen erneuern.

Gestützt auf die oben beschriebenen Tatsachen liegt im Folgenden der Fokus auf einem Energiemonitoring-System, das mit kostengünstiger Sensorik den besten Output liefert. Im Gegensatz zu den Optimierungsalgorithmen des letzten Absatzes wollen wir im ersten Schritt die thermische Energie, ähnlich dem in Sektion 2.3 vorgestellten Vorgängen, auf die einzelnen Komponenten des Heizungssystems aufteilen. Weiters filtern wir auf Basis der Messwerte fehlerhafte Komponenten aus dem System heraus.

## 2.2 Monitoring-System für Heizungssysteme - State of the Art

Aktuell sind laut Literaturrecherchen keine Berechnungsmethoden für Heizungssysteme verfügbar, die aus einem Summenverbrauch verlässlich den jeweiligen Einzelverbrauch aufschlüsseln.

Gespräche mit Bautechnikern und anderen Experten auf diesem Gebiet haben ergeben, dass das entsprechende Erweitern der Anlage und der Verbau neuer Wärmezähler die am häufigsten verwendete Lösung zum Bestimmen des Einzelverbrauchs darstellt. Die zuvor erwähnten hohen Kosten entstehen, da neben der Beschaffung benötigter Hardware auch noch Modifikationen an der bestehenden Heizungsanlage (Aufschneiden der Rohre) durchgeführt werden müssen.

Eine andere Möglichkeit zur Bestimmung des Einzelverbrauchs, die manchmal bei neuen Gebäuden angewendet wird, ist die genaue Vermessung von einzelnen Gebäudeteilen, die dann als Referenz herangezogen werden. Konkret stattet man wenige Räume oder Wohnungen mit relativ viel Messtechnik aus, um den thermischen Energieverbrauch genau zu bestimmen. Diese Werte werden dann anhand eines vorab berechneten Verteilungsschlüssels auf andere Gebäudeteile angewendet, wodurch man eine grobe Schätzung des Energieverbrauchs erhält. Unterschiedliche Gegebenheiten, wie beispielsweise die Lage des Raumes oder die Anzahl an anwesenden Personen, verfälschen diese Ergebnisse. Da diese Berechnungsmethode keine Möglichkeit bietet, sich auf geänderte Verhältnisse einzustellen, kann atypisches Verhalten, das von einer defekten Isolierung oder Ähnlichem verursacht wird, nicht herausgefiltert werden.

Mit Hilfe einer Simulation können Forscher die Wärmeflüsse in einem Gebäude sichtbar machen. Für ein italienisches Museum wurde dies gemacht. Eine Forschungsgruppe rund um Fabrizio Ascione beschreibt in [3] die Arbeitsschritte beim Entwickeln dieser Simulation. Anschließend haben die Verwalter des Museums die Möglichkeit, den Energieverbrauch einzelner Gebäudeteile und kleinerer Einheiten genau zu analysieren. Hierbei wird nicht der reale Energieverbrauch betrachtet. Die Simulation liefert den Vorteil, dass Veränderungen an der Baumasse, die Energie sparen sollen, zuvor am Computer getestet werden können. Ein weiteres Paper, welches ein Gebäude vermisst und so den Energieverbrauch bestimmt, wurde von den Autoren Athanasios Tsanas und Angeliki Xifara [38] veröffentlicht.

### 2.3 Non-Intrusive Load Monitoring NILM - State of the Art

Es existieren bereits viele Forschungsarbeiten ([25, 22, 45, 46, 44, 33]), die sich des Themas Energiemonitoring annehmen. Die Wissenschaftler beschränken ihre Betrachtungen jedoch meist auf elektrische Anwendungen. Da wir auf diese Arbeiten aufbauen, wollen wir hier eine kurze Einführung in dieses Themengebiet liefern.

#### 2.3.1 Ein Überblick

Das zugrundeliegende Konzept geht auf ein Paper von Hart [13] zurück. Hier wird eine Methode vorgestellt, die nur anhand der jeweiligen Leistungssignaturen verschiedener Verbraucher den Gesamtenergieverbrauch aufschlüsselt. Die Besonderheit liegt darin, dass auf zusätzliche Sensoren verzichtet wird, was durch den Zusatz *non-intrusive* verdeutlicht wird. Die mathematische Formulierung können wir anschreiben als

$$E(t) = e_1(t) + e_2(t) + e_3(t) + \dots + e_n(t), \quad (2.1)$$

wobei mit  $e_i(t)$  die benötigte Energie eines Verbrauchers  $i$  bis zum Zeitpunkt  $t$  bezeichnet wird.  $E$  schreiben wir für den Gesamtbedarf an Energie, gemessen am Energiezähler. Auf dem Weg zur genauen Aufschlüsselung starten wir mit einer Suche nach Schaltevents. Diese Events zeigen uns eine Zustandsänderung, welche zum Beispiel dem Einschalten eines Fernsehers entspricht, an. Sobald uns diese entscheidenden Zeitpunkte bekannt sind, können wir mit dem geforderten Load Monitoring beginnen. Hierfür benützen wir Algorithmen aus dem Umfeld des Maschinellen Lernens. Wichtig bei dem System ist die Trainingsphase, diese stellt in der späteren Anwendung sicher, dass die Fehlerquote sehr niedrig ist. Wir können im Groben zwei Möglichkeiten wählen:

1. überwachtes Lernen [14]: Die Leistungssignaturen aller Verbraucher sind uns schon zuvor bekannt. Wir füttern den Algorithmus mit dieser zusätzlichen Information und ermöglichen ihm, mit bekannten Daten zu trainieren. Hier sind wir im Anwendungsfall schnell an den Grenzen angekommen, da nicht alle Signaturen bekannt sein müssen. Messgeräte bei allen Stromabnehmern zu installieren und die Signaturen so in der Trainingsphase aufzuzeichnen, widerspricht dem *non-intrusive* Ansatz und erhöht die Kosten. Daher wenden wir uns der zweiten Möglichkeit zu.
2. unbeaufsichtigtes/freies Lernen [46]: Dieses Training funktioniert ohne jegliche a-priori Informationen. Somit ist diese Variante die kostengünstigere und ermöglicht mehrere verschiedene Verwendungen.

Für Anwender stellt sich noch die Frage, wie lange auf die Verarbeitung gewartet werden muss. In Fachkreisen benützt man hierfür die Phrasen *online* oder *offline* Verarbeitung der Daten. Vereinfacht können wir den Unterschied wie folgt formulieren: Wir erhalten am Ende einer vordefinierten Zeitspanne die aufbereiteten Messwerte (*offline*) oder wir können diese nach einer kurzen Trainingsphase jederzeit abrufen (*online*). Untersuchungen haben gezeigt, dass Energiesparpläne nur dann das gewünschte Ziel erreichen, wenn uns Menschen in Echtzeit der Verbrauch unserer Hilfsmittel vor Augen geführt wird. Der Fokus liegt daher auf Algorithmen, welche auch online funktionieren.

### 2.3.2 Elektrische Energie

Für die genaue Aufschlüsselung der verbrauchten elektrischen Energie einzelner Haushaltsgeräte existieren Monitoring-Systeme. Der Autor in [13] schlägt vor, elektrische Geräte in verschiedene Kategorien, abhängig von ihrer Verwendung, zu gliedern:

Typ-I: Hier sind jene Geräte zusammengefasst, die sich nur in zwei Stadien (ON/OFF) befinden können. Beispiele sind Lampen, Kühlschrank und andere.

Typ-II: Diese Kategorie umfasst alle Maschinen mit einer endlichen Anzahl an Zuständen (Finite States Machines FSM). Waschmaschinen und Computer haben Schaltmuster, die sich wiederholen. Sie sind daher leichter für den Algorithmus zu identifizieren.

Typ-III: Leuchten mit Dimmerfunktion sind die bekanntesten Vertreter dieser dritten Kategorie. Sie haben keine endliche Anzahl an Stadien (Continuously Variable Devices CVD) und damit ein weites Spektrum an Leistungscharakteristiken. Dadurch zählen sie zu den größten Herausforderungen unserer Energieaufschlüsselung.

Typ-IV: In [44] wird noch eine zusätzliche Kategorie vorgestellt, welche Geräte mit einer sehr langen Laufzeit umfasst. Exemplarisch für jegliche Verbraucher dieser Klasse werden hier Rauchmelder angegeben.

Nun folgen wir dem in Kapitel 2.3.1 erklärten Zugang. Hierfür müssen die Stadien, wie eingeschaltet, ausgeschaltet oder standby, automatisch in verschiedene Cluster separiert werden. Die Events, an denen es zu Veränderungen der Stadien kommt, können naiver Weise bei größeren Energiesprüngen angenommen werden. Jedoch gibt es auch professionellere Wege, den Zeitpunkt dieser Zustandsänderungen zu identifizieren. Beschrieben sind Beispiele in [22]. Die Daten sind nun soweit aufbereitet, dass die eigentliche Identifikation der Verbraucher beginnen kann. Für diesen Vorgang zeigt sich, dass Hidden Markov Models HMM, wie in [45] vorgestellt, und Artificial Neural Networks ANN [35] am besten geeignet sind. Aber auch mit Support Vector Machines SVM [46] und k-Nearest-Neighbour KNN Algorithmen [41] wurden Erfolge, hauptsächlich beim Clustern, erreicht. Mit Naiven Bayes Classifier kann der Autor in [25] ebenso den individuellen Verbrauch bestimmen. Unter dem Überbegriff *Optimization* sind Ansätze zusammengefasst, welche das Problem  $class = \arg \min_i \|\hat{y}_i - y_i\|$ , wobei  $\hat{y}_i$  eine in einer Datenbank gespeicherte Leistungssignatur und  $y_i$  die Signatur eines unbekanntes Verbrauchers ist, lösen [37, 21]. Für einen besseren Überblick über die beschriebenen Herangehensweisen sei auf Tabelle 2.1 verweisen.

Tabelle 2.1: Vergleich der Algorithmen [46]

Algorithmus	Genauigkeit (%)	Training (Überwacht/ Frei/Beides)	Online/Offline/ Beides	Typ
SVM	75-98	Ü	Online	I, II, III, IV
Bayes	80-99	Ü	B	I,II
HMM	75-95	B	Offline	I,II
ANN	80-97	Ü	Online	I,II,III
KNN	70-90	Ü	B	I,II
Optimization	60-97	Ü	B	I,II

Da viele Algorithmen zum Trainieren markierte Daten beziehungsweise Datensets mit a-priori Information neben der Summenleistungskurve benötigen, liegt aktuell ein Fokus vieler anderer Arbeiten ([33, 46]) dem Erzeugen von einheitlichen Datenbanken mit Leistungssignaturen für Einzelgeräte. Die einfachste Möglichkeit, solche Datenbanken zu generieren, besteht darin ein Gerät nach dem anderen ein- und auszuschalten. Um diesen zeit- und arbeitsintensiven Weg zu umgehen, entwickeln Forscher Systeme wie ANNOT [33]. Die Autoren verwenden hierbei verschiedene kostengünstige und einfache Sensoren, um die elektrischen Daten dem jeweiligen Erzeuger zuzuordnen und für ein NILM vorzubereiten. Nach

der Integration des ANNOT Systems in bereits existenten Algorithmen wird eine Überwachung von außen während der Lernphase hinfällig. Andere Ansätze, die eine Korrektheit der Leistungssignaturen garantieren sollen, erstrecken sich von akustischen Sensoren [12] über eine festgelegte Reihenfolge bei der Verwendung bestimmter Gerätegruppen (zum Beispiel im Arbeitszimmer: Licht an - Computer an - Computer aus - Licht aus) [42] bis zur Einbindung des Heizungssystems, um Rückschlüsse auf die Anwesenheit von Bewohnern zu ziehen [9]. Eine unverzichtbare Anforderung an Datenbanken, die Leistungssignaturen enthalten, ist die Vollständigkeit. Falls Signaturen von einzelnen Geräten nicht verfügbar sind, können diese im Laufe der Aufschlüsselung der Energien nicht bestimmt werden.

Ein weiterer wichtiger Faktor für zukünftige Entwicklungen liegt in der Vielseitigkeit der Algorithmen. Wie in Tabelle 2.1 verdeutlicht, decken die beschriebenen Varianten des Maschinellen Lernens immer nur einen Teil der vier Typen von Haushaltsgeräten ab. Die übrigen Klassen können nicht exakt oder gar nicht bestimmt werden. Hierbei scheitert es in den meisten Fällen am Erkennen des Zustandes, in dem sich unser Verbraucher befindet. Im Moment haben wir wenig Chancen, die bereits entwickelten Verfahren miteinander in fairen Tests zu vergleichen, eine Verbreiterung des Spektrums an erkennbaren Klassen würde sich jedoch positiv auf die Vergleichbarkeit auswirken. Für den kommerziellen Gebrauch eignen sich eher jene Algorithmen, welche die Daten online verarbeiten.

Zusammenfassend liegt der Fokus auf einem unbeaufsichtigt lernenden Algorithmus, der alle vier Typen von Geräteklassen abdeckt und die Daten online aufschlüsselt.

## 3 Simulationsmodell

Wie schon erwähnt, müssen wir ein Gebäude mit all seinen Wärmeflüssen simulieren, um anschließend die Load-Monitoring Algorithmen auf die generierten Daten anwenden zu können. Wir betrachten hier ein Gebäude mit drei Stockwerken und dreizehn Zimmern pro Geschoss, wobei die Anordnung in Abbildung 3.1 dargestellt ist. Dieser Grundriss hat keine identischen Räume, somit können wir anhand des Stockwerks und der Anzahl und Ausrichtung der Außenwände jedes Zimmer eindeutig identifizieren und benennen. Das Haus hat keinen Keller und es ist mit einem Flachdach ausgestattet. In den Grundzügen orientiert sich das Simulationsmodell an jenem, welches in [36] vorgestellt ist.

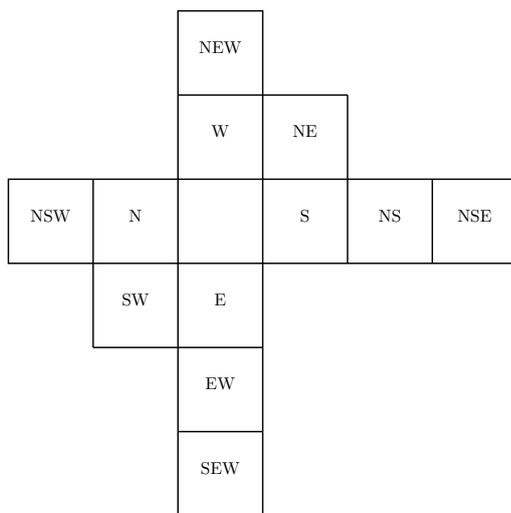


Abbildung 3.1: Schematische Raumanordnung eines Stockwerks

### 3.1 Modellierung der Wärmeflüsse

Die Abbildung 3.2 gibt eine erste Übersicht über die Wärmeflüsse, welche wir in weiterer Folge simulieren müssen. Jeder Raum verfügt über einen Heizkörper als Energiequelle und

verliert diese Wärme wieder über die Wände. Diese thermischen Vorgänge formalisieren wir in den folgenden Unterkapiteln. Die rot markierten Verbindungen leiten Leistung weiter, die blau markierten Temperatur.

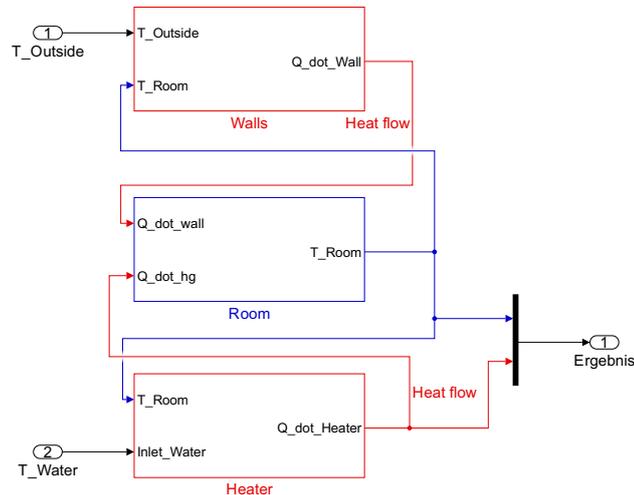


Abbildung 3.2: Wärmeflüsse im Modell

### 3.1.1 Heizkörper

Das Herz unseres Systems ist ein Heizkörper, der als Wärmequelle fungiert. In diesen Wärmetauscher fließt Wasser mit einer bestimmten Temperatur und einem Massenstrom ( $kg/s$ ). Die so eingebrachte Energie wird in weiterer Folge dem Wasser entzogen, im Heizkörper gespeichert und an die Umgebungsluft wieder abgegeben. Die folgenden Gleichungen leiten sich in den meisten Fällen von der Wärmeleitungsgleichung, auf dessen Herleitung in [8] eingegangen wird,

$$\dot{Q}_{1 \rightarrow 2} = k \cdot A \cdot \Delta T \quad (3.1)$$

ab. Dieses Gesetz formuliert den natürlichen Prozess des Wärmeflusses von Medium 1 in Medium 2. Die Stärke dieses Flusses hängt hauptsächlich von der Temperaturdifferenz  $\Delta T$  der beiden Medien ab.  $k$  bezeichnet von nun an den Wärmedurchgangskoeffizienten, welchen wir später mit einem Index versehen, um klarzustellen, welches Medium gemeint ist.  $A$  steht für den Flächeninhalt der am Austausch beteiligten Fläche.

Der Energieerhaltungssatz gewährleistet uns, dass im System keine Energie verloren geht. Zusätzlich gilt der bekannte physikalische Zusammenhang, welcher besagt, dass sich die Energie als Integral der Leistung über die Zeit berechnen lässt. Wenn  $\dot{Q}_H$  der aktuellen Leistung entspricht, die über warmes Wasser zufließt,  $\dot{Q}_{in}$ , abzüglich der, die wiederum mit dem kälteren Wasser abfließt,  $\dot{Q}_{out}$ , beziehungsweise in den Raum abgegeben wird,  $\dot{Q}_{hg}$ , können wir dies durch

$$\dot{Q}_H = \dot{Q}_{in} - \dot{Q}_{out} - \dot{Q}_{hg}$$

formalisieren.

Die Heizleistung  $\dot{Q}_{hg}$  hängt von der Temperatur  $T_{Heater}$  und der Raumtemperatur  $T_{Room}$  ab. Hierzu gehen wir von einem freien konvektiven Wärmetransport aus und benutzen den Zusammenhang

$$\dot{Q}_{hg} = UA \cdot (T_{Heater} - T_{Room}) \cdot |T_{Heater} - T_{Room}|^{n-1}, \quad (3.2)$$

gegeben in [20], wobei die Konstanten  $UA$  und  $n$  bekannt sind. Das Kürzel  $UA$  steht für das Produkt aus U-Wert und Flächeninhalt der am Austausch beteiligten Fläche. Der Wert  $UA$  beschreibt neben der Bauart (z.B. Rippen- oder Gliederheizkörper) auch das Fertigungsmaterial des Heizkörpers. Dieses können wir aus den Normwerten, die in der DIN EN 442 Norm festgelegt sind, gemeinsam mit einer geforderten Heizleistung von beispielsweise 100 Watt pro Quadratmeter berechnen. Der mit  $n$  bezeichnete Heizkörperexponent hängt vom Modell und von der jeweiligen Bauweise (zum Beispiel Fußbodenheizung, Wandheizung, Radiator) ab. Da  $n \in [1, 2] \subseteq \mathbb{R}$  liegt, müssen wir bei der Implementierung sicherstellen, dass wir unter der Wurzel keine negative Zahl erhalten. Das Vorzeichen ist für den Energiefluss jedoch sehr wichtig, daher spalten wir den Exponenten auf und führen die Rechnung wie oben angedeutet aus.

Die Differenz  $\dot{Q}_{in} - \dot{Q}_{out}$  beschreibt die Aufnahme an Leistung des Heizkörpers aus dem Wasser. Wir gehen hier von der Gleichung

$$\dot{Q}_{in} - \dot{Q}_{out} = \theta \cdot \dot{m} \cdot cp_{Water} \cdot (T_{in} - T_{out}), \quad (3.3)$$

aus, wobei  $cp_{Water}$  die spezifische Wärmekapazität von Wasser,  $\dot{m}$  die Durchflussrate und  $\theta$  die Thermostatstellung, welche im Abschnitt 3.1.2 behandelt wird, bezeichnet.

Da es im Allgemeinen schwierig ist die Temperatur  $T_{out}$  zu bestimmen, werden verschiedene Möglichkeiten betrachtet. Eine Möglichkeit besteht darin, sowohl  $T_{out}$  als auch  $T_{in}$  als konstant anzunehmen. Eine Auswirkung daraus ist, dass immer die gleiche Leistung vom Wasser zum Heizkörper übergeht, unabhängig von dessen Temperatur. Die andere Variante verwendet den Heizkörper als idealen Wärmespeicher. Die Konsequenz ist, dass die Temperatur des ausfließendes Wasser mit jener des Heizkörpers gleichgesetzt werden kann, kurz  $T_{out} = T_{Heater}$ . Auch hier verwenden wir eine konstante Vorlauftemperatur  $T_{in}$ . Die Größe  $T_{Heater}$  bezeichnet die Innentemperatur des Heizkörpers und wir erhalten aus der Heizleistung  $\dot{Q}_H$  und der Wärmekapazität des Wassers gemäßdem Gesetz, welches wir aus Gleichung (3.3) ableiten können

$$\dot{Q}_H = m \cdot cp \cdot \Delta T. \quad (3.4)$$

Wir multiplizieren mit dem Faktor  $1/(m \cdot cp_{Water})$  und integrieren nach der Zeit:

$$T_{Heater} = \int \frac{\dot{Q}_H}{m \cdot cp_{Water}} dt + T_0$$

Die Variable  $m$  steht für die Masse an Wasser im Heizkörper und  $T_0$  für die Anfangstemperatur.

