Institute of Architecture of Application Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit

A Predictive Control System for Indoor Lighting

Mirac Öztürk

Course of Study: Softwaretechnik

Examiner: Prof. Dr. Marco Aiello

Supervisor: Prof. Dr. Marco Aiello

Commenced: November 8, 2021

Completed: May 9, 2022

Abstract

Modern lighting solutions have become smart due to saving more energy and advancing technology like introducing light-emitting diodes (LEDs). LEDs offer many new possibilities on how to interact with lighting in order to provide the opportunity for smart lighting systems. Smart lighting systems combined with sensors automate the lighting process and can go beyond illumination, affect productivity, well-being, and emotions. Multiple applications exist in smart cities, warehouses, and even residentials. This Bachelor Thesis aims to examine current methods and to create a predictive control system for indoor lighting using smart lights from Philips Hue in combination with a Raspberry Pi and multiple sensors (motion, temperature, and lighting). We will also take a look at the technical and methodical background. For the experiment, we will collect user data and use it with a machine learning algorithm (Very Fast Decision Tree) to predict lighting in a real-life environment. At the end of the research, we are going to evaluate the system's accuracy by measuring the decisions with time metrics. After a sensitivity analysis, we can say that the evaluated system can make some inaccurate predictions due to changed environmental values like longer daytime and different weather status (e.g., less cloudy weather).

Contents

List of Figures

List of Tables

List of Listings

1 Introduction

In recent years, the greenhouse effect and the subsequent environmental damage which were caused by global carbon dioxide emissions as well as the increase of electricity production and consumption, there were new technical solutions in which the traditional lighting system became "smart". The rapid changes in the technological world caused new problems and challenges which exceed the illumination of lights. Introducing the light-emitting diodes (LEDs) opened the door for new functionalities like dimming and changing the light‘s color. A further functionality was the automatic control with additional sensors. Analyzing the environment give rise to challenges like energy savings, convenience, system automation, effects on human well-being, productivity, and more. Since artificial lighting plays a huge role in our daily lives (work or private), it makes sense for the environment to reduce energy costs because of the rising energy prices. For instance, in a warehouse, lighting consumes 65% of the total energy costs. The latest smart lighting systems bring a bunch of potential use cases in other areas rather than serving as an artificial light source. They can complement communication networks, serve as data transmitters, provide indoor positioning functionalities as well as improve human well-being. Typical use cases of smart lighting systems are street lighting in the context of smart cities and lighting in buildings or "smart" homes. The challenge of smart lighting systems is to adapt their behavior to the lighting system‘s environment with autonomous behavior [FGG21]. For the user of smart home solutions, it is not enough for the system to be energy efficient, but also to be convenient. The classification of modern smart home solutions can be either done by sensor-based or directive-based solutions. For the directive-based solution, the user can control the light with the help of various stationary- and remote techniques using cloud‘s services or mobile devices. However, this kind of solution requires user input and with the increasing number of home devices, the effort of the user does also increase. On the other hand, with the help of sensor-based solution, the home of the user is equipped with sensors that measure different physical parameters (e.g. temperature or luminosity). This system proves to be much more convenient for the user because the smart home system‘s behavior adjusts to the measured physical parameters, e.g. the light turns on whenever the sensors measure motion as well as a little luminosity. This classification is typically called the user behavior approach [VMP13].

There are smart lights like the products from the Philips Hue series. This Bachelor Thesis aims to create a predictive control system for indoor lighting using smart lights combined with a Raspberry Pi, a PIR motion sensor, a light sensor, a temperature sensor, and machine learning algorithms using a python script. The service will collect data for this research and use it for the machine learning algorithm. The system’s goal is to correctly predict the state of the smart lights by measuring the environment with its sensors and turning the lights either on or off, being able to adjust it as close as possible to the user’s behavior. For this, the service will be evaluated after some metrics.