### 3.1.2 Heizkörperthermostat

Die Funktionsweise des Heizkörperthermostats ist schnell erklärt. Durch das Öffnen und Schließen wird die Durchflussrate an Wasser geregelt. Wir können somit in der Gleichung (3.3) den eintreffenden Wärmestrom mit Hilfe von  $\theta$  regeln. In unseren Fall wählen wir  $\theta \in \{0,1\}$ , was bedeutet, dass sich das Thermostat ganz öffnet beziehungsweise schließt, wenn die Raumtemperatur außerhalb der Hysterese um eine Grundtemperatur liegt. Wir verwenden einen 2-Punkt Regler mit einer Hysterese von  $\pm 2^\circ\text{C}$  um  $21^\circ\text{C}$ . Eine andere Möglichkeit ist auch, das Thermostat stufenlos zu gestalten, genauer  $\theta \in [0,1]$ , jedoch ist dies in dieser Anwendung nicht relevant.

### 3.1.3 Wohnraum

Da wir hier ein geschlossenes System haben, kommt wiederum der Energieerhaltungssatz zur Anwendung. Formuliert für diese Problemstellung lautet dieser

$$\dot{Q}_{Room} = \dot{Q}_{hg} - \dot{Q}_{Window} - \dot{Q}_{Wall}. \quad (3.5)$$

Mit anderen Worten, das Gasvolumen  $V$  im Raum wird mit der Heizkörperleistung  $\dot{Q}_{hg}$  erhitzt, aber verliert durch den Wärmestrom durch Wand und Fenster an Wärme. Nun berechnen wir die Raumtemperatur mit einer Anpassung der Gleichung (3.4) für Gase aus [36]

$$\dot{Q}_{Room} = \frac{p \cdot V \cdot cp_{Air}}{R_{Air} T_{Room}} \Delta T_{Room},$$

wobei  $R_{Air}$  die Gaskonstante und  $cp_{Air}$  analog zu oben die spezifische Wärmekapazität von Luft beschreibt. Um daraus die aktuelle Raumtemperatur zu erhalten, müssen wir die Gleichung nach der Zeit integrieren und zuvor umformen,

$$T_{Room} = \int \frac{R_{Air} \cdot T_{Room}}{p \cdot V \cdot cp_{Air}} \cdot \dot{Q}_{Room} dt + T_{Room,0}.$$

### 3.1.4 Wand und Fenster

Den Wärmetransport durch ein ein- oder mehrschichtiges Medium modellieren wir ebenso mit der Formel (3.1). Den Wärmedurchgangskoeffizienten  $k$  wählen wir nun abhängig von Dicke und Art des Materials der einzelnen Schichten. Die Dämmwerte der  $n$  Schichten summieren sich ähnlich zu einer seriellen Schaltung elektrischer Widerstände [19],

$$k_{Wall/Window} = \sum_{i=1}^n \frac{d_i}{\lambda_i},$$

mit  $\lambda_i$  als der Wärmeleitfähigkeit und  $d_i$  für die Dicke der  $i$ -ten Schicht. Es ergibt sich somit ein Wärmefluss durch die Trennwand, der gegeben ist mit

$$\dot{Q}_{Wall/Window} = k_{Wall/Window} \cdot A_{Wall/Window} \cdot (T_1 - T_2).$$

$T_1$  und  $T_2$  bezeichnen die Temperaturen des auf jeder Seite der Trennwand angrenzenden Mediums. Anhand des Vorzeichens von  $\dot{Q}_{Wall/Window}$  erkennen wir die Flussrichtung des Wärmestroms, positives Vorzeichen gibt einen Fluss von  $T_1$  nach  $T_2$  an, negatives einen von  $T_2$  nach  $T_1$ .

## 3.2 Implementierung in Matlab/Simulink

### 3.2.1 Modellaufteilung

Um die Simulation durchzuführen, vernetzen wir die zuvor beschriebenen Wärmeflüsse. Da hier viele verschiedene Komponenten zusammenspielen, überlegen wir uns eine Strategie, welche uns die Übersicht erleichtert. Ein Vorschlag ist die folgende Aufteilung:

1. Wände gemeinsam mit Fenstern bilden ein eigenes Modell, wobei mit Hilfe einer Maske, wie in Abbildung 3.3 gezeigt, die Art, Innenwand, Außenwand, Decke oder Boden basierend auf dem  $k$ -Wert, und die Größe der Wand- und Fensterfläche , spezifiziert werden kann. Als Input fungieren die Temperaturen der angrenzenden Bereiche, welche danach in einen gerichteten Wärmefluss transformiert werden.

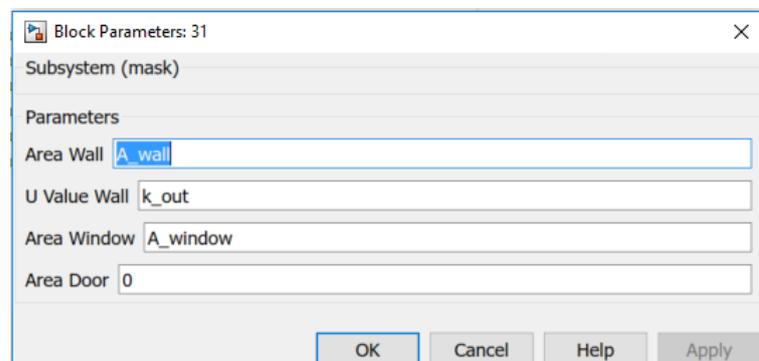


Abbildung 3.3: Eingabemaske für eine Wand

2. Das Modell für einen Wohnraum beinhaltet
  - (i) einen Heizkörper, welchen wir neben der Raumtemperatur zusätzlich mit der Vorlauftemperatur und der Durchflussrate versorgen, dargestellt in Abbildung 3.5; das Thermostat fügen wir hier als Untermodell ein; und

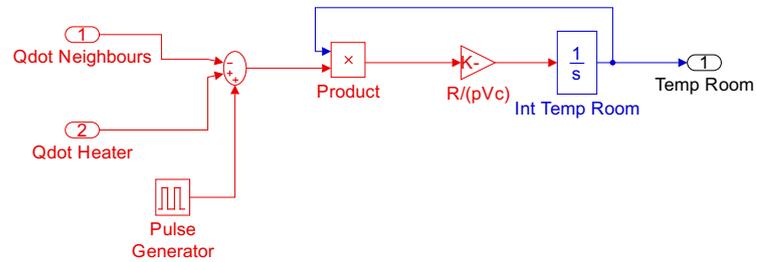


Abbildung 3.4: Modell eines Raumes

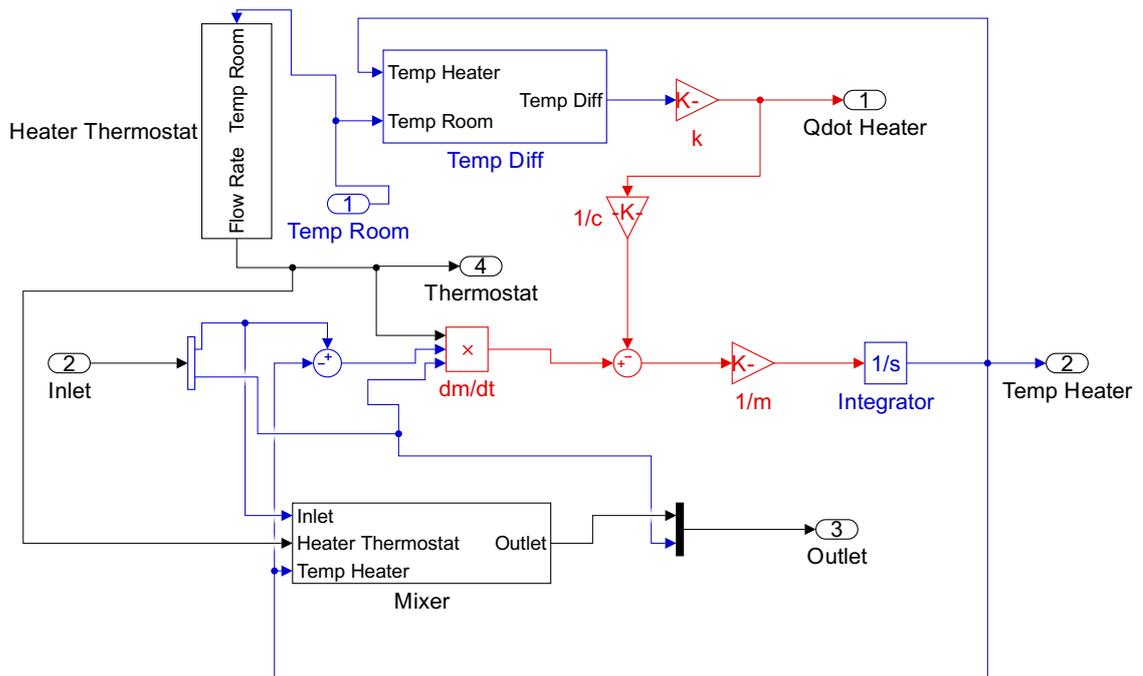


Abbildung 3.5: Modell eines Heizkörpers

- (ii) den Raum selbst. Wie in Gleichung (3.5) und Abbildung 3.4 erkennbar, benötigen wir für die Berechnung der Wärmeflüsse neben der Heizleistung auch die Verluste durch die Wände beziehungsweise den Boden und die Decke, welche als Eingang vom Wand-Modell geliefert werden. Wir betrachten bei unserem Modell ein Bürogebäude, daher addieren wir im Zeitraum von 8:00 bis 18:00 Uhr mit Hilfe eines Pulsgenerators, siehe Abbildung 3.4, eine konstante Leistung von 120 Watt, welcher der durchschnittlichen Wärmeabstrahlung eines Menschen entspricht.

Insgesamt erhalten wir ein Modell mit sechs Eingängen und der Raumtemperatur als Output. Im Speziellen interessiert uns die verbrauchte Wärmeenergie jedes Raumes. Hierfür integrieren wir die Heizleistung über die Zeit und multiplizieren sie mit der Thermostatstellung,

$$E = \int \dot{Q}_{hg} \cdot \theta \, dt.$$

In der Praxis spiegelt sich im Wärmehähler nur die Energie wider, welche bei einem geöffneten Thermostat verbraucht wird, was wir durch diese Simulation sicherstellen. Sobald das Thermostat geschlossen ist, kühlt der Heizkörper nur mehr langsam ab. Der Wärmehähler benötigt für seine Messung einen Durchfluss und bei geschlossenen Ventil ist dieser nicht vorhanden. Für den Fall, dass beim Thermostat eine stufenlose Stellung möglich ist, ersetzen wir  $\theta$  durch  $[\theta]$ .

#### 3.2.2 Konstanten

Zusätzlich müssen wir die diversen Konstanten definieren. Hierzu verwenden wir zur besseren Lesbarkeit eine eigene `.mat`-Datei, die im Anhang als Listing (6.1) beigefügt ist. Auf einige wenige dieser Konstanten möchten wir genauer eingehen:

1. Für die Bauweise der einzelnen Wände orientierten wir uns an den Vorgaben des Fachverbandes für Stein und Keramische Industrie [39]. In den Tabellen 3.1, 3.2, 3.3 und 3.4 sind die verschiedenen Bauarten angeführt. Die Innenwand wählen wir analog zur Außenwand, verzichten aber auf die Dämmung und die Endbeschichtung.

Tabelle 3.1: Außenwandsystem 4: HLZ 25 + WDVS

Bezeichnung Schicht	Dicke	$\lambda$ -Wert
	m	$\frac{W}{mK}$
Kalkgipsputz	0.015	0.7
Ziegel HLZ 25	0.25	0.25
Spachtelung	0.004	0.8
Dämmung	0.1	0.04
Spachtelung/Gewebe	0.003	0.7
Endbeschichtung	0.002	0.7

Tabelle 3.2: STB Decke

Bezeichnung Schicht	Dicke	$\lambda$ -Wert
	m	$\frac{W}{mK}$
Estrich	0.07	1.4
Trennlage	0.0001	0.5
Trittschalldämmung EPS-T	0.03	0.044
Beschüttung gebunden	0.08	0.06
Stahlbetondecke	0.22	2.5

Tabelle 3.3: Erdanliegender Fußboden: Wärmedämmung in FB-Aufbau

Bezeichnung Schicht	Dicke	$\lambda$ -Wert
	m	$\frac{W}{mK}$
Estrich	0.07	1.4
Trennlage	0.001	0.5
Trittschalldämmung EPS-T	0.03	0.044
Wärmedämmung EPS W20	0.1	0.038
Beschüttung gebunden	0.07	0.06
Abdichtung	0.002	0.23
Stahlbetonplatte	0.25	2.5
Sauberkeitsschicht	0.1	1.4

Tabelle 3.4: Warmdach: Stahlbetondecke (Flachdach)

Bezeichnung Schicht	Dicke	$\lambda$ -Wert
	m	$\frac{W}{mK}$
Oberflächenschutz	0.05	0.7
Schutzschicht	0.0002	0.5
Abdichtung	0.0003	0.17
Gefälledämmung EPS W20 im Mittel 30 cm	0.3	0.038
Dampfsperre	0.004	0.17
Stahlbetonplatte	0.25	2.5

2. Wir müssen den Heizkörper für unsere Räume ausreichend dimensionieren. Hierfür beziehen wir uns auf die bereits erwähnte Norm DIN EN 442, welche eine Vorlauftemperatur von 75°C, eine Rücklauftemperatur von 65°C und einer Raumtemperatur von 20°C vorgibt. Laut Erfahrungswerten benötigen wir einen Heizkörper mit einer Normleistung von 100 Watt pro Quadratmeter<sup>1</sup> unter diesen standardisierten Bedingungen. Da wir unser System nur mit einer Temperatur von 55°C/45°C betreiben wollen, reduziert sich die Leistung um einen Faktor. Diesen Faktor bestimmen wir, wie in [19] erklärt, mit folgender Formel:

$$1600 = \dot{Q}_S = \dot{Q}_N \cdot F$$

$$F = \left( \frac{\Delta T_S}{\Delta T_N} \right)^n$$

$$\Delta T = \frac{T_{in} - T_{out}}{\log \frac{T_{in} - T_{Room}}{T_{out} - T_{Room}}}$$

Da die Temperaturdifferenz zwischen Raumtemperatur und Vorlauftemperatur relativ groß ist, berechnen wir  $\Delta T$  mit dem Zusammenhang, der unter dem Fachbegriff Übertemperatur<sup>2</sup> bekannt ist. In unserer weiteren Simulation spiegelt sich die Größe des Heizkörpers im Produkt  $UA$  (3.2) wider. Nun haben wir genug Informationen gesammelt, um dies und die Durchflussrate zu berechnen,

$$UA = \frac{\dot{Q}_S}{(\Delta T_S)^n} = 12.5661 \text{ W/K}$$

$$\dot{m} = \frac{\dot{Q}_S}{(T_{in} - T_{out}) \cdot cp_{Water}} = 0.0160 \text{ kg/s.}$$

3. Eine sehr wichtige Konstante in unserem Modell ist die Außentemperatur. Hier benötigen wir Temperaturwerte im Stundenintervall über den gesamten Simulationszeitraum. Wir benutzen die Temperatur von Basel<sup>3</sup>, da diese Werte im Internet in ausreichender Qualität frei verfügbar sind. Aus einer `.csv`-Datei können wir sie unkompliziert in `Matlab` einlesen und für die weitere Verwendung aufbereiten. Für die Simulation werden die Temperaturwerte dann linear interpoliert.
4. Zur Vereinfachung unserer Simulation wählen wir alle Räume, wie im Grundriss 3.1 angedeutet, quadratisch. Dies hat den entscheidenden Vorteil, dass jede Wand die

---

<sup>1</sup>Quelle: <http://www.heiz-tipp.de/ratgeber-436.html>

<sup>2</sup>Quelle: <http://www.ibo-plan.de/tools/heizmitteluebertemperatur-mittlere-uebertemperatur-online-berechnung.html>

<sup>3</sup>Download unter [https://www.meteoblue.com/de/wetter/archive/export/basel\\_schweiz\\_2661604](https://www.meteoblue.com/de/wetter/archive/export/basel_schweiz_2661604)

gleiche Größe besitzt und wir hierfür nur die Konstante  $A_{wall}$  benötigen. Das Volumen  $V_{physical} = h \cdot l \cdot w$  stimmt aufgrund dieser Annahme auch bei allen Räume überein. Über das Luftvolumen wird in diesem Zusammenhang der Wärmespeicher eines Raumes beschrieben. Um das thermische Speicherpotential der Raumausstattung (Möbel, etc.) zu berücksichtigen, wird das Volumen noch, wie in der Praxis üblich [20], um einen Faktor korrigiert

$$V_{therm} = h \cdot l \cdot w \cdot 3.$$

### 3.3 Ergebnisse und besondere Auffälligkeiten

Wir wählen möglichst viele Variablen des Systems als Output. Hierbei nutzen wir den Vorteil einer Simulation gegenüber der Realität und speichern die berechneten Ergebnisse für Temperatur, Heizleistung, Wärmeenergie und Thermostatstellung jedes einzelnen Raumes. Zusätzlich summieren wir die Wärmeenergien auf, wodurch wir die gewünschten Daten eines Wärmezählers erhalten, die uns in weiterer Folge die Grundlage der Berechnungen liefern. Gemeinsam mit der Außentemperatur ist der Summenzähler über den dreijährigen Simulationszeitraum in Abbildung 3.6 festgehalten.

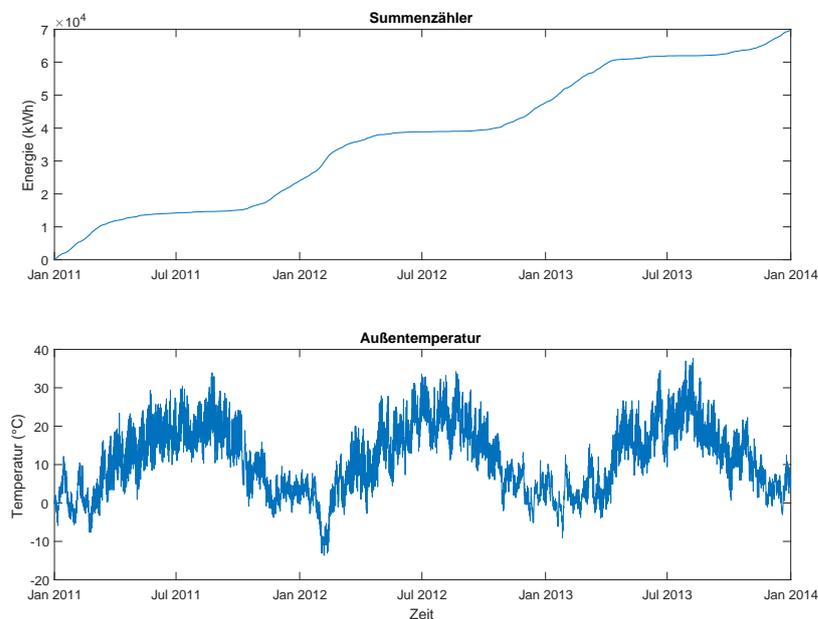


Abbildung 3.6: Plot des Summenzählers und der Außentemperatur

Ein Auszug der Kurven, die wir pro Raum erhalten, ist in Abbildung 3.7 dargestellt. Hier erkennen wir auch mehrere Schaltvorgänge des Thermostats. Über den ganzen Tag verteilt, steigt die Raumtemperatur über  $23^{\circ}\text{C}$ , wodurch eine Schließung des Thermostats bewirkt wird. Sobald die Temperatur unter  $19^{\circ}\text{C}$  gefallen ist, öffnet sich das Thermostat und der Heizkörper kann mit voller Leistung den Raum erwärmen.

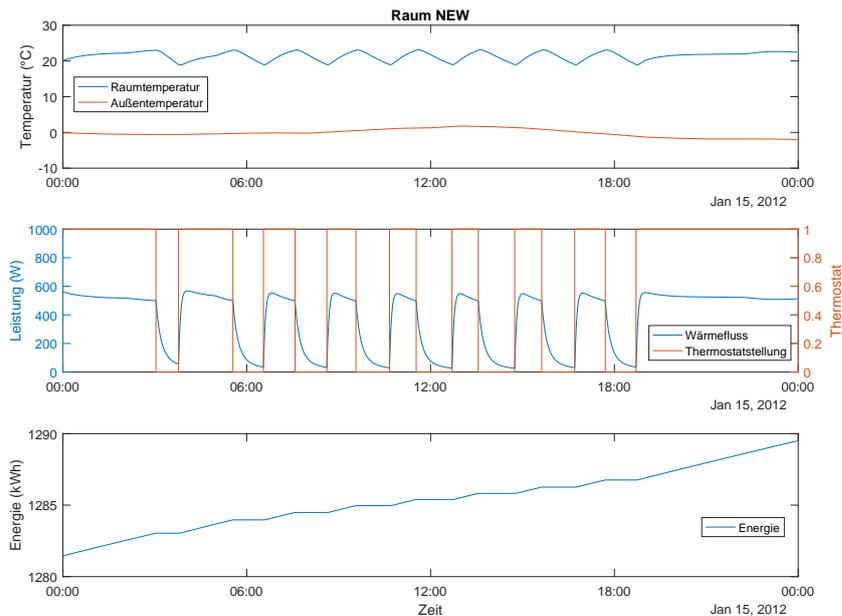


Abbildung 3.7: Simulationsergebnis für einen Tag

Besonders auffällig sind die Plots der Räume genau in der Mitte jedes Stockwerks. In Abbildung 3.8 können wir ablesen, dass dieser Raum über den gesamten Zeitraum nicht geheizt werden muss. Der Wärmefluss durch die vier Innenwände, den Boden und die Decke reicht aus, um die Raumtemperatur über  $19^{\circ}\text{C}$  zu halten. Das Thermostat wird nie geöffnet. Als weitere Konsequenz daraus scheinen diese drei Räume nicht in unserer Summe, dem Graph in Abbildung 3.6, auf. Im Gegensatz zu den übrigen Räumen erhalten wir hier zeitweise eine negative Heizleistung. Diese entsteht, wenn die Temperatur des Heizkörpers unter der Raumtemperatur liegt und somit die Luft den Heizkörper erwärmt.

Wir haben nun ein vollständiges Datenset, welches 39 Räume umfasst, vorliegen. Das Verhalten jedes Raumes in den drei simulierten Jahren beschreiben wir mit vier Zeitreihen, die aus jeweils um die 1.5 Millionen Datenpunkten bestehen. Exemplarisch sind diese Datenreihen anhand von Raum Room0NEW in Abbildung 3.9 dargestellt.

### 3 Simulationsmodell

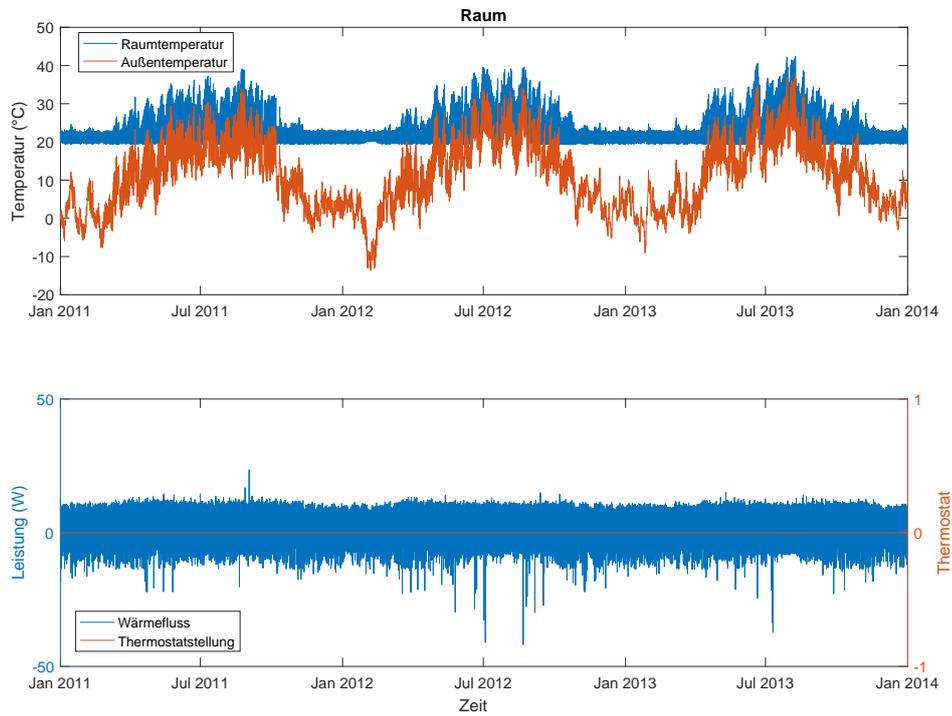


Abbildung 3.8: Ergebnisse des Raumes in der Mitte über drei Jahre

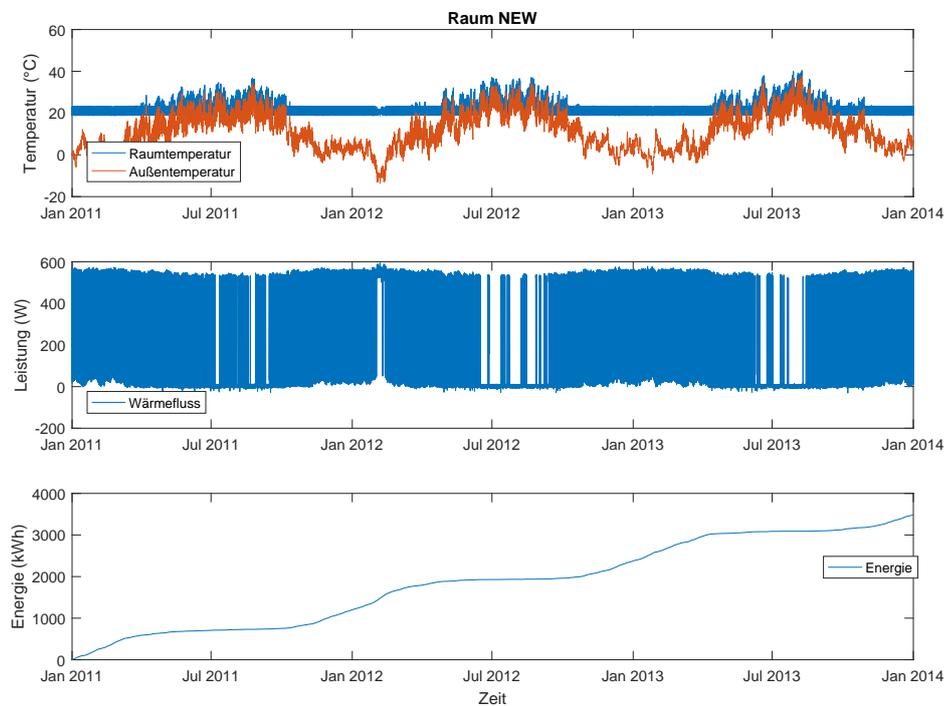


Abbildung 3.9: Ergebnisse des Raumes NEW über drei Jahre

## 4 Energiemonitoring

Das Ziel der Methoden, die wir im Folgenden untersuchen, sind die Bestimmung des Einzelverbrauches in einem Gebäude auf Basis des Energiesummenzählers und Fehlfunktionen einzelner Komponenten im Heizungssystem automatisch zu erkennen. Dieser Zähler liefert gleichzeitig noch den Wert der Gesamtleistung, welche als Ableitung der Gesamtenergie berechnet werden kann. Der Summenverbrauch alleine reicht nicht aus, daher ziehen wir zusätzlich noch die Raumtemperatur, die Thermostatstellung jedes Raumes und die Außentemperatur heran. Diese Informationen können auch durch nachträglich verbaute nichtinvasive Sensorik sehr einfach gemessen werden. Wir stützen unsere weiteren Betrachtungen somit auf den Verlauf von zumindest einer der folgenden vier Kurven: Summenverbrauch, Innentemperatur in jedem Raum, Schalterstellung jedes Raumes zu jedem Zeitpunkt und Außentemperatur.

Im Bereich der elektrischen Energie gibt es sehr viele verschiedene Methoden ([25, 22, 45, 46, 44, 33]), die es erlauben, Verbrauchswerte für diverse elektrische Geräte in einem Haushalt aufzuschlüsseln. Das Kapitel 2.3 beschäftigt sich ausführlich mit diesem Thema. Diese Ideen versuchen wir im Rahmen unseres Anwendungsfalles auf thermische Energie umzulegen. Da im physikalischen Sinn Äquivalenzen zwischen diesen beiden Energieformen vorhanden sind, können wir unsere Vorgehensweise auf bereits funktionierende Ansätze stützen.

### 4.1 Zugang 1: Hidden Markov Modell HMM

#### 4.1.1 Vorbemerkungen

Fassen wir hier nochmal die Anforderungen an unser Monitoring-System im Kontext von Non-Intrusive Load Monitoring zusammen: Jeder Raum ist durch seine Lage und die Wände (spezifiziert durch die Wandparameter) ein Unikat, wodurch wir uns auf keine bereits vorhandenen Leistungssignaturen stützen können. Diese Signaturen selbst auszumessen ist

auch nicht möglich, da wir hierfür zu viele Sensoren verbauen müssten, die wiederum unser Monitoring-System nach dem Einbau hinfällig machen würden. *Supervised* Algorithmen benötigen während der Trainingsphase zusätzliche Informationen, wie diese Leistungssignaturen. Die Ausführungen in Kapitel 2.3 beschreiben, warum wir auf diese Art von Algorithmus verzichten. Ein Blick auf Tabelle 2.1 verrät uns, dass Methoden, die auf einem Hidden Markov Modell basieren, in Frage kommen könnten.

Dieser Zugang liefert uns vor allem wirtschaftliche Vorteile. Der einzige Zähler, der benötigt wird, ist ein thermischer Energiezähler, dieser ist in den meisten Fällen bereits vorhanden. Ergänzende Hardware wie beispielsweise Thermostate, die ihre Schaltung aufzeichnen, entfallen ebenso wie die Kosten der Installation.

### 4.1.2 Algorithmus

#### Einführung

Bei einem Hidden Markov Model HMM handelt es sich um einen stochastischen Prozess mit zwei Ebenen. Jede Ebene beschreibt ein eigenständiges Verfahren, wobei eines der beiden nicht beobachtbar (*hidden*) ist [30]. Dieser versteckte Vorgang kann nur durch den zweiten stochastischen Prozess, welcher eine Serie von Outputsymbolen erzeugt, untersucht werden. Um dieses Konzept leichter verständlich zu gestalten, wenden wir uns einem Beispiel eines Hidden Markov Prozess zu.

#### Beispiel 4.1. Münzwurf [30]

*Um den versteckten Prozess zu simulieren, stellen wir uns vor, wir sitzen in einem Raum mit einem blickdichten Vorhang. Hinter diesem Vorhang hält sich eine zweite Person auf, welche eine Münze wirft. Die für uns nicht sichtbare Person gibt uns nur die Ergebnisse seiner Würfe bekannt, klärt uns jedoch nicht auf, mit welchem Experiment er auf diese Beobachtungen kommt. Beim Output könnte es sich um eine Sequenz der Art*

$$O = \mathcal{K}, \mathcal{K}, \mathcal{K}, \mathcal{Z}, \mathcal{K}, \mathcal{Z}, \mathcal{Z}, \mathcal{K}, \dots, \tag{4.1}$$

*wobei  $\mathcal{K}$  für Kopf und  $\mathcal{Z}$  für Zahl steht, handeln.*

Wie das Beispiel zu illustrieren versucht, besteht ein HMM aus drei Elementen [30]:

- (i) Es liegt eine endliche Anzahl von Zuständen vor, in denen sich unser Modell befinden kann.

- (ii) Das Modell wechselt zu jedem Zeitpunkt in einen neuen Zustand, welcher durch eine Übergangsverteilung  $A$  abhängig vom vorherigen Zustand, festgelegt ist.
- (iii) Nach jedem Zustandswechsel wird ein Output generiert, welcher nur vom aktuellen Zustand abhängt. Diese Observations sind nach der Observationsverteilung  $B$  verteilt. Diese ändert sich zu keinem Zeitpunkt des Prozesses.

Kehren wir noch einmal zurück zu dem Beispiel des Münzwurfs. Bei dem versteckten Prozess kann es sich um verschiedene Experimente handeln. Im einfachsten Fall wirft die Person hinter dem Vorhang eine einzige Münze. Wir wissen, dass die Wahrscheinlichkeit für Kopf oder Zahl jeweils bei 50% liegt. Der Prozess wäre somit leicht beschrieben und damit bekannt. Jedoch kann die zweite Person nicht nur mit einer Münze, sondern mit zwei, drei, vier oder noch mehr Münzen werfen, diese Ergebnisse dann anschließend nach einer gewissen Verteilung kombinieren und uns nur eine Abfolge der Form (4.1) weitergeben. Da wir keine Information über diesen von unserem Standpunkt nicht einsehbaren Prozess haben, reicht das Wissen nicht aus, direkt ein HMM aufzustellen. Übereinstimmend mit den zuvor beschriebenen Elementen muss ein HMM die folgenden drei Probleme lösen, welche die Autoren in [30] formuliert haben:

Problem 1 Wir kennen die Sequenz der Beobachtungen  $O = O_1, O_2, O_3, \dots, O_T$  und die Parameter des Modells  $\lambda = (A, B, \pi)$ . Wie können wir die Wahrscheinlichkeit  $P(O|\lambda)$  der Beobachtungssequenz berechnen?

Problem 2 Wir haben  $O$  gegeben. Wie wählen wir eine in einem gewissen Sinne optimale Sequenz von Zuständen  $I = i_1, i_2, i_3, \dots, i_T$ ?

Problem 3 Wie können wir die Modellparameter  $\lambda = (A, B, \pi)$  adjustieren, um  $P(O|\lambda)$  zu maximieren?

Zum Lösen dieser Probleme gibt es viele verschiedene Ansätze und Algorithmen, wovon wir einen ausgewählten im Folgenden Unterkapitel genauer betrachten. Um den Bogen zurück zum Gebiet des Maschinellen Lernens zu spannen, nennen wir den Prozess, der diese drei Problemstellungen behandelt, Training eines HMM. Sobald wir eine Antwort auf jede dieser drei Fragen gefunden haben, können wir mit Hilfe des aufgestellten HMM anhand einer neuen Beobachtung den versteckten Zustand ermitteln.

### Factorial Hidden Markov Modell FHMM

In der Praxis zeigt sich, dass ein trainiertes HMM oft zu wenig ist, um komplexe Zusammenhänge zu beschreiben. Hier kann die Theorie zu sogenannten Factorial Hidden Markov Modellen erweitert werden. An Stelle eines einzigen HMM werden nun mehrere Modelle parallel entwickelt. Der beobachtete Output setzt sich aus einer Kombination aller versteckten Zustände zusammen [11]. Die Autoren in [16] betrachten einen Spezialfall, bei welchem jedes HMM eine reelle Zahl als Output produziert, und die Summe dieser Outputs beobachtet wird. Wir beschreiben im Folgenden den AFAMAP Algorithmus der Entwickler Kolter und Jaakkola, der alle drei zuvor gestellten Probleme löst. Beim Energiemonitoring für elektrische Anwendungen erzielt diese Variante sehr gute Ergebnisse, daher entscheiden wir uns, diese Methode für die Aufschlüsselung der thermischen Energie anzuwenden.

Das zugrundeliegende FHMM lautet

$$\begin{aligned} x_1^{(i)} &\sim \text{Mult} \left( \phi^{(i)} \right) \\ x_t^{(i)} | x_{t-1}^{(i)} &\sim \text{Mult} \left( P_{x_{t-1}^{(i)}}^{(i)} \right) \\ \bar{y}_t | x_t^{(1:N)} &\sim \mathcal{N} \left( \sum_{i=1}^N \mu_{x_t^{(i)}}^{(i)}, \Sigma \right), \end{aligned} \tag{4.2}$$

wobei  $N$  für die Anzahl an HMM,  $m_i$  für die Anzahl an Zuständen des  $i$ -ten HMM und  $T$  für die Anzahl an Zeitschritten steht. Des Weiteren bezeichnet  $x_t^{(i)} \in \{1, \dots, m_i\}$  den Zustand des  $i$ -ten HMM zum Zeitpunkt  $t$  und  $\bar{y}_t \in \mathbb{R}^n$  den beobachteten zusammengefassten Output. Für die modellbeschreibenden statistischen Größen wählen wir analog die Bezeichnungen  $\mu_j^{(i)} \in \mathbb{R}^n$  für den Erwartungswert des  $i$ -ten HMM für den Zustand  $j$ ,  $\Sigma \in \mathbb{R}^{n \times n}$  für die Varianz der Beobachtungen,  $\phi^{(i)} \in [0, 1]^{m_i}$  für die initiale Zustandsverteilung und  $P^{(i)} \in [0, 1]^{m_i \times m_i}$  für die Übergangsmatrix des  $i$ -ten HMM.

Durch die enthaltene Normalverteilung kann unser Modell Ausreißer, die beispielsweise durch selten beheizte Räume im thermischen Zusammenhang oder durch nicht kategorisierte oder modellierte Geräte im elektrischen Fall entstehen, nicht gut kontrollieren. Wir müssen

zum Modell (4.2) einen Parameter, der die Normalverteilung robuster macht, hinzufügen:

$$p(z_{1:T}) = \frac{1}{Z(\lambda, T)} \exp\left(-\lambda \sum_{t=1}^{T-1} \|z_{t+1} - z_t\|_1\right) \quad (4.3)$$

$$\bar{y}|x_t^{(1:N)}, z_t \sim \mathcal{N}\left(\sum_{i=1}^N \mu_{x_t^{(i)}}^{(i)} + \Sigma^{1/2} z_t, \Sigma\right). \quad (4.4)$$

Die Voraussetzung  $p(z_{1:T})$  entsteht durch eine Regularisierung des Outputs, hier verwenden wir die Regularisierung der Totalen Variation, welche in [23] vorgestellt ist.  $Z(\lambda, T)$  beschreibt eine Größe, die durch die Regularisierung gegeben ist. Für weitere Details möchten wir auf [32] verweisen.

Mit dem Aufstellen des FHMM ist erst ein Teil des Algorithmus abgehandelt. Als weitaus arbeitsintensiverer Prozess gestaltet sich das optimale Kombinieren der einzelnen Modelle. Wir werden im Bereich der Bayes-Schätzer fündig und implementieren ein Schätzverfahren nach der Maximum-a-posteriori-Methode (MAP) [10]. Für die Berechnung dieses Schätzers haben wir verschiedene Möglichkeiten. Eine Variante, vorgestellt in [16], optimiert die Indikatorvariablen

$$\mathcal{Q} = \left\{ Q\left(x_t^{(i)}\right) \in \mathbb{R}^{m_i}, Q\left(x_{t-1}^{(i)}, x_t^{(i)}\right) \in \mathbb{R}^{m_i \times m_i} \right\},$$

wobei  $Q\left(x_t^{(i)}\right)_j = 1 \Leftrightarrow x_t^{(i)} = j$  und  $Q\left(x_{t-1}^{(i)}, x_t^{(i)}\right)_{j,k} = 1 \Leftrightarrow x_{t-1}^{(i)} = j \wedge x_t^{(i)} = k$  gilt. Die Menge der Übergangsmatrizen können wir nicht beliebig wählen. Die Elemente  $Q$  dieser Menge müssen als wichtigste Forderung eine lokal zulässige Verteilung repräsentieren. Diese Forderungen können wir mit der Menge  $\mathcal{L}$  anschreiben

$$\mathcal{L} = \left\{ Q : \begin{array}{l} \sum_j Q\left(x_t^{(i)}\right)_j = 1 \\ \sum_k Q\left(x_{t-1}^{(i)}, x_t^{(i)}\right)_{jk} = Q\left(x_{t-1}^{(i)}\right)_j \\ \sum_k Q\left(x_{t-1}^{(i)}, x_t^{(i)}\right)_{kj} = Q\left(x_t^{(i)}\right)_j \\ Q\left(x_{t-1}^{(i)}, x_t^{(i)}\right)_{kj}, Q\left(x_t^{(i)}\right)_j \geq 0 \end{array} \right\}$$

$Q$  muss in einem *local marginal polytope* liegen, welches in [29] gemeinsam mit [40] definiert und ausgeführt ist.

Da das entstehende Optimierungsproblem

$$\min_{Q \in \mathcal{L} \cap \{0,1\}, \Delta z_{2:T}} \frac{1}{2} \sum_t \left\| \Delta \bar{y}_t - \Sigma^{1/2} \Delta z_t - \sum_{i,j,k} \Delta \mu_{kj}^{(i)} Q \left( x_{t-1}^{(i)}, x_t^{(i)} \right)_{jk} \right\|^2 + \sum_{t,i,j,k} Q \left( x_t^{(i)}, x_{t+1}^{(i)} \right)_{jk} \left( -\log P_{kj}^{(i)} \right) + \lambda \sum_t \|\Delta z_t\| \quad (4.5)$$

in der exakten Form sehr rechenintensiv ist, betrachten wir eine weitere Möglichkeit. Wir nehmen zuerst an die Zeit sei in unserem Modell eine kontinuierliche Größe. Unter dieser Voraussetzung ist es offensichtlich, dass nur ein HMM zu fast jedem Zeitpunkt den Zustand wechselt. Da wir in der Numerik nicht mit kontinuierlichen Zahlen rechnen können, müssen wir die Zeitachse diskretisieren. Wir nennen diese Bedingung *one-at-a-time* und wollen auf sie nicht verzichten, wir übertragen sie mit dem folgenden Satz und der Definition ins Diskrete.

**Definition 4.2.** [2] Gegeben sei eine Familie von Zufallsvariablen  $Y = (X_t)_{t \in \mathbb{N}}$ , wobei alle  $X_t$  nur Werte aus dem höchstens Abzählbaren Zustandsraum  $S = \{s_1, s_2, s_3, \dots\}$  annehmen. Dann heißt  $Y$  eine Markovkette genau dann, wenn

$$P(X_{t+1} = s_{j_{t+1}} | X_t = s_{j_t}, X_{t-1} = s_{j_{t-1}}, \dots, X_0 = s_{j_0}) = P(X_{t+1} = s_{j_{t+1}} | X_t = s_{j_t}).$$

**Satz 4.3.** [16]

Für jedes HMM  $i$  sei  $P^{(i)}$  die Diskretisierung der kontinuierlichen Markovkette mit der Inzidenzmatrix  $Q^{(i)} \in \mathbb{R}^{m_i \times m_i}$  und dem Diskretisierungsintervall  $\Delta t \in \mathbb{R}_+$ . Die ganze Zeitspanne bezeichnen wir mit  $t_f$ , somit hat das FHMM insgesamt  $t_f/\Delta t$  Zeitschritte. Sei  $A$  das Event, welches durch einen zeitgleichen Zustandswechsel zweier HMM definiert ist,

$$A = \left\{ \exists t, i \neq j : x_t^{(i)} \neq x_{t+1}^{(i)} \wedge x_t^{(j)} \neq x_{t+1}^{(j)} \right\},$$

dann gilt

$$p(A) \leq \mathcal{O}(\Delta t). \quad (4.6)$$

Dieser Satz liefert eine Aussage über den Zusammenhang zwischen Zeitdiskretisierung und der Wahrscheinlichkeit für sich zeitgleich ändernde Zustände zweier HMM in Ungleichung (4.6). Es gilt, je feiner die Zeitintervalle gewählt sind, desto unwahrscheinlicher sind zeitgleiche Zustandswechsel. Falls die Anzahl an HMM zu hoch wird, muss diese Ungleichung

nicht mehr erhalten bleiben, da wir im Diskreten das Zeitintervall nicht beliebig verkleinern können. Diese Tatsache wird im anschließenden Beweis sichtbar.