The Bachelor Thesis starts with the State of the Art of smart lighting systems (section 2), giving some general information about smart lighting systems and then explaining real-life applications of smart lighting systems. Furthermore, we will dive into some background information (section 3) about machine learning, Raspberry Pi, and Philips Hue that will be important for the experiment and the research. Then, we will look at the Design (section 4) and Implementation (section 5) of the experiment, followed by the Evaluation (section 6). Finally, I will draw a conclusion about the state of smart lighting systems in the context of human behavior prediction and the future of smart lighting systems (section 7).

2 State of the Art

Lighting can be classified as natural and artificial light. In this paper, we are going to focus on the latter. Comparing the traditional lighting systems to smart lighting systems, the first consists of more than one switch which turns the light on and controls the brightness. Smart lighting systems can be applied to every application where light is required. There are use cases like interior lighting in offices or homes and outdoor lighting in public streets. Smart lighting systems provide better light quality and energy efficiency because there is also progress in lighting itself with new technologies by the likes of light-emitting diodes (LEDs). LEDs enable new possibilities, e.g. adapting the light to the user's needs (changing the intensity or the color) or even automatic control with additional sensors. Using smart lighting systems is environment friendly because it lowers light pollution. Light pollution is the wasted energy caused by bad lighting design. A bad lighting design can be explained through the user‘s excessive use of artificial light source. Lighting design is not enough for the user's acceptance. The smart light systems must offer utility and the possibility of integrating new components in existing systems. Another positive aspect for the user is to implement smart lighting systems to regulate its circadian rhythm. According to the human-centric lighting (HLC) approach, smart lighting systems can improve the human‘s well-being [FGG21].
The introduction of LEDs and their technological progresses have created new possibilities in the world of lights like the improvement of light control, the efficient dimming techniques in comparison with traditional lighting, and it is the fundament of modern lighting systems. While traditional lighting sources had issues with coolants and lubricants combined with dust burning because of heat radiation, the problem does not arise with LEDs. The absence of heat radiation occurs with lower maintenance and replacement cost. It is possible to integrate other light sources except of LEDs with smart lighting systems, but they cannot satisfy all requirements. Although LEDs are a huge part of smart lighting systems, they do not qualify as smart lighting systems according to the authors of [FGG21]. However, they are the technical fundament of smart lighting systems and they are used in almost every kind of smart lighting system. There are some applications of lighting systems with daylight- and motion sensors. Daylight sensors are utilized in places where they can be combined with natural lighting sources. The artificial lighting only complements natural lighting when it is required. Motion sensors are used in rooms or places which are not crowded. Therefore, the room‘s lighting is barely used. Due to the use of the motion and lighting sensors, large energy savings were reported in studies. With motion sensors, the energy savings range between 17% and 60%. Moreover, the energy savings with daylight sensors are approximately 40%. The combination of the two sensors result in energy savings ranging between 13% and 73.2%. Smart lighting systems can also be controlled by network systems. Network systems enable central monitoring and controlling of light [FGG21].

The development of lighting will allow us to observe our environment in order to accomplish tasks. It will also provide other functionalities like improving productivity, affecting people's mood and well-being. If lighting offers functionalities beyond illumination (e.g., emotional effects), fully manual user control is not desirable. On the other hand, a fully automatic lighting system is also not desirable because of the inevitable mistakes and wrong decisions that the system could eventually make. Therefore, the goal should be a hybrid system, including user control and system control systems. Hybrid system means that it will act autonomously to a certain degree and learn new behavior through user interaction. There are potential prospects from experience and functional perspectives. One of the experience perspectives is that autonomous systems will control numerous light sources. It is impossible to control each LED of a lighting platform due to the complexity and the amount of LEDs. The system has to program the lights' location, shape, size, and intensity for every appropriate moment. Another experience perspective is to add additional light features, e.g. light features which influence people. This sort of light features require more complexity for total user control. Finally, functional experiences are significant energy savings resulting from smart lighting systems, and the installation and configuration of smart lighting systems. A "plug and play" approach makes the installation easier. The interaction with smart lighting systems means that the relationship between user and system can range between the following ambiguous types:

- switch/select presets
- dialogue (user and system interact with each other to find the best option)
- direct manipulation from the user

The user often controls the on/off function which is present in most of the lighting systems. Providing the additional control of the light behavior could lead to dynamic atmospheres that fit to the user's current situation. The highest level of interaction between the user and system would be the awareness of the user's context (e.g., reading) By being aware of the user‘s context, the system can select and apply the appropriate light [EOE12].