Bevor wir den Beweis durchführen können, benötigen wir ein paar Definitionen aus dem Gebiet der Markovketten. Wir nehmen an, das System sei im Zustand  $j$  zu einer Zeit  $t$ , die Dauer  $dt$  dieses Zustandes leiten wir aus einer Exponentialverteilung mit Rate  $-Q_{jj}$  ab. Wir wählen hier die Exponentialverteilung, da diese analog zur Markovkette, siehe Definition 4.2, gedächtnislos ist. Der Übergang zu einem anderen Zustand  $k$  erfolgt dann mit Wahrscheinlichkeit  $Q_{kj}/Q_{jj}$ , formalisiert:

$$dt|x(t) = j \sim \text{Exp}(-Q_{jj})$$

$$x(t + dt) \sim \text{Mult}(q), \quad q_k = \begin{cases} \frac{-Q_{kj}}{Q_{jj}}, & k \neq j \\ 0, & k = j \end{cases}.$$

Für ein Zeitintervall lautet die diskretisierte Übergangsmatrix

$$P(\Delta t)_{kj} = p(x(t + \Delta t) = k | x(t) = j).$$

Um diese zu berechnen, nutzen wir die Matrixexponentialfunktion

$$\exp(X) = \sum_{i=0}^{\infty} \frac{1}{i!} X^i$$

und erhalten

$$P(\Delta t) = \exp(\Delta t Q).$$

Diese Zusammenhänge nutzen wir nun im Beweis.

*Beweis.* Wir bemerken, dass für  $r = -\min_j Q_{jj}$

$$\min_j P(\Delta t)_{jj} \geq 1 - r\Delta t$$

gilt. Dies leitet sich aus der Matrixungleichung  $\exp(X) \geq I + X$  und einer Eigenschaft der Matrixexponentialfunktion für diagonalisierbare Matrizen - die Exponentialfunktion wird nach einer Transformation nur auf die Einträge in der Diagonale angewandt - ab. Für jeden

dieser Diagonaleinträge hält die Ungleichung  $\exp(x) \geq 1 + x$ . Ebenso setzen wir voraus, dass  $N$  sehr viel kleiner als  $1/\Delta t$  ist.

Analysieren wir nun einen einzelnen Zeitschritt, an welchem nur eines oder keines der  $N$  HMM den Zustand wechselt. Bezeichnen wir dieses Event als  $A^0$ . Unter Berücksichtigung der oberen Abschätzungen mit  $r = p - \min_{i,j} Q_{jj}^{(i)}$  und mit hinreichend kleinen  $\Delta t$ , so dass  $r\Delta t < 1$ , gilt:

$$\begin{aligned} P(A^0) &\geq (1 - r\Delta t)^N + Nr\Delta t (1 - r\Delta t)^{N-1} \\ &\geq (1 - r\Delta t)^N + Nr\Delta t (1 - r\Delta t)^N \\ &= (1 + Nr\Delta t) (1 - r\Delta t)^N. \end{aligned}$$

Daraus können wir ableiten, dass die Bedingung, ein oder kein HMM pro Zeiteinheit seinen Zustand wechselt, für alle  $t_f/\Delta t$  Zeitpunkte mit einer Wahrscheinlichkeit von  $P(\neg A) \geq P(A^0)^{t_f/\Delta t}$  gehalten wird. Als letztes müssen wir zeigen, dass wir die geforderte Ordnung erfüllen. Hierfür benützen wir die Taylor-Approximation des Logarithmus  $\log(1+x) \approx x - 1/2x^2 + \mathcal{O}(x^3)$ , um die untenstehende Ungleichung zu vereinfachen.

$$\log P(\neg A) \geq \frac{t_f}{\Delta t} (\log(1 + Nr\Delta t) + N \log(1 - r\Delta t)) \quad (4.7)$$

Die rechte Seite können wir nun umformen zu

$$\begin{aligned} \log(1 + Nr\Delta t) + N \log(1 - r\Delta t) &\approx Nr\Delta t - \frac{1}{2}N^2r^2\Delta t^2 - Nr\Delta t - \frac{1}{2}Nr^2\Delta t^2 + \mathcal{O}(\Delta t^3) \\ &= -\frac{1}{2}(N + N^2)r^2\Delta t^2 + \mathcal{O}(\Delta t^3). \end{aligned}$$

Gemeinsam mit (4.7) erhalten wir

$$\log P(\neg A) \geq -\frac{1}{2}t_f(N + N^2)r^2\Delta t + \mathcal{O}(\Delta t^2).$$

Zum Finalisieren fassen wir alle hergeleiteten Ungleichungen und Definitionen zusammen

$$\begin{aligned}
 P(A) &= 1 - P(\neg A) \\
 &= 1 - \exp(\log P(\neg A)) \\
 &\leq -\log P(\neg A) \\
 &\leq \frac{1}{2} t_f (N + N^2) r^2 \Delta t + \mathcal{O}(\Delta t^2) = \mathcal{O}(\Delta t).
 \end{aligned}$$

Somit haben wir die Ungleichung (4.6) gezeigt und den Satz bewiesen. In diesem letzten Beweisschritt wird die Rolle der Anzahl  $N$  an HMM ersichtlich. Falls  $N > 1/\Delta t$  ist, verschwindet der Faktor  $(N + N^2)$  in der Größenordnung  $\mathcal{O}(\Delta t)$ . Die daraus resultierende Abschätzung

$$P(A) \leq \mathcal{O}(N^2)$$

würde nicht zum Ziel führen. □

Mit dem Satz 4.3 betreffend die diskrete *one-at-a-time* Bedingung können wir fordern, dass nur jene Matrizen für unsere Optimierung in Betracht gezogen werden, welche

$$\mathfrak{D} = \left( \mathcal{Q} : \sum_{i,j,k \neq j} Q(x_{t-1}^{(i)}, x_t^{(i)})_{jk} \leq 1 \right)$$

erfüllen. Beschränken wir die Menge der zugelassenen Matrizen nicht, enden wir in dem quadratischen Optimierungsproblem (4.5). Mit dieser kleineren Auswahl können wir jedoch ein lineares Problem formulieren

$$\begin{aligned}
 \min_{\mathcal{Q} \in \mathcal{L} \cap \mathfrak{D} \cap \{0,1\}} & \frac{1}{2} \sum_{t,i,j,k \neq j} \left\| \Delta \bar{y}_t - \Delta \mu_{kj}^{(i)} \right\|^2 Q(x_{t-1}^{(i)}, x_t^{(i)})_{jk} \\
 & + \frac{1}{2} \sum_t D(\Sigma^{-1/2} \Delta \bar{y}_t, \lambda) \left( 1 - \sum_{i,j} Q(x_{t-1}^{(i)}, x_t^{(i)})_{jj} \right) \\
 & + \sum_{t,i,j,k} Q(x_t^{(i)}, x_{t+1}^{(i)})_{jk} \left( -\log P_{kj}^{(i)} \right),
 \end{aligned}$$

wobei  $D$  die *Huber-loss*-Funktion

$$D(y, \lambda) = \min_z \left\{ \|y - z\|_2^2 + \lambda \|z\|_1 \right\}$$

bezeichnet. Diese Minimierung kann als Optimierung der *log-likelihood*-Funktion des Modells betrachtet werden. Daher leiten sich die gestellten Nebenbedingungen ab. Diese gesammelten Ergebnisse müssen nun in einen Algorithmus verpackt werden. Für weitere Details verweisen wir auf [16].

Dieses Unterkapitel beleuchtet hauptsächlich das Problem 2 der Auflistung in 4.1.2. Wir ermitteln mit Hilfe der Optimierung die Schaltvorgänge in unserem System, woraus sich die Heizzyklen der Heizkörper ableiten. Die Matrix  $Q$  kodiert diese optimale Abfolge von (*hidden*) Zuständen. Das Lösen des ersten Problems, welches die Bewertung einer gegebenen Beobachtungssequenz beinhaltet, liefert in dieser Herangehensweise eine Startmatrix für die Optimierung. Das dritte und letzte Problem adjustiert die Modellparameter, sodass diese zu den Trainingsdaten passen. Wir verfügen über keine Trainingsdaten, da wir einen *unsupervised* Zugang wählten. Daher wird dieses Problem nicht behandelt. Im nächsten Abschnitt widmen wir uns der Implementation der oben ausgeführten Ergebnisse.

### 4.1.3 Implementierung

Diese Methode ist Teil des Software-Projekts NILM-Eval<sup>1</sup>. Der Algorithmus trainiert ohne Beaufsichtigung und benötigt keine a-priori Informationen über die zu identifizierenden Geräte. Als Ziel verfolgt der Algorithmus für jeden Verbraucher (in unserem Kontext entsprechen die Verbraucher den Räumen) ein HMM aufzustellen. Hierfür muss dieser zuvor die Anzahl an Verbrauchern abschätzen und deren Leistungssignaturen aus der Gesamtleistung herausfiltern. Als nächstes zerteilt er die Leistungskurve in kleinere Teile, wobei jeder Teil einem Einschaltzyklus eines Verbrauchers entspricht. Danach trainiert der Algorithmus zu jedem Stück ein HMM und fasst die Stücke, die am wahrscheinlichsten von einem Gerät erzeugt werden, zusammen. Es liegt nun ein FHMM als Ergebnis vor, welches die einzelnen HMM mit Hilfe der zuvor beschriebenen Optimierungs- und Approximierungstechnik AFAMAP verschränkt. Daraus kann der Einzelverbrauch ermittelt werden.

Der Arbeitsweise des Algorithmus ist in 4.1 angedeutet. Wir übergeben diesem ausschließlich die Gesamtleistung `true_power` des Heizungssystems. Im ersten Schritt wird störendes

---

<sup>1</sup>Link: <https://github.com/beckel/nilm-eval>

Rauschen aus den Daten entfernt. Die Gleichung (4.3) formalisiert dies. Um einen Startwert für die im vorherigen Abschnitt beschriebene Optimierung zu finden, leiten wir aus der Übergangswahrscheinlichkeit der Teilstücke eine Adjazenzmatrix, wie in Zeile 12 angedeutet, ab. Nach dem Ablauf des AFMAP-Algorithmus liegt eine zuverlässige Schätzung des Energieverbrauchs für alle erkannten Geräte als Output vor.

Listing 4.1: Kolter-Algorithmus Überblick[16]

```
1 % total variation denoising
2 [data, ~, ~] = tvdip(true_power);
3
4 % generate snippets from power levels
5 snippets = generateSnippets(data, true_power);
6
7 % generate HMMs from snippets
8 HMMs = generateHMMsFromSnippets(snippets);
9
10 % calculate probability of one snippet generating another
11
12 % build adjacency Matrix from k-nearest neighbor graph
13     mu, Q
14
15 % run AFAMAP algorithm
16 [X0, Z, G] = myAfamap(data, mu, Q);
```

Dieser Zugang erreicht beim Aufschlüsseln der elektrischen Energie eine Erfolgsquote von 87%. Ausführlichere Tests sind im Artikel [5] festgehalten.

#### 4.1.4 Ergebnisse und Probleme

Umfangreiche Experimente mit diesem und ähnlichen Algorithmen haben leider nicht den gewünschten Erfolg geliefert. Die Grundzüge der Methode bestehen darin, zuerst Teilstücke mit gleichen Leistungssignaturen auszuschneiden und in einem Cluster zusammenzufassen. Das erste Clustering ergibt eine Trennung in die vier Typen von Verbrauchern, die in Abschnitt 2.3.1 aufgezählt sind. Unser Simulationsmodell umfasst nur Heizkörper die analog zum Typ-I funktionieren. Innerhalb eines Clusters werden anschließend die Mittelwerte der einzelnen Teilstücke ermittelt. Auf Basis derer schätzt der Algorithmus die Anzahl an

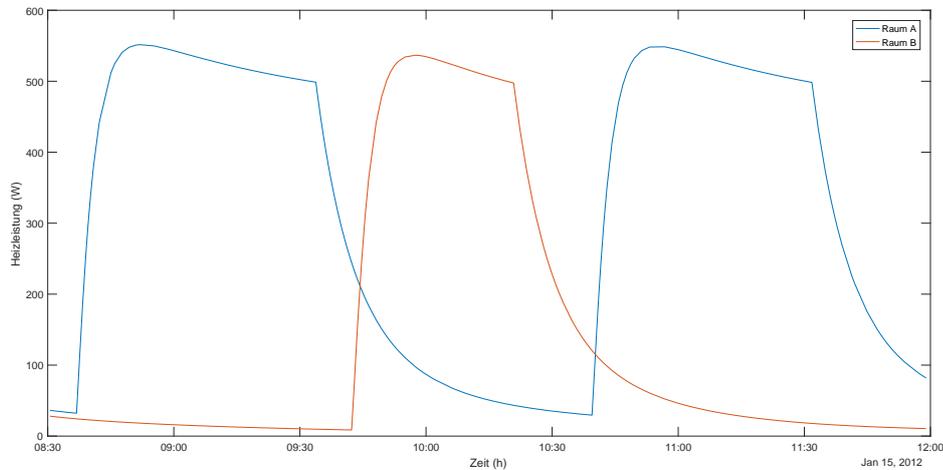


Abbildung 4.1: Beispiele für thermische Leistungsverläufe

Verbrauchern, um danach ein verschränktes FHMM mit der zuvor hergeleiteten AFMAP-Methode zu trainieren. Somit kann der jeweilige Einzelverbrauch errechnet werden. Die Berechnung im Cluster scheitert in unserem Fall. Zurückführen lässt sich dieses Fehlverhalten auf die Ähnlichkeit unserer Leistungsniveaus der einzelnen Räume. Betrachten wir ein typisches Verhalten der Heizleistung in einem Raum A beim Einschalten des Thermostats, erkennen wir eine Kurve, dargestellt in Abbildung 4.1. Die Leistungssignatur ist ähnlich zu jener des Raumes B, wodurch die Räume in ein Cluster fallen, aber das Leistungsniveau und somit der Mittelwert ist ebenfalls gleich. Einzig die Häufigkeit beziehungsweise die Zeitdauer, in welcher die Heizung aktiv ist, ist divergent. Darauf bezieht sich diese Herangehensweise aber nicht. Betrachten wir im Vergleich dazu die Leistungssignaturen eines Kühlschranks und einer Herds, abgebildet in 4.2. Beide Geräte haben den selben Typ-I und landen somit in einem Cluster. Das Leistungsniveau und damit der Mittelwert sind aber verschieden, wodurch der Algorithmus diese Geräte trennen kann. Die Abbildung stellt ungefähr ein Teilstück für den Herd und die zeitgleich passierenden Teilstücke für den Kühlschrank dar.

Dieser Zugang mag für elektrische Energie sehr gut funktionieren, in unserem thermischen Anwendungsfall jedoch können wir keinen Erfolg verbuchen. Die vom wirtschaftlichen Standpunkt beste Variante müssen wir verwerfen und uns neuen Ideen, die mehr Messpunkte benötigen, zuwenden.

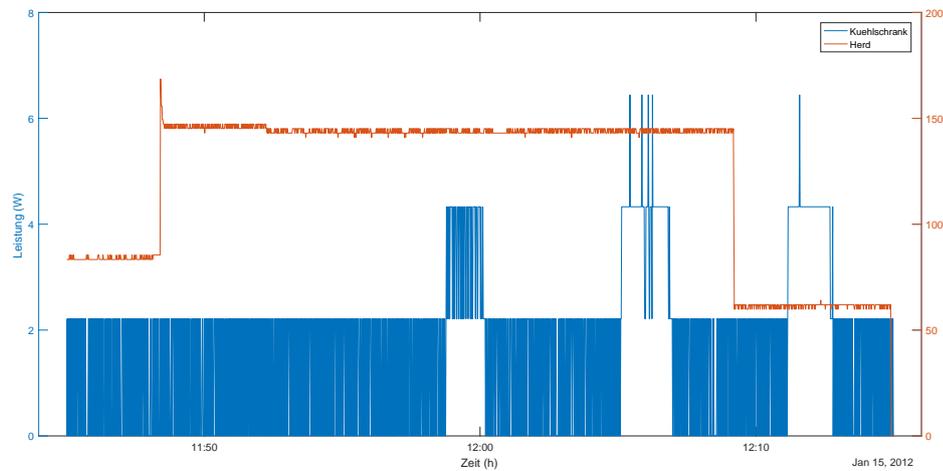


Abbildung 4.2: Elektrische Leistungsverläufe eines Kühlschranks und eines Herds entnommen dem ECO-Dataset [5]

## 4.2 Zugang 2: Linearisiertes Modell

### 4.2.1 Vorbemerkungen

Abhängig von der Raum- und der Vorlauftemperatur ergibt sich nach dem Einschalten der Heizung ein Anstieg der Temperatur in einem Raum. Da sich jeder Heizvorgang im Gesamtenergieverlauf auswirkt, erkennen wir einen Anstieg dessen. Wir wollen im Folgenden aus dieser Abhängigkeit den Einzelverbrauch jedes Heizkörpers bestimmen. Zusätzlich zum Gesamtleistungsverlauf, der sich aus der Steigung des Energieverlaufs berechnet, benötigen wir die Schaltzeitpunkte der Thermostate. Moderne Heizungssysteme haben die entsprechende Hardware schon verbaut, daher müssen wir ausschließlich ein Modul zum Aufzeichnen der Schalterstellung hinzufügen.

Offensichtlicher Weise erhält man durch Addition der Einzelleistungen der Räume zu jedem Zeitpunkt den Summenleistungsverlauf  $\Sigma P$  des Hauses, wobei ein geöffnetes Thermostat Voraussetzung für eine abgegebene Heizleistung ist. Zurück gebrochen auf einen Raum liegt es auf der Hand, dass der Verlauf der Leistung von der Differenz aus Vorlauf- und Raumtemperatur sowie der maximalen Heizleistung abhängig ist. In der Simulation verwendeten wir die Gleichungen (3.2) und (3.3). Setzen wir  $n = 1$  sind beide Abhängigkeiten linear. Daher nehmen wir an, dass dieser Zusammenhang linear ist, und leiten in weiterer Folge zwei systembestimmende Konstanten ab.

- (i)  $K_{R_i}$ : Diese Konstante ist spezifisch für den Raum  $i$ . Hier spiegelt sich die relative abgegebene Leistung wider.
- (ii)  $K(t)$ : Diese Größe ist für jeden Zeitpunkt eine Systemkonstante. Sie skaliert lediglich die Temperaturdifferenz und die Leistung.

Zusammengefasst erhalten wir

$$\Sigma P(t) = \sum_{i=1}^N \Delta T_{R_i}(t) K_{R_i} K(t), \quad (4.8)$$

wobei  $N$  die Anzahl der Räume und  $\Delta T_{R_i} = T_{in} - T_{Room_i}$  die Temperaturdifferenz des  $i$ -ten Raumes beschreibt. Für spätere Betrachtungen wollen wir die Abhängigkeit von der Zeit  $t$  an dieser Stelle nicht unerwähnt lassen. Nun können wir diese Konstanten verwenden, um mit dem Integral über die Zeit den jeweiligen Einzelverbrauch eines Raumes zu bestimmen

$$E_{R_i,t} = \int_{t-1}^t \Delta T_{R_i}(s) K_{R_i} K(s) ds + E_{R_i,t-1} \quad \text{mit } E_{R_i,0} = 0.$$

Da in der hergeleiteten Beziehung die maximale Heizleistung ein versteckter Faktor ist, passen wir unser Simulationsmodell ein wenig an. Bis zum jetzigen Zeitpunkt verfügen alle Räume über die gleiche Heizleistung. Um ein zusätzliches Unterscheidungsmerkmal einzuführen, multiplizieren wir während der Simulation die Leistung mit einem Faktor aus dem Intervall  $[0.5, 1.5]$ . So simulieren wir verschiedene Heizkörpergrößen, die über den  $UA$ -Wert in Gleichung (3.2) definiert sind.

### 4.2.2 Algorithmus

Die zuvor aufgebaute Theorie wollen wir nun in die Praxis übersetzen. Unser Hauptaugenmerk liegt hierbei im Bestimmen beziehungsweise Abschätzen der Konstanten.

#### $\mathbf{K}_{R_i}$

Um diesen relevanten Faktor für die abgegebene Leistung eines Raumes zu bestimmen, extrahieren wir aus dem Summenleistungsverlauf bestimmte Zeitintervalle. Dazu erheben wir aus den vorliegenden Thermostatdaten des jeweiligen Raumes die Einschaltzeitpunkte. Rund um diese Zeitpunkte betrachten wir den Leistungsverlauf genauer. Wir bezeichnen

einen Einschaltzeitpunkt mit  $s$  und schneiden aus dem Summenverlauf das empirisch ermittelte Intervall  $[s - 300, s + 5400]$  heraus. Synchronisiert um diesen Einschaltzeitpunkt summieren wir alle Intervalle auf. Das Rauschen in den Daten, welches durch die Schaltungen der anderen Räume verursacht wird, relativiert sich durch diese Summation. So wird der Einfluss der Heizleistung eines Raumes auf die Gesamtleistung sichtbar, da die Schaltevents in allen Räumen unabhängig voneinander sind. Daraus ergibt sich eine mittlere Leistung vor und während des Intervalls, in dem das Thermostat geöffnet ist. Abbildung 4.3 dient dazu, dies zu verdeutlichen. Aus der Differenz von Minimum und Maximum gewichtet mit der Anzahl  $m_i$  an summierten Teilstücken ergibt sich der Faktor  $K_{R_i}$  für den Raum  $i$

$$K_{R_i} = \frac{P_{i,max} - P_{i,min}}{m_i}.$$

Hierbei dürfen wir zwei wichtige Tatsachen nicht außer acht lassen. Allen voran erweist sich die Anzahl der Stichproben essentiell für die Güte der Schätzung. Übersetzt in unseren Kontext bedeutet dies, je mehr Intervalle wir summieren, desto besser beschreibt unser Faktor  $K_{R_i}$  das Verhalten des Raumes. Außerdem können durch beispielsweise ein im Winter geöffnetes Fenster Fehler aufgrund von atypischem Heizverhalten entstehen. Diese Tatsache muss im Vorhinein berücksichtigt werden, und im besten Fall beziehen wir jene Zeitintervalle nicht in die Berechnungen mit ein.

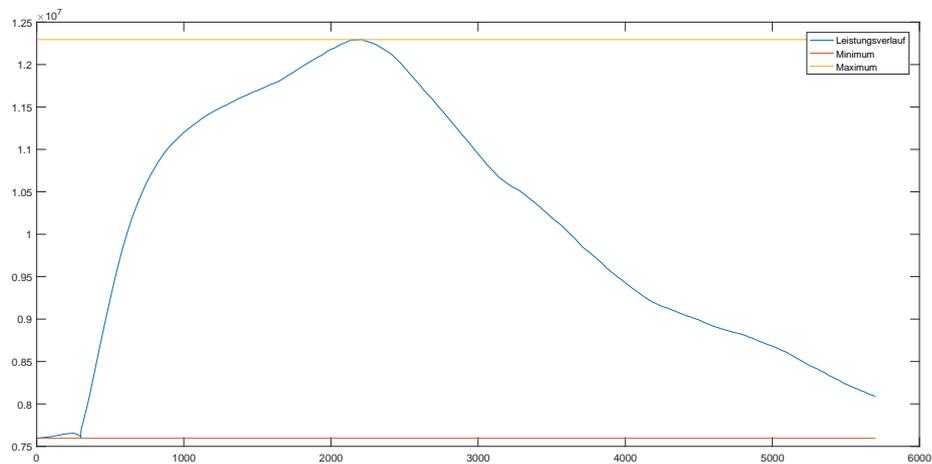


Abbildung 4.3: Beispiel einer Summe der Intervalle eines Raumes

**K**

Die zweite Größe  $K(t)$  ist zeitabhängig, jedoch unabhängig vom Raum. Daher können wir diese aus der Summe in Gleichung (4.8) herausheben und mit einem Umformungsschritt die Äquivalenz formulieren als

$$\begin{aligned}\Sigma P(t) &= K(t) \sum_{i=1}^n \Delta T_{R_i}(t) K_{R_i} \\ K(t) &= \frac{\Sigma P(t)}{\sum_{i=1}^n \Delta T_{R_i}(t) K_{R_i}}.\end{aligned}$$

Da die Summenleistung als erste Ableitung der Kurve des Summenenergieverbrauchs zu jedem Zeitpunkt bekannt ist, können wir die unbekannte Größe  $K(t)$  berechnen.

**Berechnung des Einzelverbrauchs**

Das Ziel ist es den Energieverbrauch eines Raumes zu extrahieren. Folglich müssen wir diesen mit Hilfe des kumulativen Integrals bestimmen. Hierfür integrieren wir die Funktion

$$E_{R_i} = \sum_{j=1}^{N-1} \int_{t_j}^{t_{j+1}} \Delta T_{R_i}(t) K_{R_i} K(t) dt,$$

wobei  $N$  die Anzahl an Beobachtungszeitpunkten bezeichnet. In `Matlab` bedienen wir uns der vor implementierten Funktion `cumtrapz`, welche mit Hilfe der Trapezregel das Integral numerisch annähert.

Listing 4.2: Integral

```
1 kr = k(room,1)*K;  
2 tIn = temperature(room,:);  
3  
4 estimateP2(room,:) = kr.*tIn;  
5 estimateE(room,:) = cumtrapz(steps,estimateP2(room,:));
```

### 4.2.3 Ergebnisse und Probleme

Zu Beginn versuchen wir die raumabhängige Konstante  $K_{R_i}$  zu berechnen. Die dabei aufsummierten Leistungsprofile sollten annähernd dieselbe Gestalt aufweisen wie ein einziger Heizungsimpuls, dargestellt in Abbildung 4.1. Theoretisch besteht die Kumulation aus lauter gleichartigen Signaturen, wodurch sich die Ähnlichkeit zu einem einzelnen Impuls begründen lässt. Die Grafiken in 4.4 zeigen, dass die Formen der summierten Leistungssignaturen voneinander abweichen. Das Trainingsdatenset beinhaltet nicht für jeden Raum die gleiche Anzahl an Schaltzyklen. Ein Raum mit nur einer Außenwand muss weniger Heizen als ein Raum mit drei Außenwänden, zusätzlich verfügen alle Räume über unterschiedliche Heizkörpergrößen, die sich wiederum auf die Anzahl der Heizzyklen auswirkt. In den jeweiligen Überschriften zu den Plots sind mit Großbuchstaben (zum Beispiel E, NSW, SW) die Lage und die Orientierung der Außenwände kodiert, die Zahlenwerte beziehen sich auf den Faktor, um welchen die Standardgröße der Heizung reduziert oder vervielfacht wird (07 bedeutet 70% der im Kapitel 3 beschriebenen Größe)

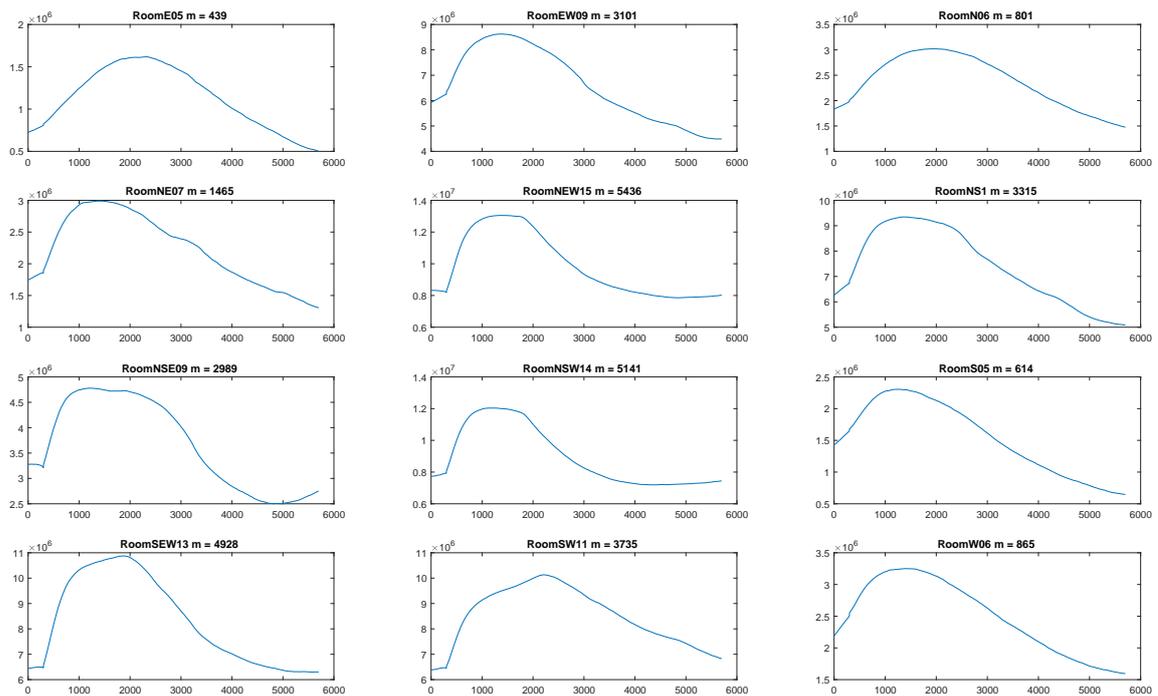


Abbildung 4.4: Summe der Heizintervalle aller Räume eines Stockwerks mit unterschiedlicher Anzahl  $m$

Die Abbildung 4.5 dokumentiert, wie sich die summierte Leistungskurve eines Raumes mit der Anzahl der summierten Intervalle verbessert. Das in Abschnitt 4.2.2 erwähnte Rauschen

verschwindet nach und nach.

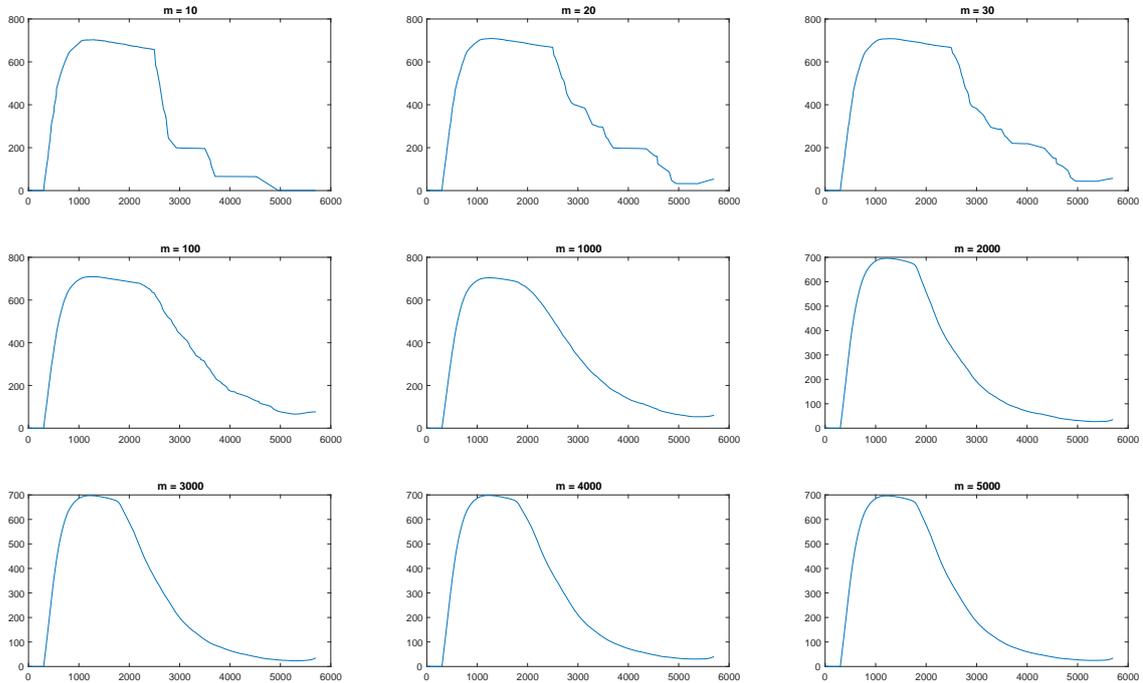


Abbildung 4.5: Evolution der summierten Intervalle eines Raumes

Anschließend führen wir die Integration zur Bestimmung der Energien durch. Auch hier spiegelt sich das oben beschriebene Verhalten wider. Die Fehler, die schon zuvor bei der Abschätzung passiert sind, setzen sich fort. Da wir mit Simulationsdaten arbeiten, können wir die realen Energiewerte aus der Simulation mit den berechneten vergleichen. Zugehörig zu den Größen, die aus den Daten, welche in Abbildung 4.4 dargestellt werden, abgeleitet sind, erhalten wir die Ergebnisse in Abbildung 4.6. Wir können somit nur die Größenordnung des Energieverbrauchs abschätzen. Auf mögliche Verbesserungen gehen wir in einem der folgenden Absätzen ein.

In einem zweiten Experiment reduzieren wir die Anzahl an Räumen. Wir verwenden nur mehr sieben Räume, welche eine hohe Anzahl an summierbaren Intervallen aufweisen. Durch diese Reduktion der Dimension erhalten wir deutlich bessere Ergebnisse und der Unterschied zwischen der realen und der berechneten Energie ist deutlich geringer, wie Abbildung 4.7 belegt.

## 4 Energiemonitoring

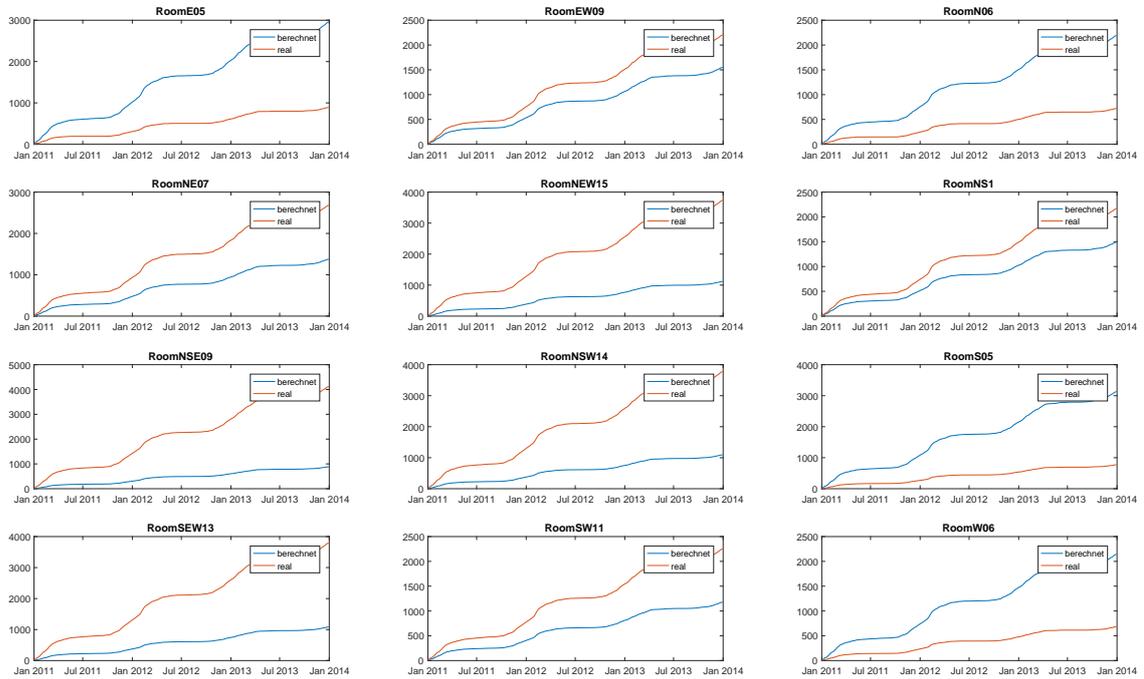


Abbildung 4.6: Energieverläufe aller Räume

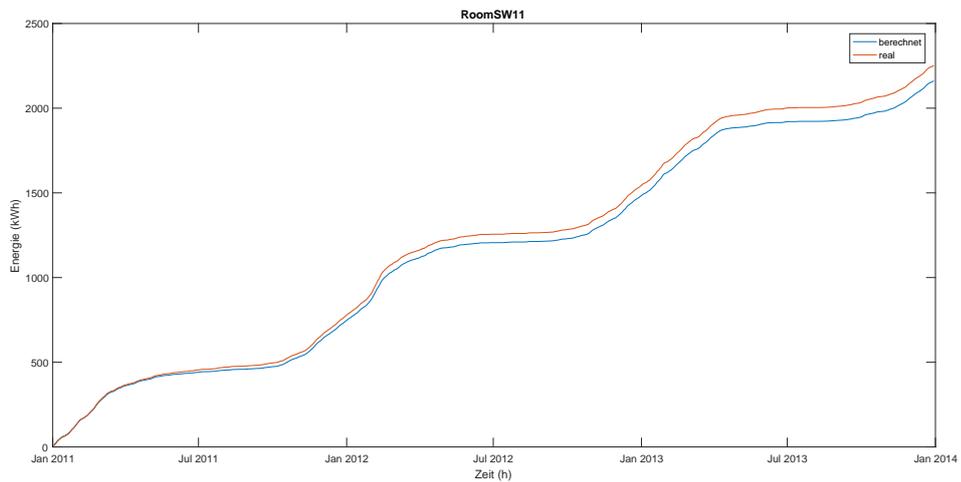


Abbildung 4.7: Vergleich zwischen berechneten und simulierten Energieverlauf

Um den Zusammenhang zwischen Anzahl der Auswertungen  $m$  und dem relativen Fehler

$$d_{Room} = \frac{estimateEnergy_{Room,end} - simulatedEnergy_{Room,end}}{simulatedEnergy_{Room,end}}$$

zu verifizieren, betrachten wir die Abbildung 4.8. Wir berechnen den absoluten Fehler am Ende des Beobachtungszeitraums und setzen diesen in Relation zum Energieverbrauch des Raumes bis zum Endzeitpunkt. Im Zuge des zweiten Experiments erhöhen wir schrittweise die Anzahl  $m_i$  analog zu Abbildung 4.5 und berechnen  $d_{Room}$ . Exemplarisch ist hier der Verlauf des relativen Fehlers für den Raum RoomSW11 dargestellt. Wir können jedoch keine Konvergenz des Fehlers gegen Null beweisen.

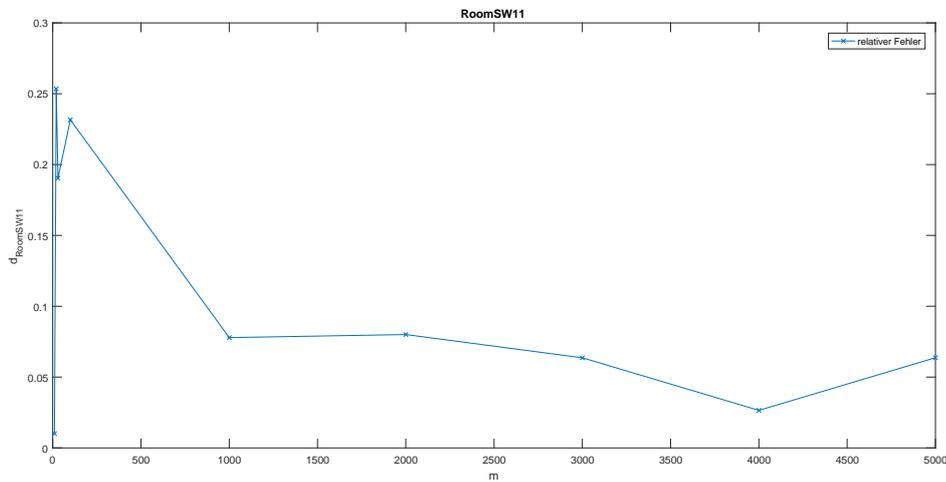


Abbildung 4.8: Relativer Fehler eines Raumes am Ende der Berechnungszeitspanne

Nachdem es sich hierbei um eine Berechnung mit beschränkten Informationen handelt, ist das Ergebnis naturgemäß mit Fehlern behaftet:

- (i) Ein Fehler ergibt sich durch die gewählte Linearisierung des Zusammenhangs zwischen Temperaturdifferenz und Leistungsabgabe. Die Gleichung (3.2), welche im Grunde den Wärmefluss vom Heizkörper in den Raum beschreibt, verhält sich hochgradig nichtlinear. Um ein entsprechendes Modell zu erstellen, bräuchten wir mehr als zwei Parameter. Aus unseren Daten jedoch sind nur die beiden oben verwendeten Konstanten ableitbar.
- (ii) Des Weiteren müssen wir berücksichtigen, dass sich der zeitliche Verlauf der abgegeben Leistung vom Verlauf der gemessenen Leistung unterscheidet. Aus diesem Grund

deckt sich der zeitliche Verlauf der Summe der berechneten Leistungen nicht mit dem gemessenen Summenverlauf. Da diese Größen allerdings energetisch gleich sind und im Wesentlichen der längerfristige Energieverlauf von Interesse ist, hat dieser Effekt vernachlässigbare Auswirkungen.

- (iii) Beim vorgestellten Verfahren wird die abgegebene Leistung aus einer Differenz aus Vorlauf- und Raumtemperatur gebildet, da jene Werte kostengünstig abgelesen werden können oder bereits vorhanden sind. Für exaktere Berechnungen müssten wir statt der Vorlauftemperatur die Temperatur des Heizungssystems heranziehen, welche wiederum von der Raumtemperatur und der Durchflussgeschwindigkeit des Transportmediums durch den Heizkörper abhängt. Wir vernachlässigen diese komplexen und schwieriger messbaren Abhängigkeiten, da der Temperaturunterschied verhältnismäßig klein ist und wir die Durchflussrate als konstant annehmen. Diese Faktoren fassen wir somit auch in der Konstante  $K_{R_i}$  zusammen.

Neben der aufgezählten limitierenden Faktoren können wir diesen Ansatz in Zukunft weiterverfolgen. Wenn wir den Beobachtungszeitraum verlängern, werden die Ergebnisse noch besser. Hier müssen wir aber einen Kompromiss zwischen Beobachtungszeit und Fehler eingehen, da Kunden in der Realität nicht erst mehrere Jahre nach der Installation Ergebnisse geliefert haben wollen.

### 4.3 Zugang 3: Clustern

#### 4.3.1 Vorbemerkungen

Da die Zugänge, mit welchen wir direkt über die Leistung zum Energieverbrauch gelangen wollen, zu keinem Erfolg geführt haben, verbauen wir nun weitere Sensoren und zeichnen die Außentemperatur zu jedem Zeitpunkt auf. Zusätzlich kennen wir wieder die Vorlauftemperatur, mit der das Heizungssystem betrieben wird. Bevor wir den geforderten Einzelverbrauch bestimmen, streben wir nach einer Klassifizierung der vorliegenden Daten. Vor allem beim zuvor beschriebenen Zugang benötigen wir gesicherte Aussagen über die Güte der Werte. Das Ziel ist es, fehlerfreie von fehlerbehafteten Datensätzen zu trennen.

Unsere zugrunde liegende Idee beruht sich auf einen Zusammenhang zwischen der Steigung der Temperaturkurve und der Heizleistung, wie die Gleichung (3.5) und die davon abgeleiteten darlegen. Um unseren Raum von einer Grundtemperatur auf eine andere aufzuheizen, brauchen wir Leistung. Je höher diese ist, desto schneller wird es im Raum wärmer, was sich

wiederum in einer steileren Temperaturkurve zeigt. Eine entscheidende Rolle spielt hierbei jedoch auch der Wärmefluss durch Fenster oder Wände. Dieser Verlust oder Gewinn an Wärme spiegelt sich ebenso in unserer Steigung wider. Die zugehörigen Temperaturwerte, an denen wir die Steigung messen, müssen immer ident sein. Damit ist eine Vergleichbarkeit gegeben. Die Funktionsweise einer Heizungsanlage mit Thermostatsteuerung liefert uns hier automatisch Referenzpunkte. Wir wählen als Grundtemperatur  $T_G$   $19^\circ\text{C}$ , weil diese In-entemperatur in unserer Simulation gleichzeitig das Signal zum Aktivieren der Heizung in dem Raum gibt. Unsere Aufzeichnung der Temperaturkurve beenden wir bei willkürlich festgelegten  $21^\circ\text{C}$ . Obligatorisch ist es, dass unser System die geforderte Endtemperatur  $T_E$  immer erreichen kann. Die Zuordnung, welche eine Solltemperatur von  $21^\circ\text{C}$  und eine Hysterese für Thermostatschaltungen von  $\pm 2^\circ\text{C}$  umfasst, kann trivialer Weise nie eine Endtemperatur von  $25^\circ\text{C}$  nur durch Heizleistung erreichen. Haben wir nun den Zeitpunkt  $t_1$ , der  $19^\circ\text{C}$  markiert, und jenen  $t_2$ , an welchem wir die Endtemperatur überschreiten, bestimmt, nähern wir die Steigung an

$$k = \frac{T_E - T_G}{t_2 - t_1}.$$

Nun können wir für das anschließende Clustering verschiedene Ansätze wählen, die unter anderem auch weitere Ziele verfolgen.