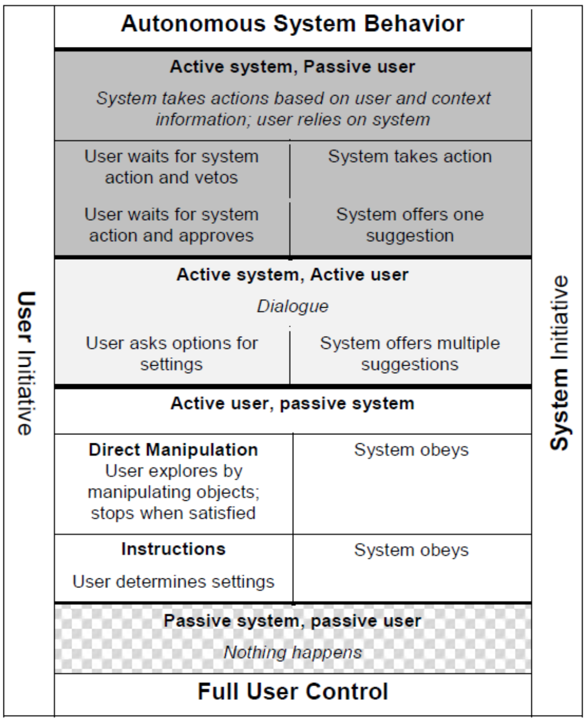

Figure 2.1: Types of Interaction between user and system [EOE12]

Figure 2.1 describes different types of interaction and behavior between the user and the system. For example, if there exists an autonomous system behavior, the user will be entirely passive and dependent on the system when an action happens. An approach for this system is the use of Multi-Agent Systems. An agent in this context represents either light sources, sensors, or people. These agents interact with their environment and turn information back. These systems are also denoted as decentralized systems because a single agent cannot make the decisions. Machine Learning algorithms benefit from Multi-Agent Systems because they can learn and represent the desired behavior from the interactions with the agents [EOE12].

There are several types of standards in home automation. The first one is coined as the Simple Bus standard which was also one of the initial proposals for home automation. Every device is connected to the same bus in a closed network. This standard cannot be extended, and its scalability is limited. The primary use of the Simple Bus standard is to control lights and window shades. Furthermore, this standard is ideal for smaller applications. The next one is the Closed Centralized standard which introduces a controller that connects with the devices. This standard often uses client-server as its topology. Similar to the Simple Bus standard, the protocol between the controller and the devices is closed. The Closed Centralized standard is typical for companies who want to sell all of their products to the user because he or she can only access the devices over the controller, making it a monopoly for the companies. In addition, this standard has its limits when it comes to scalability and it is not the best solution for dealing with heterogeneity. Another standard is the Open Server-Based hierarchy, which uses public standards for communication. A central server (usually a PC) is connected to all devices. Furthermore, a controller is introduced to connect sensors and actuators with the central server. They can not directly connect with the central server. Finally, the central server is connected with the internet to enable remote control for the user. This system has a great scalability because it can easily add and remove devices. The only drawback is the central server because it can cause a single point of failure when it is not working. But there is also the Web Service P2P architecture to avoid the single point of failure. The difference in contrast to the Open Server-Based hierarchy is to use a peer-to-peer architecture instead of a centralized unit. Therefore, there are multiple gateways to connect with the devices. The Web Service P2P architecture also enhances the scaling [AD08].