### Ansatz 1

Wir nehmen an, dass die Steigung der Temperaturkurve eines Raumes direkt von der Temperatur in den Nachbarräumen abhängt. Da jeder Raum über bis zu sechs angrenzende Zimmer verfügt, betrachten wir ein Problem in einer relativ hohen Dimension. Um dies zu verdeutlichen, möchten wir in Tabelle 4.1 einen beispielhaften Datensatz zu Raum Room0SW angeben. Die Nachbarn liegen in der Reihenfolge Nord, Süd, Ost, West, Unten und Oben vor, welche durch den Grundriss in Abbildung 3.1 gegeben ist.

Tabelle 4.1: Room0SW mit den Nachbarn (im)

Nord ( $^\circ\text{C}$ )	Süd ( $^\circ\text{C}$ )	Ost ( $^\circ\text{C}$ )	West ( $^\circ\text{C}$ )	Unten ( $^\circ\text{C}$ )	Oben ( $^\circ\text{C}$ )	Klasse
19.9	0.2	20	0.2	19	18.9	$14 \cdot 10^{-4}$
19.8	0.2	19.1	0.2	19	19.1	$12 \cdot 10^{-4}$
20.3	0.2	20.7	0.2	19	19.2	$18 \cdot 10^{-4}$

Die Klassenzugehörigkeit wird durch die berechnete Steigung  $k$  bestimmt. Dies bedeutet, dass Objekte, welche die gleiche Steigung, gerundet auf vier Nachkommastellen, haben,

derselben Klasse angehören. Die Klassifizierung sollte anschließend autonom entscheiden, ob dieser Raum die richtige Steigung und somit ein normales Heizverhalten hat oder nicht.

## Ansatz 2

Die hohe Dimension des vorherigen Weges kann zu Problemen führen und die Rechenzeit des Clusterings verlängern. Wir suchen daher nach einer Möglichkeit, die mit weniger Information arbeitet. Die Raumtemperatur in den umliegenden Räumen hält sich bedingt durch unseren Versuchsaufbau meistens im Intervall zwischen  $19^{\circ}\text{C}$  und  $23^{\circ}\text{C}$  auf beziehungsweise ist sie ähnlich zu der Temperatur des betrachteten Raumes. Dadurch entfernen wir die Temperatur der Nachbarräume aus unserer Abhängigkeit.

Betrachten wir die Temperaturkurve zum Schaltzeitpunkt des Thermostats in Abbildung 4.9 genauer, bemerken wir, dass diese trotz eingeschalteter Heizung noch fällt. Im Ansatz zuvor schenkten wir dieser Auffälligkeit keinerlei Beachtung. Damit wir diese versteckte Information nützen können, ermitteln wir die Steigung auf einem kleineren Intervall um den Schaltzeitpunkt. Zehn Sekunden zuvor und danach geben eine ausreichende Näherung der ersten Ableitung der Kurve. Diesen Vorgang wiederholen wir am Endzeitpunkt des Analysefensters, sobald  $21^{\circ}\text{C}$  überschritten werden, und erhalten nun zwei Werte für die Steigung.

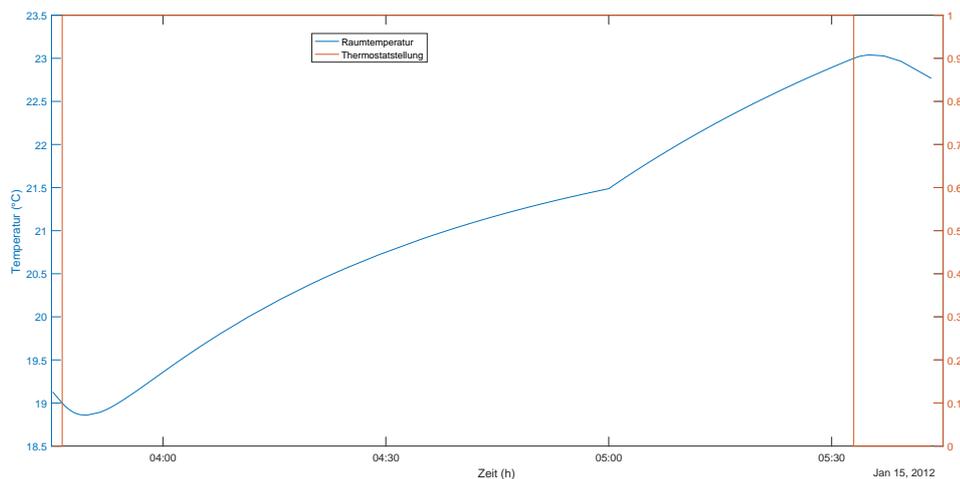


Abbildung 4.9: Temperaturkurve einer Raumes

Die Anzahl an Klassen gilt es im Vergleich zu unserem ersten Ansatz ebenso zu reduzieren. Da die Außentemperatur die größte Abweichung von der Raumtemperatur aufweist, ist der Wärmeverlust durch die Außenwände am höchsten. Aufgrund dieser Tatsache entscheiden wir uns für ein Clustering mit nur drei Klassen, welche wir von der Anzahl an Außenwänden ableiten. Der für die Klassifizierung vorbereitete Datensatz hat die Form von Tabelle 4.2.

Tabelle 4.2: Room0SW

Außentemperatur (°C)	Steigung 19°C	Steigung 21°C	Klasse
2	0.8193	0.0014	2
0	-0.0490	0.0187	2

Da wir die dreidimensionale Punktwolke ohne Transformation darstellen können, färben wir jede Klasse in einer anderen Farbe und generieren die Abbildung 4.10. Hier ist schon erkennbar, dass die Trennung dieses Punktehaufens eine große Herausforderung mit sich bringt.

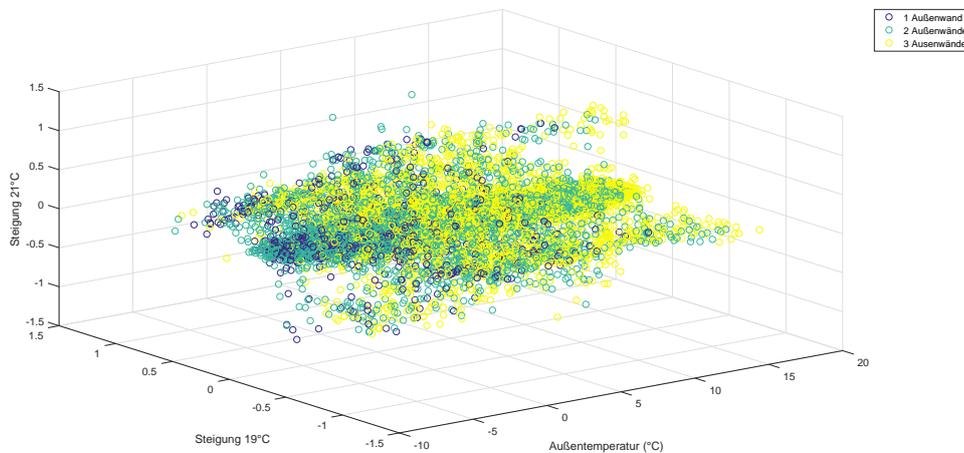


Abbildung 4.10: Punktwolke, die im zweiten Ansatz klassifiziert wird

### 4.3.2 Support Vector Machine SVM

Dieses Unterkapitel orientiert sich an dem Buch [6] und dem Paper [7]. Hierauf möchten wir auch für weiterführende Informationen verweisen.

## Grundlegende Idee und Funktionsweise

Die SVM konstruiert eine Hyperebene oder eine Menge von Hyperebenen, welche die Objekte in Klassen trennt. Jedes Objekt der Trainingsdaten wird hierfür durch einen Vektor in einem Vektorraum geeignet repräsentiert.

Da es sich bei der SVM um einen *Large-Margin-Classifer* handelt, versucht die SVM den Abstand der Punkte, die am nächsten zur Hyperebene liegen, zu maximieren. Dadurch wird sichergestellt, dass hinzukommende Objekte, die nicht unseren Trainingsobjekten entsprechen, auch möglichst zuverlässig der richtigen Klasse zugeordnet werden. Sowohl beim Einpassen der Hyperebene als auch beim mathematischen Beschreiben selbiger benötigen wir nur die nächstgelegenen Vektoren. Diese Vektoren stützen somit die Hyperebene, woraus sich der Name SVM (deutsch Stützvektormaschine) ableitet.

Bis jetzt haben wir nur Trainingsobjekte betrachtet, die linear trennbar sind. Im Allgemeinen ist dies bei realen Anwendungen nicht immer möglich. Hierbei bedienen wir uns des *Kernel-Tricks*. Der damit verbundene Vorgang ist, die Vektoren in einen höherdimensionalen Raum zu transformieren. In diesem neuen Raum, dessen Dimension auch unendlich sein kann, ist es uns möglich, die Trainingsobjekte wieder linear zu trennen. Nun müssen wir die Trennebene mit einer geeigneten Rücktransformation auf die Anfangsdimension bringen, wodurch eine nichtlineare Grenze entsteht.

## Klassifizierung mit maximaler Margin

Um die Funktionsweise der SVM besser zu verstehen, beginnen wir mit der zwei Klassenklassifizierung, welche durch lineare Modelle der Bauart

$$y(x) = w^T \phi(x) + b, \tag{4.9}$$

wobei  $\phi(x)$  eine Transformation in einen fixen Merkmalsraum und  $b$  den Verzerrungsparameter beschreibt, definiert ist. Um auf explizite Berechnungen in diesem Merkmalsraum zu verzichten, werden wir im weiteren Verlauf zur dualen Formulierung übergehen und auf Kernel-Funktionen zurückgreifen. Die Trainingsdaten beinhalten  $N$  Vektoren  $x_1, \dots, x_N$  mit den dazugehörigen Zielwerten  $t_1, \dots, t_N$ , wobei  $t_n \in \{-1, 1\}$ . Neue Datenpunkte  $x$  werden nach dem Vorzeichen von  $y(x)$  klassifiziert.

Für den Moment nehmen wir an, dass die Menge von Trainingsdaten linear trennbar ist. Somit existiert zumindest eine Wahl der Parameter  $w$  und  $b$ , sodass für die Funktion der

Form (4.9)  $y(x_n) > 0$  für Punkte mit Zielwert  $t_n = +1$  und  $y(x_n) < 0$  für Punkte der anderen Klasse gilt. Für alle Trainingsdaten ist  $t_n y(x_n)$  positiv.

Verständlicherweise können wir mehrere Hyperebenen im Merkmalsraum finden, die unsere Trainingsdaten trennen. Abhängig von der Wahl der Startwerte von  $w$  und  $b$  beziehungsweise der Reihenfolge der Trainingsdaten ändert sich unsere Lösung. Wie oben erwähnt, wählen wir bei der SVM die Hyperebene, welche die breiteste Margin hat. Die Margin ist definiert als der kleinste Abstand zwischen der Entscheidungsgrenze und einem Punkt, wie in Abbildung 4.11 dargestellt.

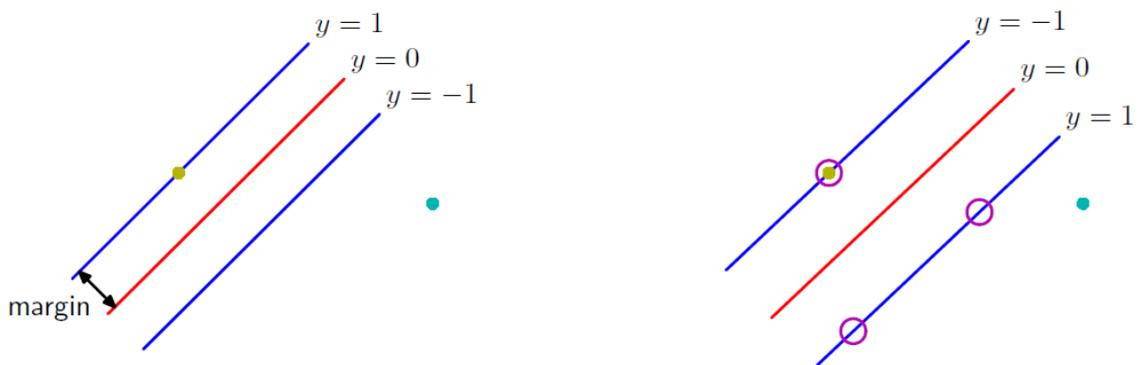


Abbildung 4.11: Margin [6]

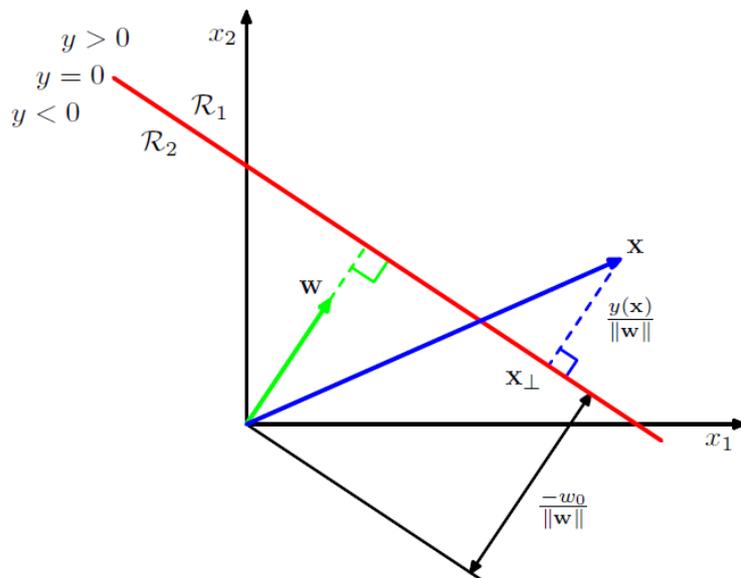


Abbildung 4.12: Geometrische Überlegungen zur Gleichung (4.10) [6]

Durch einfache geometrische Überlegungen (s. Abbildung 4.12) ergibt sich, dass wir den Normalabstand von einem beliebigen Punkt  $x$  im Merkmalsraum zur Hyperebene, die durch  $y(x) = 0$  mit (4.9) beschrieben wird, durch  $\frac{|y(x)|}{\|w\|_2}$  berechnen können. Zusätzlich sind wir nur an Lösungen interessiert, welche alle Datenpunkte korrekt klassifizieren, formalisieren lässt sich dies durch  $t_n y(x_n) > 0$  für alle  $n$ . Der Abstand für einen Punkt  $x_n$  zur Entscheidungsebene ist somit gegeben durch

$$\frac{t_n y(x_n)}{\|w\|_2} = \frac{t_n (w^T \phi(x_n) + b)}{\|w\|_2}. \quad (4.10)$$

Wir wollen nun die Parameter  $w$  und  $b$  optimieren, um die maximale Margin zu erhalten. Die *Maximale-Margin*-Lösung erreichen wir durch das Lösen der Optimierungsaufgabe

$$\arg \max_{w,b} \left\{ \frac{1}{\|w\|_2} \min_n [t_n (w^T \phi(x_n) + b)] \right\}, \quad (4.11)$$

wobei wir den Faktor  $\frac{1}{\|w\|_2}$  aus der Minimierung über  $n$  herausgezogen haben, da  $w$  nicht von  $n$  abhängt. Direktes Lösen der Optimierung wäre sehr komplex, daher probieren wir nun diese in ein äquivalentes Problem umzuformen, das einfacher zu lösen ist. Hierfür benützen wir die Tatsache, dass sich der Abstand zwischen einem Punkt  $x_n$  und der Hyperebene unter der Umskalierung  $w \rightarrow \kappa w$  und  $b \rightarrow \kappa b$ , gegeben durch  $\frac{t_n y(x_n)}{\|w\|_2}$ , nicht ändert. Dies können wir verwenden, um den Abstand des nächsten Punktes zur Entscheidungsgrenze zu normieren

$$t_n (w^T \phi(x_n) + b) = 1. \quad (4.12)$$

In diesem Fall erfüllen alle Datenpunkte die Nebenbedingung

$$t_n (w^T \phi(x_n) + b) \geq 1, \quad n = 1, \dots, N. \quad (4.13)$$

Für alle Punkte, welche die Ungleichung (4.13) erhalten, nennen wir die Nebenbedingung aktiv, sonst bezeichnen wir sie als inaktiv. Es gibt immer mindestens einen Punkt mit einer aktiven Bedingung, da per definitionem ein nächster Punkt existiert. Sobald die Margin maximiert ist, sind mindestens zwei Bedingungen aktiv. Das Optimierungsproblem vereinfacht sich somit zum Maximieren von  $\|w\|_2^{-1}$ , was äquivalent zum Minimieren von  $\|w\|_2^2$  ist.

Wir lösen

$$\arg \min_{w,b} \frac{1}{2} \|w\|_2^2 \quad (4.14)$$

mit den Nebenbedingungen (4.13). Den Faktor  $\frac{1}{2}$  fügen wir hinzu, um eine komfortablere Lösung zu erhalten. Hier sehen wir ein klassisches Beispiel eines *Quadratic-programming*-Problems, bei dem wir ein Minimum einer quadratischen Funktion unter linearen Nebenbedingungen suchen. Auf den Parameter  $b$  kann bei der Optimierung nicht verzichtet werden, da Änderungen in  $b$  auch  $\|w\|_2$  verändern.

Um in weiterer Folge dieses Optimierungsproblem zu lösen, benutzen wir *Lagrange*-Multiplikatoren mit je einem Multiplikator  $a_n$  pro Nebenbedingung in (4.13). Daraus ergibt sich die *Lagrange*-Funktion

$$L(w, b, a) = \frac{1}{2} \|w\|_2^2 - \sum_{n=1}^N a_n \{t_n (w^T \phi(x_n) + b) - 1\}, \quad (4.15)$$

wobei  $a = (a_1, \dots, a_n)$  ist. Setzen wir die Ableitungen von  $L(w, b, a)$  nach  $w$  und  $b$  Null, folgen die zwei Bedingungen

$$w = \sum_{n=1}^N a_n t_n \phi(x_n) \quad (4.16)$$

$$0 = \sum_{n=1}^N a_n t_n. \quad (4.17)$$

Nun können wir  $w$  und  $b$  aus  $L(w, b, a)$  eliminieren und enden bei der dualen Formulierung unseres Problems (4.14), bei welcher wir

$$\tilde{L}(a) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(x_n, x_m) \quad (4.18)$$

unter den Nebenbedingungen

$$a_n \geq 0 \quad n = 1, \dots, N \quad (4.19)$$

$$\sum_{n=1}^N a_n t_n = 0 \quad (4.20)$$

maximieren. Hier ist die Kernel-Funktion definiert als  $k(x, x') = \phi(x)^T \phi(x')$ . Diese entspricht somit einem Skalarprodukt in einem un spezifizierten hochdimensionalen Merkmalsraum.

Man kann zeigen, dass Optimierungen dieser Form die *Karush-Kuhn-Tucker* (KKT) Bedingung [18] erfüllen, die in diesem Zusammenhang

$$\begin{aligned} a_n &\geq 0 \\ t_n y(x_n) - 1 &\geq 0 \\ a_n \{t_n y(x_n) - 1\} &= 0 \end{aligned}$$

lautet. Daraus folgt, dass für jeden Punkt der Trainingsdaten  $a_n = 0$  oder  $t_n y(x_n) = 1$  gilt. Dadurch treten in der Summe in (4.21) nur Punkte auf, für die  $t_n y(x_n) = 1$  gilt, also jene, die genau auf der maximalen Margin liegen. Diese Punkte nennt man, wie früher bereits erwähnt, Stützvektoren.

Wir haben nun das *Quadratic-programming*-Problem gelöst und einen Vektor  $a$  gefunden. Zum Bestimmen des Parameters  $b$  verwenden wir während der Optimierung einen Stützvektor  $x_n$  und lösen gemeinsam mit (4.21) die Gleichung

$$t_n \left( \sum_{m \in \mathcal{S}} a_m t_m k(x_n, x_m) + b \right) = 1$$

für  $b$ , wobei  $\mathcal{S}$  die Menge der Indices der Stützvektoren ist. Einen Durchschnitt über alle Stützvektoren zu verwenden, stellt eine numerisch stabilere Variante dar

$$b = \frac{1}{N_{\mathcal{S}}} \sum_{n \in \mathcal{S}} \left( t_n - \sum_{m \in \mathcal{S}} a_m t_m k(x_n, x_m) \right)$$

mit  $N_{\mathcal{S}} = |\mathcal{S}|$  [6].

Neue Daten können nun mit diesem trainierten Modell anhand des Vorzeichens von  $t_n y(x)$  klassifiziert werden. Hierfür benutzen wir (4.9) und (4.16) und berechnen

$$y(x) = \sum_{n=1}^N a_n t_n k(x, x_n) + b. \tag{4.21}$$

### Kernel-Funktion

Die Kernel-Funktion entspricht einem inneren Produkt im Merkmalsraum, dessen Dimension sogar unendlich sein kann. Es liegt die Vermutung nahe, dass wir durch direktes Arbeiten mit Kernel-Funktionen dem Dimensionsproblem aus dem Weg gehen können. Dies ist nicht der Fall, wie das folgende einfache Beispiel zeigt. Hierfür betrachten wir eine polynomielle Kernel-Funktion der Ordnung 2, die wir erweitern können

$$\begin{aligned}k(x, z) &:= (1 + x^T z)^2 = (1 + x_1 z_1 + x_2 z_2)^2 = \\&= 1 + 2x_1 z_1 + 2x_2 z_2 + x_1^2 z_1^2 + 2x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\&= \left(1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1 x_2, x_2^2\right) \left(1, \sqrt{2}z_1, \sqrt{2}z_2, z_1^2, \sqrt{2}z_1 z_2, z_2^2\right)^T \\&= \phi(x)^T \phi(z).\end{aligned}$$

Hier beschreibt die Kernel-Funktion ein inneres Produkt in einem sechsdimensionalen Merkmalsraum, obwohl die Ausgangsvektoren nur zweidimensional sind. Durch die Funktion  $\phi$  wird die Transformation von dem Ursprungsraum in den Merkmalsraum definiert. Daher kann jede Teilmenge von Punkten im originalen Raum in genau eine nichtlineare zweidimensionale Mannigfaltigkeit im sechsdimensionalen Merkmalsraum eingebettet werden.

Die optimale Kernel-Funktion hängt daher stark vom gegebenen Klassifizierungsproblem ab und wir müssen diese selbst wählen. Einige populäre Kernel-Funktionen sind [28]:

- (i) linear:  $k(x, y) = \langle x, y \rangle$
- (ii) quadratisch:  $k(x, y) = \langle x, y \rangle^2$
- (iii) kubisch:  $k(x, y) = \langle x, y \rangle^3$
- (iv) Gaußsche Radial Basisfunktion (RBF):  $k(x, y) = \exp\left(-\gamma \|x - y\|_2^2\right)$  mit  $\gamma > 0$

### 4.3.3 Ergebnisse und Probleme

Wie die Abbildung 4.13 zeigt, liefert das Clustering keine annehmbaren Ergebnisse. Deutlich erkennbar ist ein dichter Haufen an Punkten, der gleichzeitig das Problem für unsere Algorithmen darstellt. Die große Masse der Auswertungen liegt sehr eng beisammen beziehungsweise überschneidet sich, wenn sie auf zwei Dimensionen reduziert dargestellt werden. Keine Klassifizierung hat eine Erfolgsquote von über 50%, wodurch auch dieser Zugang für die weitere Betrachtung und Analyse nicht relevant ist. In Tabelle 4.3 sind die Ergebnisse

des zweiten Ansatzes festgehalten. Der erste Ansatz umfasst über 100 Klassen, wodurch die Klassifizierung in keiner annehmbaren Rechenzeit möglich ist. Eine SVM kann immer nur zwei Klassen voneinander trennen, dies wird uns bei dieser großen Anzahl an verschiedenen Klassen zum Verhängnis.

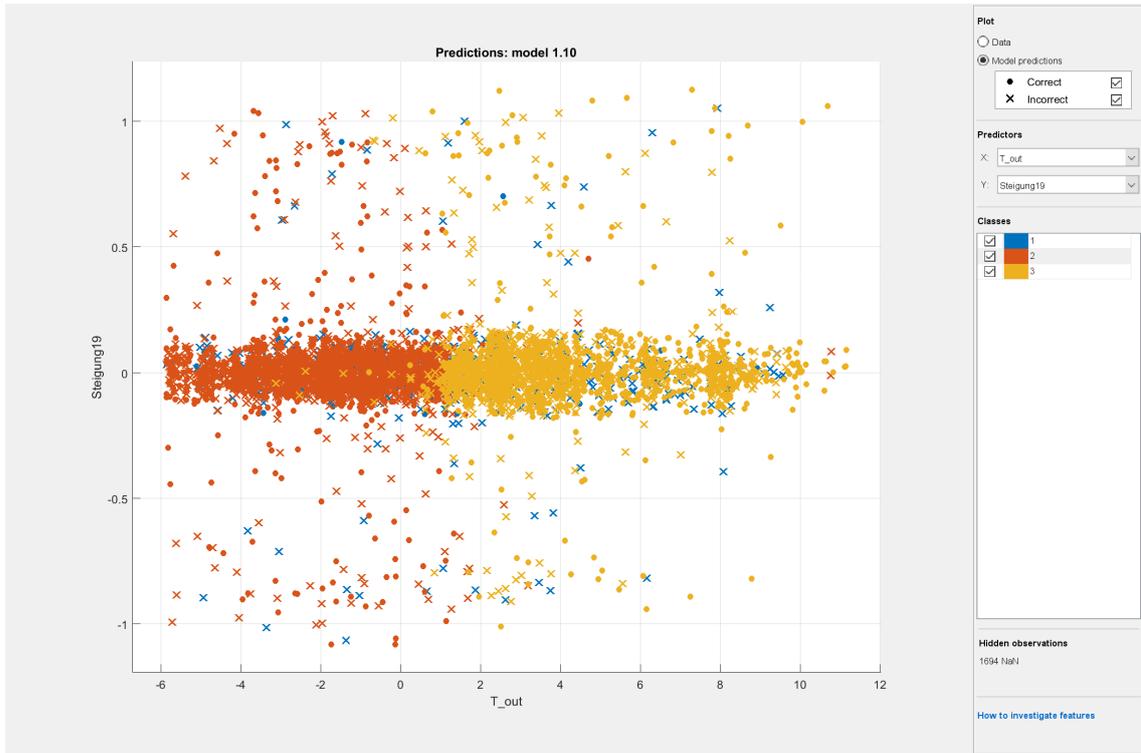


Abbildung 4.13: Clustering mit einer SVM

Tabelle 4.3: Ergebnisse des Clustering nach dem zweiten Ansatz

Kernel	lin	quad	kubisch	RBF ( $\gamma = \sqrt{3}/4$ )	RBF ( $\gamma = \sqrt{3}$ )	RBF ( $\gamma = \sqrt{3} \cdot 4$ )
<b>Erfolgsquote</b>	38.2%	33.4%	34.8%	39.7%	40.1%	40.0%

Die Steigung der Temperaturkurve liefert somit keinen neuen Informationen über den Zustand des Heizungssystems.

## 4.4 Zugang 4: Atypisches Heizverhalten

### 4.4.1 Vorbemerkungen

Alle bisher genannten Algorithmen benutzen zum Training bereits vorhandene Daten. Wenn wir fehlerhafte Daten in der Trainingsphase ausschließen können, erhöht sich die Erfolgsquote des Energiemonitorings. Ebenso stellt diese Fehleranalyse einen wichtigen Indikator für Gebäudeverwalter dar, um Räume gezielt und frühzeitig zu überprüfen und Schäden am Gebäude rasch zu erkennen und zu beheben. Im Folgenden beschreiben wir eine Anwendung, die atypische Heizzyklen aus dem Datensatz identifiziert. Hierfür ziehen wir lediglich das jeweilige Schaltverhalten des Raumes und die Kurve der Außentemperatur heran.

Wir geben vor, dass die Heizungsanlage die Raumtemperatur in einem gewissen Bereich halten soll. Daraus leiten sich ein Beginn, die Unterschreitung einer bestimmten Temperatur, und ein Ende, das Erreichen der Temperaturobergrenze, der Heizperiode ab. Die Dauer dieses Heizvorgangs hängt neben der konstanten Vorlauftemperatur auch von der Raumtemperatur der angrenzenden Räume und der Außentemperatur ab. Diese Zusammenhänge wollen wir für unsere Analyse nutzen.

In der Regel sollte der Heizvorgang jeweils unter ähnlichen Bedingungen annähernd dieselbe Dauer aufweisen. Wir nehmen an, dass die Erwärmung des Raumes bei gegebener Außen- und Vorlauftemperatur immer einen gewissen Zeitraum  $\Delta t = t + \epsilon$ , wobei  $\epsilon$  einen Fehlerterm darstellt, benötigt. Betrachten wir alle Heizperioden eines einzelnen Raumes, können wir eine Verteilung für  $\Delta t$  beziehungsweise den Fehler  $\epsilon$  ablesen.

Hier gilt in Analogie zu den oben genannten Zugängen, dass die Güte der Verteilung direkt proportional zur Anzahl an Datensätzen ist. Da wir nur eine begrenzte Beobachtungsperiode von drei Jahren zur Verfügung haben und die Verteilung für jeden externen Faktor eine andere ist, müssen wir diese Abhängigkeiten reduzieren. Das Haus in unserer Simulation verfolgt eine Heizstrategie, wonach in jedem Raum die Temperatur in einem gewissen, relativ kleinen, Bereich liegt. Den Wärmefluss durch die Innenwände vernachlässigen wir somit und verringern die Verknüpfungen. Die Vorlauftemperatur ist zu jedem Zeitpunkt im System konstant und wir müssen diese in unserem Modell nicht weiter berücksichtigen. Der übrig gebliebene Faktor, die Außentemperatur, teilt die Heizperioden  $\Delta t$  in verschiedene Klassen, die wir einzeln statistisch weiterverarbeiten. Hier gilt es wieder, die richtige Balance zwischen der Anzahl an Klassen und der jeweiligen Klassengröße zu finden. Experimente haben gezeigt, dass wir die besten Ergebnisse mit den folgenden sechs Klassen erreichen:  $(-\infty, -10]$ ,  $(-10, -5]$ ,  $(-5, 0]$ ,  $(0, 5]$ ,  $(5, 10]$  und  $(10, \infty)$ . An dieser Stelle sei angemerkt,

dass die abgeleiteten Temperaturbereiche im Allgemeinen nicht für alle geografischen Orte gleich gewählt werden können. Nach dieser Einteilung schätzen wir für jede Klasse eine Verteilung. Können wir einen Wert in der Klasse nicht der durch die restlichen Werte bestimmten Verteilung zuordnen, spricht dies für eine signifikante Änderung des Fehlerterms  $\epsilon$  und damit für ein atypisches Heizverhalten des Raumes.

In der Simulation finden wir kein atypisches Heizverhalten, welches der Algorithmus entdecken soll. Um trotzdem ein geeignetes Testdatenset vorliegen zu haben, starten wir eine zweite Simulation mit geänderten Voraussetzungen. Bei drei der 39 Räume ersetzen wir die Außenwand durch eine Innenwand und simulieren so einen Schaden an der Isolierung. Auch für diesen schadhafte Fall betrachten wir die Heizperiode  $\Delta t$ . Für die Tests des Algorithmus mischen wir die fehlerhaften Daten unter jene der ersten korrekten Simulation.

### 4.4.2 Autoregressions-Modell AR

Wir haben Bedarf an einem Algorithmus, der selbstständig Ausreißer in einer Zeitreihe findet und meldet. Fündig werden wir in einem Paper der japanischen Autoren Yamanishi und Takeuchi [43]. Wir nehmen an, dass unsere Zeitintervalle als Zeitreihe vorliegen. Die beliebteste Methode zur Analyse solcher Reihen ist ein AR Modell. Der vorgestellte Algorithmus schätzt iterativ verschiedene Parameter und entscheidet anhand eines Scores, ob es sich um einen falschen Wert handelt. Von den herkömmlichen Methoden hebt sich dieser Algorithmus ab, da er

- (i) Daten online verarbeiten und
- (ii) auf nichtstationäre Zeitreihen adaptiert werden kann.

Der entscheidende Grund für die Wahl dieser Methode liegt offensichtlich im ersten Punkt. Wir wollen atypisches Heizverhalten erkennen, sobald es eintritt und nicht erst einer nach Beobachtungsphase.

### Autoregressions-Modell

Wir sprechen von einem AR Modell, wenn die Elemente der Zeitreihe  $\{x_t : t = 1, 2, \dots\}$  linear und statisch von ihrem Vorgänger abhängen [15]:

$$x_t = \beta_1 \cdot x_{t-1} + \beta_0 + \epsilon_t,$$

wobei  $\beta_0$  die Konstante des Modells,  $\beta_1$  die Parameter und  $\epsilon_t$  ein Rauschen beschreibt. Falls die Abhängigkeit von mehreren - konkret  $p$  - vorherigen Werten gegeben ist, spricht man von einem AR Modell der Ordnung  $p$ .

### SDAR Algorithmus

Zusätzlich zu unserer Zeitreihe  $x_t$  betrachten wir eine besondere, welche als Erwartungswert für die Initialwerte 0 annimmt. Für diese Zeitreihe  $\{z_t : t = 1, 2, \dots\}$  gilt der Zusammenhang

$$z_t = \omega \cdot z_{t-k}^{t-1} + \epsilon.$$

Hier ist  $\epsilon$  von einer normalverteilten Zufallsvariable mit Erwartungswert 0 und Standardabweichung  $\Sigma$  erzeugt,  $\mathcal{N}(0, \Sigma)$ .  $z_t$  ist ein AR Modell  $k$ -ter Ordnung und  $z_{t-k}^{t-1} = (z_{t-1}, z_{t-2}, \dots, z_{t-k})$  und  $\omega = (\omega_1, \dots, \omega_k)$  bezeichnen  $k$ -dimensionale Vektoren aus den reellen Zahlen. Die eigentliche Zeitreihe  $x_t$ , die durch unsere Experimente gegeben ist, leitet sich von  $z_t$  ab

$$x_t = z_t + \mu.$$

Daher können wir für  $x_{t-k}^{t-1} = (x_{t-1}, x_{t-2}, \dots, x_{t-k})$  die Dichtefunktion der Wahrscheinlichkeitsverteilung wie folgt ableiten

$$p(x_t | x_{t-k}^{t-1}; \theta) = \frac{1}{(2\pi)^{k/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (x_t - w) \Sigma^{-1} (x_t - w)\right), \quad (4.22)$$

wobei  $w = \omega (x_{t-k}^{t-1} - \mu) + \mu$ . Im Parameter  $\theta$  fassen wir alle unsere Verteilung beschreibenden Faktoren zusammen,  $\theta = (\omega_1, \dots, \omega_k, \mu, \Sigma)$ .

Da wir die Parameter  $\theta$  nicht kennen, müssen wir sie schätzen oder uns ihnen iterativ annähern. Hierzu bedienen wir uns zu Beginn standardisierter Verfahren beim Arbeiten mit AR Modellen. Wir definieren ein  $t$  und ein  $k$  mit  $t > k$  und

$$\hat{\mu} = \frac{1}{t-k} \sum_{i=k+1}^t x_i \quad (4.23)$$

$$C_j = \frac{1}{t-k} \sum_{i=k+1}^t (x_i - \hat{\mu})(x_{i-j} - \hat{\mu}). \quad (4.24)$$

Dieser Rechenschritt realisiert bereits die erste Annäherung an den Erwartungswert  $\mu$  und die Kovarianzfunktion  $C_j$  unseres AR Modells. Die Koeffizienten  $\omega_1, \dots, \omega_k$  des AR Modells ermitteln wir beim Lösen des linearen Gleichungssystem

$$C_j = \sum_{i=1}^k \omega_i C_{|j-i|} \quad j = 1, \dots, k. \quad (4.25)$$

Da ein Element von der betrachteten Zeitreihe von  $k$  Vorgängern abhängt, behandeln wir hier eine partielle Autokorrelation [17]. Diese Korrelation liefert den Zusammenhang der Kovarianzen und das Gleichungssystem (4.25).

Bezeichnen wir die Lösung von (4.25) mit  $\hat{\omega}_1, \dots, \hat{\omega}_k$  ergibt sich nun auch für die letzte Größe  $\Sigma$  ein Schätzer mit

$$\hat{\Sigma} = C_0 - \sum_{i=1}^k \hat{\omega}_i C_i. \quad (4.26)$$

Bis hierher muss die gesamte Zeitreihe bereits vorliegen. Die Vorgehensweise unterscheidet sich nicht von anderen lernenden Regressions-Modellen. Die Autoren in [43] nennen diese Phase des Algorithmus Initialisierungsphase.

Nach dieser Aufwärmphase wollen wir nun das Namensgebende des *Sequentially discounting AR model learning* SDAR Algorithmus anführen. Wir modifizieren den oben beschriebenen Weg mit einem neuen Parameter beziehungsweise Faktor. In den Gleichungen (4.23), (4.24) und (4.26) multiplizieren wir  $(1 - r)$  auf der rechten Seite. Hiermit erreichen wir, dass bei jedem Update die vergangene Information gewichtet wird. Es gilt, je kleiner der Parameter  $r > 0$  gewählt ist, desto größer gestaltet sich der Einfluss bereits vergangener Datensätze auf das neue Ergebnis.

Nun liegen uns für alle Unbekannten in (4.22) numerische Werte vor und wir bestimmen  $p(x_t | x_{t-k}^{t-1}; \theta) = p_{t-1}$ .

### Implementierung in Matlab

Bei der Implementierung in `Matlab` ist einzig das Lösen des Gleichungssystems (4.25) nicht trivial. Hierfür muss der Vektor  $C$  korrekt in eine Matrix umgeschrieben werden, um anschließend das System mit dem Backslash-Operator zu lösen. Betrachten wir diese Gleichung

genauer und schreiben wir die darin enthaltene Summe für  $k = 3$  aus, können wir die Systemmatrix ablesen

$$\begin{aligned}C_1 &= \omega_1 C_0 + \omega_2 C_1 + \omega_3 C_2 \\C_2 &= \omega_1 C_1 + \omega_2 C_0 + \omega_3 C_1 \\C_3 &= \omega_1 C_2 + \omega_2 C_1 + \omega_3 C_0.\end{aligned}$$

Die Übersetzung in `Matlab`-Code ist hier angedeutet.

Listing 4.3: Gleichungssystem implementiert

```
1 for j = (1:k)
2     C(j) = 1/(t-k) * (x((k+1):t)-mu_0) * (x((k+1-j):(t-j))-mu_0)';
3 end
4
5 for i = (1:(k-1))
6     C_mat(i, (i+1):end) = C(1:(k-i));
7     C_mat((i+1):end, i) = C(1:(k-i));
8 end
9
10 omega = C_mat\C';
```

## Score

Der zu unserer neuen Eingabe  $x_t$  berechnete Wert  $p_{t-1}(x_t)$  liefert uns eine Aussage, ob der neue Wert aus der gleichen Verteilung wie unsere vergangenen Elemente stammt. Diese Tatsache verdeutlichen wir noch mit einem Score, den wir wie folgt berechnen:

$$\text{Score}(x_t) = -\log p_{t-1}(x_t).$$

Wir verwenden eine sogenannte *logarithmic-loss*-Funktion, welche uns einen höheren Score liefert, wenn der Wert nicht derselben Verteilung entstammt. Begründen lässt sich dies mit der Eigenschaft des Logarithmus, für  $x \rightarrow 0$  gegen  $-\infty$  zu streben.

Empirisch ermitteln wir eine Grenze, die uns Ausreißer anzeigt. Bei Tests hat sich gezeigt, dass das 99,5%-Quantil eine geeignete Wahl ist.

### 4.4.3 Ergebnisse und Probleme

Der in Unterkapitel 4.4.1 angeführte Versuchsaufbau führt zu einem Erfolg. Dieser Zugang ermöglicht uns in nahezu allen Fällen eine Fehlfunktion der Heizung eines Raumes zu bestimmen. Die von uns willkürlich eingefügten fehlerbehafteten Messpunkte werden vom SDAR-Algorithmus in den meisten Fällen erkannt. Das Parametertripel  $(t, k, r) = (20, 5, 0.04)$  hat, empirisch ermittelt, die höchste Erfolgsquote. Die Abbildungen 4.14 und 4.15 sind beispielhafte Plots der Scorefunktion. Wir erkennen hier, dass die falschen Werte mit einem viel höheren Score belegt werden als die richtigen. Zugrunde liegt die Zeitreihe, welche wir aus den Simulationsdaten extrahieren. An zufällig gewählten Stellen fügen wir fehlerhafte Daten ein. Wir gehen davon aus, dass Hausverwalter den Fehler sofort nach seinem Auftreten beheben. Eine defekte Isolierung kann nicht binnen eines Heizzyklus repariert werden, aber um Fehler zu simulieren, war dies der beste Weg. Wenn eine Störung über einen längeren Zeitraum auftritt, muss diese System abgeschaltet werden. Aufgrund der Autoregression würde sonst die Verteilung, auf welcher der Score basiert, negativ beeinflusst werden. Sobald das Heizungssystem gewartet ist, können die alten Berechnungen wieder herangezogen werden.

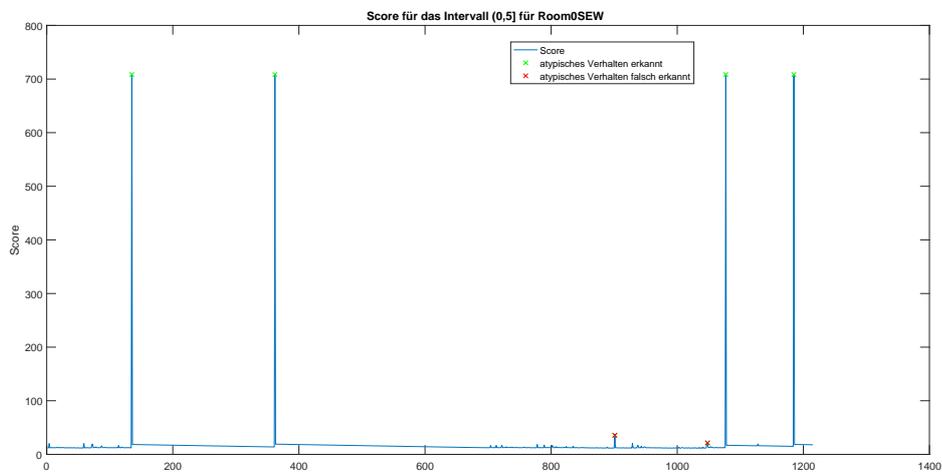


Abbildung 4.14: atypisches Heizverhalten entschieden nach 99,5%-Quantil

Bedingt durch die Abhängigkeit der Heizdauer von der Außentemperatur, benötigen wir für unterschiedliche Temperaturbereiche je eigene Modelle. Solange die Zeitreihen der einzelnen Intervalle lange genug sind, funktioniert unsere Suche nach fehlerhaften Heizvorgängen einwandfrei. Die in den Abbildungen 4.14 und 4.15 als falsch markierten Punkte beschreiben

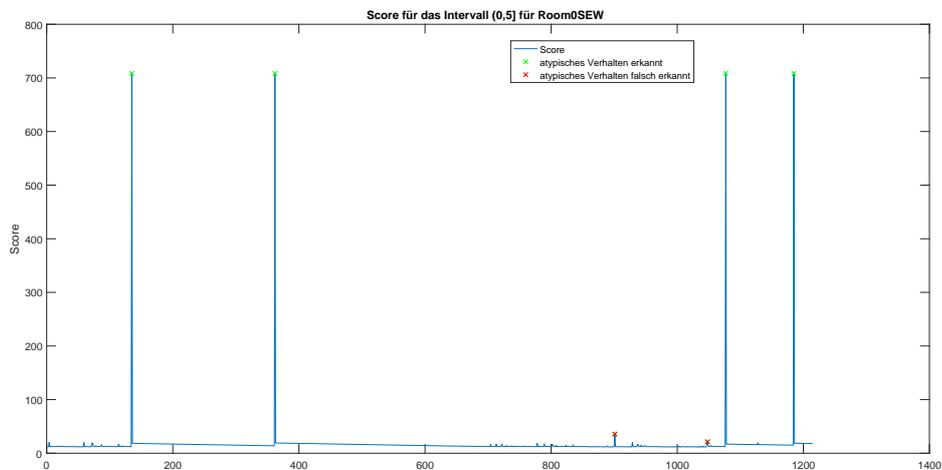


Abbildung 4.15: atypisches Heizverhalten entschieden nach 99,5%-Quantil

korrekte Heizzyklen, die der SDAR-Algorithmus irrtümlich als Fehlfunktion interpretiert. Ebenso sind die eingelernten Modelle nur bedingt auf andere Räume übertragbar, wodurch die Anzahl an eigenständigen Modellen noch erhöht wird. Dieser Teil des Monitoring-System muss für jeden Raum und jeden Temperaturbereich ein eigenes Modell speichern und verwalten. Aufbauend auf der Simulation von Kapitel 3 bedeutet dies  $39 \cdot 6 = 243$  Modelle.

Nichtsdestotrotz liefert dieser vierte Zugang annehmbare Ergebnisse. Wir können einen notwendigen Funktionalität eines Monitoring-Systems, atypisches Heizverhalten zu erkennen, mit dem vorgestellten AR Modell abdecken. Dadurch haben wir gezeigt, dass es möglich ist, basierend auf einer Analyse der Daten Aussagen über die Funktionsweise eines Heizungssystems zu machen. Zusätzlich können wir, falls ein Defekt gemeldet wird, einen Techniker gezielt in einen bestimmten Raum zur Kontrolle und Reparatur schicken.

## 5 Zusammenfassung und Ausblick

Ziel der vorliegenden Arbeit war es, ein funktionstüchtiges Energiemonitoring-System zu implementieren. Im Lauf der Arbeit hat sich herausgestellt, dass für eine umfassende Aufschlüsselung des Eigenverbrauchs weitere Forschungsschritte erforderlich sind. Einen essentiellen Teil dieser Anwendung können wir mit dem SDAR-Algorithmus abdecken. Eine Option, die atypisches Heizverhalten erkennt, erweist sich als unabdinglich in solchen Systemen.

Wenden wir uns zur Verdeutlichung dieser Annahme der Sektion 4.2 zu. Hier beschreiben wir einen Algorithmus, der jeweils die Leistungskurven eines Raumes im Zeitraum, in dem das Thermostat geöffnet ist, aufsummiert und so Rückschlüsse auf die Heizsystemkonstanten zieht. Fehlerhafte Werte der Heizleistung verfälschen unser Ergebnis, daher müssen diese aus der Berechnung ausgegliedert werden, wofür sich wiederum der SDAR-Algorithmus eignet. Im Laufe der Simulation wird sichergestellt, dass dem System keine defekten Heizkörper oder sonstige fehlerhafte Teile angehören, in der Realität jedoch können wir davon nicht ausgehen.

Zukünftige Forschungsarbeiten werden voraussichtlich auf dem zweiten Ansatz aufbauen. Dieser liefert unter Laborbedingungen als einziger ansatzweise die gewünschten Ergebnisse. Um die Resultate realitätsnäher zu gestalten, müssen wir die Schätzung der Systemkonstanten verbessern. Ein möglicher Weg zum Ziel ist die Verlängerung des Beobachtungszeitraums. Für Benützer des Systems würde dies eine längere Wartezeit nach der Installation der Hardware auf den aufgeschlüsselten Energieverbrauch bedeuten. Dieser Verbesserungsansatz ist somit nicht der beste. Ein alternativer Weg ersetzt den von uns gewählten linearen Zusammenhang zwischen Temperatur und Heizleistung durch einen Zusammenhang mit höherem Grad.

Neben den zwei erfolgreichen Ansätzen scheiterten wir bei dem Versuch den Energieverbrauch mit Hilfe eines FHMM aufzuteilen. Im Gegensatz zum Leistungsverlauf bei elektrischen Verbrauchern ist jener der thermischen (Heizkörper) zu ähnlich. Aus dem Gesamtleistungsverlauf kann daher kein Einzelverbrauch abgeleitet werden. Im übrigen Zugang

versuchten wir über die Steigung der Temperaturkurve an bestimmten Punkten Informationen über das Heizungssystem abzuleiten. Die Erfolgsquote der trainierten SVM war jedoch sehr niedrig. Die extrahierten Unterscheidungsmerkmale ließen keinen Rückschluss auf beispielsweise die Anzahl an Außenwänden zu.

Die Aufmerksamkeit des Lesers möchten wir jedoch noch einmal auf das vorgestellte Simulationsmodell lenken. Speziell bei diesem verwenden wir im Vergleich zur Realität einige Vereinfachungen:

- (i) Das Thermostat entspricht nur einem 2-Kanal Regler. In den klassischen Heizungssystemen sind im Gegensatz dazu stufenlose Thermostate verbaut, wodurch die Heizleistung besser geregelt werden kann und die Temperatur konstanter gehalten wird.
- (ii) Eine variable Vorlauftemperatur, die hauptsächlich von der Außentemperatur abhängt, ist ebenso weiter verbreitet als der von uns gewählte konstante Simulation.
- (iii) Äußere Einflüsse wie Wind und Sonneneinstrahlung verändern vorrangig den Wärmegewinn und -verlust über die Außenwände. In unserem Modell berücksichtigen wir diese Komponenten nicht, da sie stark von der Lage und vom Umfelds des Gebäudes abhängen. Mit ausreichender Information über beispielsweise die Beschattung der Fenster besteht die Möglichkeit, nachträglich den Gewinn an Energie einzubeziehen.
- (iv) Wir beachten die Benützung eines Raumes nicht. Abgesehen von einer Heizleistung, die jener der Anwesenheit einer erwachsenen Person entspricht, addieren wir keine derartigen Größen zu unserer Systemleistung. In realen Häusern wechselt die Belegung der Räume, ohne einem Muster zu folgen, wodurch sich die Wärmeflüsse verschieben.

Wir entschieden uns im Vorhinein ganz bewusst für jede der aufgelisteten Änderungen zur Realität. Bei dieser Arbeit handelt es sich um einen ersten Versuch, ein Energiemonitoring zu implementieren. Allein eine stufenlose Thermostatschaltung würde offensichtlich die Komplexität der weiteren Betrachtungen um ein Vielfaches erhöhen.

Wir setzen es uns zum Ziel, in Zukunft einen Algorithmus entwickeln, der neben vorgefertigten Simulationsdaten auch fehlerbehaftete, in der Praxis gemessene Rohdaten verarbeiten kann und zuverlässige Ergebnisse liefert.

## 6 Anhang

Listing 6.1: Alle Konstanten definiert

```
1 %% Define all constants
2 %% Constants for room
3 l = 4;
4 w = 4;
5 h = 3;
6
7 A_floor = l*w;
8 A_wall = l*h;
9
10 c_air = 1009;
11 R_air = 287;
12 p = 101325;
13 V_room = l*w*h*3;
14
15 T_roomIC = 20+273.15;
16
17 Q_internal_heat = 120;
18
19 %% Constants for heater
20 T_inlet_nominal = 273.15+75;
21 T_outlet_nominal = 273.15+65;
22 T_room_nominal = T_roomIC;
23 T_inlet = 50+273.15;
24 T_outlet = 40+273.15;
25 Q_dot_nominal = 100*A_floor;
26 n = 1.24;
27 Q_dot_system = Q_dot_nominal*(logDifference(T_inlet, T_outlet,
      T_room_nominal)) / ...
```

```
28         logDifference ( T_inlet_nominal , T_outlet_nominal
                , T_room_nominal ) ^ n ;
29 UA_heater = Q_dot_system / logDifference ( T_inlet , T_outlet ,
        T_room_nominal ) ^ n ;
30
31 c_water = 4190 ;
32 m = 5 ;
33
34 mdot = Q_dot_system / c_water / ( T_inlet - T_outlet ) ;
35
36 T_set = 21 + 273.15 ;
37 Temp_diff_DayNight = 0 ;
38
39 phi = 2 ; %%Hysteresis
40
41 %% Window/Door
42 A_window = 4 ;
43 k_window = 1.3 ; %%[W/(m^2*K)]
44
45 A_door = 0 ;
46 k_door = 3 ;
47
48
49 %% Wall outside
50 lambda_out = [0.7 0.25 0.8 0.04 0.7 0.7] ; %%[W/(m*K)]
51 d_out = [0.015 0.25 0.004 0.2 0.003 0.002] ;
52
53 k_out = 1 / sum ( d_out ./ lambda_out ) ;
54
55 %% Wall inside
56 lambda_in = [0.7 0.58 0.7] ;
57 d_in ( 1 ) = [0.015 0.365 0.015] ;
58
59 k_in = 1 / sum ( d_in ./ lambda_in ) ;
60
61 %% Ground
```

```
62 lambda_ground = [1.4 0.5 0.044 0.06 2.5 0.035 1.4];
63 d_ground = [0.07 0.0001 0.03 0.07 0.25 0.1 0.1];
64
65 k_ground = 1/sum(d_ground./lambda_ground);
66
67 %% Floor
68 lambda_floor(1) = [1.4 0.5 0.044 0.06 2.5];
69 d_floor = [0.07 0.0001 0.03 0.07 0.22];
70
71 k_floor = 1/sum(d_floor./lambda_floor);
72
73 %% Roof
74 lambda_roof(1) = [0.7 0.5 0.17 0.038 0.17 2.5];
75 d_roof = [0.05 0.0002 0.0003 0.3 0.004 0.22];
76
77 k_roof = 1/sum(d_roof./lambda_roof);
78
79 %% Import data from text file .
80
81 temp = readtable('aussentemperatur.csv','Delimiter',';');
82 temperatureOut = temp.Temperature+273.15;
83 time = 1:3600:3600*length(temperatureOut);
```

Listing 6.2: Logarithmisches Verfahren zur Bestimmung der Übertemperatur

```
1 function [deltat] = logDifference(ti,to,tr)
2
3 if to<=tr
4     deltat = (ti-to)/2-tr;
5 else
6     deltat = (ti-to)/log((ti-tr)/(to-tr));
7 end
8
9 end
```

Listing 6.3: Einlesen der .mat-Dateien

```
1 %% Read in mat-Files
2
3
4 mat = dir('*.mat');
5
6 load(mat(1).name)
7 room = {strrep(mat(1).name, '.mat', '')};
8 time = export(1,:);
9 temperature = export(2,:) - 273.15;
10 power = export(3,:);
11 energy = export(4,:) ./ 3600. / 1000;
12 onOff = export(5,:);
13
14
15
16 for i=2:length(mat)
17
18     load(mat(i).name)
19     room{i} = strrep(mat(i).name, '.mat', '');
20     temperature = [temperature; export(2,:) - 273.15];
21     power = [power; export(3,:)];
22     energy = [energy; export(4,:) ./ 3600. / 1000];
23     onOff = [onOff; export(5,:)];
24
25 end
26
27 energy = [energy; sum(energy)];
28 room{40} = 'Sum';
29
30 time = datetime(2015,09,1)+seconds(time);
```

# Abbildungsverzeichnis

1.1	Wärmezähler der Firma <i>Techem</i> . . . . .	2
3.1	Schematische Raumanordnung eines Stockwerks . . . . .	10
3.2	Wärmefflüsse im Modell . . . . .	11
3.3	Eingabemaske für eine Wand . . . . .	15
3.4	Modell eines Raumes . . . . .	16
3.5	Modell eines Heizkörpers . . . . .	16
3.6	Plot des Summenzählers und der Außentemperatur . . . . .	20
3.7	Simulationsergebnis für einen Tag . . . . .	21
3.8	Ergebnisse des Raumes in der Mitte über drei Jahre . . . . .	22
3.9	Ergebnisse des Raumes NEW über drei Jahre . . . . .	22
4.1	Beispiele für thermische Leistungsverläufe . . . . .	34
4.2	Elektrische Leistungsverläufe eines Kühlschranks und eines Herds entnommen dem ECO-Dataset [5] . . . . .	35
4.3	Beispiel einer Summe der Intervalle eines Raumes . . . . .	37
4.4	Summe der Heizintervalle aller Räume eines Stockwerks mit unterschiedlicher Anzahl $m$ . . . . .	39
4.5	Evolution der summierten Intervalle eines Raumes . . . . .	40
4.6	Energieverläufe aller Räume . . . . .	41
4.7	Vergleich zwischen berechneten und simulierten Energieverlauf . . . . .	41
4.8	Relativer Fehler eines Raumes am Ende der Berechnungszeitspanne . . . . .	42
4.9	Temperaturkurve einer Raumes . . . . .	45
4.10	Punktwolke, die im zweiten Ansatz klassifiziert wird . . . . .	46
4.11	Margin [6] . . . . .	48
4.12	Geometrische Überlegungen zur Gleichung (4.10) [6] . . . . .	48
4.13	Clustering mit einer SVM . . . . .	53
4.14	atypisches Heizverhalten entschieden nach 99,5%-Quantil . . . . .	59
4.15	atypisches Heizverhalten entschieden nach 99,5%-Quantil . . . . .	60

# Tabellenverzeichnis

2.1	Vergleich der Algorithmen [46]	8
3.1	Außenwandsystem 4: HLZ 25 + WDVS	18
3.2	STB Decke	18
3.3	Erdanliegender Fußboden: Wärmedämmung in FB-Aufbau	18
3.4	Warmdach: Stahlbetondecke (Flachdach)	18
4.1	Room0SW mit den Nachbarn (im)	44
4.2	Room0SW	46
4.3	Ergebnisse des Clustering nach dem zweiten Ansatz	53

## Literaturverzeichnis

- [1] Abdul Afram and Farrokh Janabi-Sharifi. Theory and applications of hvac control systems—a review of model predictive control (mpc). *Building and Environment*, 72:343–355, 2014.
- [2] Gerold Alsmeyer. *Stochastische Prozesse*. Inst. für Math. Statistik, 2000.
- [3] Fabrizio Ascione, Francesca Ceroni, Rosa Francesca De Masi, Filippo de’Rossi, and Maria Rosaria Pecce. Historical buildings: Multidisciplinary approach to structural/energy diagnosis and performance assessment. *Applied Energy*, 185:1517–1528, 2017.
- [4] N Aste, P Scudo, R Fedrizzi, C Del Pero, and F Leonforte. Energy retrofit of residential buildings: A multifunctional toolkit. In *Clean Electrical Power (ICCEP), 2017 6th International Conference on*, pages 62–67. IEEE, 2017.
- [5] Christian Beckel, Wilhelm Kleiminger, Romano Cicchetti, Thorsten Staake, and Silvia Santini. The eco data set and the performance of non-intrusive load monitoring algorithms. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, pages 80–89. ACM, 2014.
- [6] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.
- [7] Bing Dong, Cheng Cao, and Siew Eang Lee. Applying support vector machines to predict building energy consumption in tropical region. *Energy and Buildings*, 37(5):545–553, 2005.
- [8] Christof Eck, Harald Garcke, and Peter Knabner. *Mathematische Modellierung*, volume 2. Springer, 2008.
- [9] Varick L Erickson, Yiqing Lin, Ankur Kamthe, Rohini Brahme, Amit Surana, Alberto E Cerpa, Michael D Sohn, and Satish Narayanan. Energy efficient building environment control strategies using real-time occupancy measurements. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 19–24. ACM, 2009.

- [10] J-L Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE transactions on speech and audio processing*, 2(2):291–298, 1994.
- [11] Zoubin Ghahramani and Michael I Jordan. Factorial hidden markov models. In *Advances in Neural Information Processing Systems*, pages 472–478, 1996.
- [12] M Amac Guvensan, Z Cihan Taysi, and Tommaso Melodia. Energy monitoring in residential spaces with audio sensor nodes: Tinyears. *Ad Hoc Networks*, 11(5):1539–1555, 2013.
- [13] George William Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.
- [14] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Overview of supervised learning. In *The elements of statistical learning*, pages 9–41. Springer, 2009.
- [15] Genshiro Kitagawa and Will Gersch. *Smoothness priors analysis of time series*, volume 116. Springer Science & Business Media, 2012.
- [16] J Zico Kolter and Tommi Jaakkola. Approximate inference in additive factorial hmms with application to energy disaggregation. In *Artificial Intelligence and Statistics*, pages 1472–1482, 2012.
- [17] Jens-Peter Kreiß and Georg Neuhaus. *Einführung in die Zeitreihenanalyse*. Springer-Verlag, 2006.
- [18] Harold W Kuhn. Nonlinear programming: a historical view. In *Traces and emergence of nonlinear programming*, pages 393–414. Springer, 2014.
- [19] P. Kurzweil, B. Frenzel, and F. Gebhard. *Physik Formelsammlung: Für Ingenieure und Naturwissenschaftler*. Studium Technik. Vieweg+Teubner Verlag, 2007.
- [20] Berkeley Lab. Modelica buildings library. Website, 2017. Online einsehbar [http://simulationresearch.lbl.gov/modelica/releases/v3.0.0/help/Buildings\\_Examples\\_Tutorial\\_Boiler.html#Buildings.Examples.Tutorial.Boiler](http://simulationresearch.lbl.gov/modelica/releases/v3.0.0/help/Buildings_Examples_Tutorial_Boiler.html#Buildings.Examples.Tutorial.Boiler); abgerufen am 01.07.2017.
- [21] Jian Liang, Simon KK Ng, Gail Kendall, and John WM Cheng. Load signature study—part i: Basic concept, structure, and methodology. *IEEE transactions on power Delivery*, 25(2):551–560, 2010.

- [22] Gu-yuan Lin, Shih-chiang Lee, Jane Yung-jen Hsu, and Wan-rong Jih. Applying power meters for appliance recognition on the electric panel. In *Industrial Electronics and Applications (ICIEA), 2010 the 5th IEEE Conference on*, pages 2254–2259. IEEE, 2010.
- [23] Max A Little and Nick S Jones. Sparse bayesian step-filtering for high-throughput analysis of molecular machine dynamics. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 4162–4165. IEEE, 2010.
- [24] Sunil Mamidi, Yu-Han Chang, and Rajiv Maheswaran. Improving building energy efficiency with a network of sensing, learning and prediction agents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 45–52. International Foundation for Autonomous Agents and Multiagent Systems, 2012.
- [25] Alan Marchiori, Douglas Hakkarinen, Qi Han, and Lieko Earle. Circuit-level load monitoring for household energy management. *IEEE Pervasive Computing*, 10(1):40–48, 2011.
- [26] Claudio Martani, David Lee, Prudence Robinson, Rex Britter, and Carlo Ratti. Ener-net: Studying the dynamic relationship between building occupancy and energy consumption. *Energy and Buildings*, 47:584–591, 2012.
- [27] MONICA MATEO-GARCIA, ABDULLAHI AHMED, and DANNY MCGOUGH. Non-invasive approaches for low-energy retrofit of buildings: Implementation, monitoring and simulation in a living lab case study. *Structural Studies, Repairs and Maintenance of Heritage Architecture XV*, 171:183, 2017.
- [28] Mathworks. Choose classifier options - support vector machine. Website, 2017. Online einsehbar <https://de.mathworks.com/help/stats/choose-a-classifier.html#bunt0n0-1>; abgerufen am 13.10.2017.
- [29] Daniel Prusa and Tomas Werner. Universality of the local marginal polytope. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1738–1743, 2013.
- [30] Lawrence Rabiner and B Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.
- [31] Paula Rocha, Afzal Siddiqui, and Michael Stadler. Improving energy efficiency via smart building energy management systems: A comparison with policy measures. *Energy and Buildings*, 88:203–213, 2015.

- [32] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.
- [33] Anthony Schoofs, Antonio Guerrieri, Declan T Delaney, Gregory MP O’Hare, and Antonio G Ruzzelli. Annot: Automated electricity data annotation using wireless sensor networks. In *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*, pages 1–9. IEEE, 2010.
- [34] Pervez Hameed Shaikh, Nursyarizal Bin Mohd Nor, Perumal Nallagownden, Irraivan Elamvazuthi, and Taib Ibrahim. A review on optimized control systems for building energy and comfort management of smart sustainable buildings. *Renewable and Sustainable Energy Reviews*, 34:409–429, 2014.
- [35] D Srinivasan, WS Ng, and AC Liew. Neural-network-based signature recognition for harmonic source identification. *IEEE Transactions on Power Delivery*, 21(1):398–405, 2006.
- [36] Michael Sturm. Simulation einer gebäudeheizung. *Technische Dokumentation*, 1998.
- [37] Kosuke Suzuki, Shinkichi Inagaki, Tatsuya Suzuki, Hisahide Nakamura, and Koichi Ito. Nonintrusive appliance load monitoring based on integer programming. In *SICE Annual Conference, 2008*, pages 2742–2747. IEEE, 2008.
- [38] Athanasios Tsanas and Angeliki Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings*, 49:560–567, 2012.
- [39] Fachverband Stein und keramischen Industrie Initiative Ziegel. Wärmebrücken-katalog im ziegelmassivbau. Website, 2015. Online erhältlich unter [http://www.zieglerverband.at/waermebrueckenkatalog\\_gesamt.pdf](http://www.zieglerverband.at/waermebrueckenkatalog_gesamt.pdf); abgerufen am 01.07.2017.
- [40] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.
- [41] Fangfei Weng, Qingshan Jiang, Lifei Chen, and Zhiling Hong. Clustering ensemble based on the fuzzy knn algorithm. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on*, volume 3, pages 1001–1006. IEEE, 2007.

- [42] G Wood and M Newborough. Influencing user behaviour with energy information display systems for intelligent homes. *International journal of energy research*, 31(1):56–78, 2007.
- [43] Kenji Yamanishi and Jun-ichi Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 676–681. ACM, 2002.
- [44] Michael Zeifman and Kurt Roth. Nonintrusive appliance load monitoring: Review and outlook. *IEEE transactions on Consumer Electronics*, 57(1), 2011.
- [45] Tehseen Zia, Dietmar Bruckner, and Adeel Zaidi. A hidden markov model based procedure for identifying household electric loads. In *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society*, pages 3218–3223. IEEE, 2011.
- [46] Ahmed Zoha, Alexander Gluhak, Muhammad Ali Imran, and Sutharshan Rajasegarar. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, 12(12):16838–16866, 2012.