2.1 Smart Lighting in public streets

An intelligent lighting system was installed in a light traffic route to change the lighting behavior and to add new functionalities. The system's design should save energy and is required to give the user a unique experience while using the system. In addition, lighting and its infrastructure are located in areas where people and activities are situated in, to make collecting usage data more accessible for the system in order to develop. The system detects the person and even the direction of the movement to correctly adjust to the lighting behavior. Because the system provides the user a unique experience as well as it is developed to save energy, it has higher user acceptance. Traditionally street lighting is controlled by clock or daylight sensors. However, for an advanced control like dimming the lights, traditional lighting sources did not offer any possibilities to do that. Changing traditional lighting sources with LED lights led to modern developed street lighting systems. In the modern developed system, there are 28 control modules installed to gain information in order to change the light for different use conditions. For light systems in public streets, simple on-off functionality is not acceptable for safety and aesthetic reasons. The developed system uses PIR (passive infrared) sensors to detect human bodies and their movement because the PIR sensors are able to easily detect human bodies with infrared radiation. In addition, PIR sensors are often integrated into lighting because of their low cost, small size, and functionality to detect humans. Another aspect of PIR sensors is that there is no need for lighting and very little computation for the detection. Furthermore, PIR sensors detect the thermal features of an object, which makes these sensors more likely to use in many applications to avoid privacy concerns. However, detecting thermal features is also a drawback for PIR sensors because of other sources of thermal energy such as temperature changes or an engine. Since the sensors react to temperature changes, the objects need either to move or the sensor must use techniques by the likes of infrared radiation‘s chopping or sensor rotation. Another option for movement detection is vision-based camera recognition. However, compared to PIR sensors, vision-based camera recognition is more expensive. Vision-based camera recognition depends on the surrounding‘s illumination, making the image recognition process require a lot of capacity. People who does not like camera installations and who potentially violate their privacy were also another reason for not installing these cameras in the street lighting system. The lighting system used adaptive lighting control, dimming all the lights to a low level.When a person entered the road, the lighting system has brightened the essential lights. Furthermore, after the person passed the route, the system dimmed the lights again. The lighting system also brightened the lights in front of the user. The developed system’s motion sensor consists of a microcontroller unit and PIR elements. The microcontroller unit received the computed data from the PIR sensors and sent it via mesh network to a central gateway. The central gateway processed all the data from all motion sensors and addressed to the necessary microcontroller units, which can control the lighting. Therefore, the gateway can change the lighting of many units parallel. The PIR elements of the motion sensor consist of three PIR sensors. Two of these sensors were responsible for detecting moving objects from a long distance. The detection from a long distance was necessary because the sensor had to detect objects moving either towards or against the sensor. The third PIR sensor was in the middle of the other PIR sensors and it was responsible for short-distance detection. Since three different PIR sensors were pointing at different directions, it was possible to observe different scenarios. When any motion was detected, the microcontroller unit started a detection window that lasted for 10 seconds. It recorded all PIR sensors‘ events and sent them to the gateway [JSE+18].

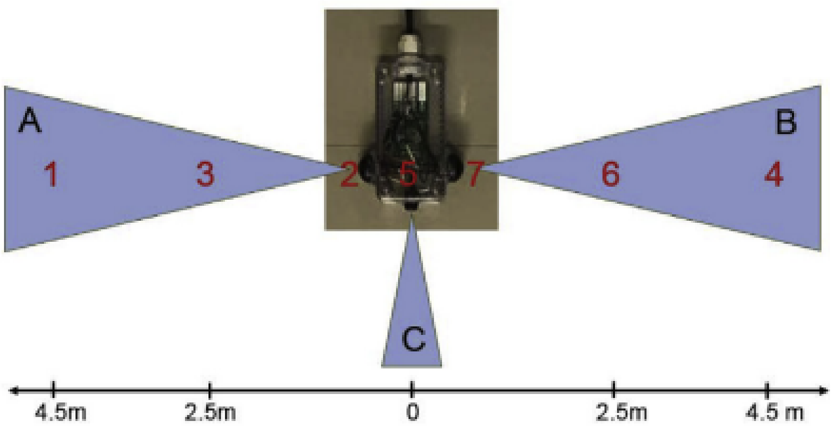

Figure 2.2: PIR elements of motion sensor [JSE+18]

The authors of [Bha16] introduced another light system for street lighting, where the main goal was low power consumption. The sensors, which were used in this system, were LDR (light depending resistor) sensors to measure the sunlight intensity and motion sensors. The general idea is to identify day/night conditions in order to switch the light on/off with the LDR sensors and to change the intensity of the lights depending on whether any motion gets detected with the motion sensors [Bha16].

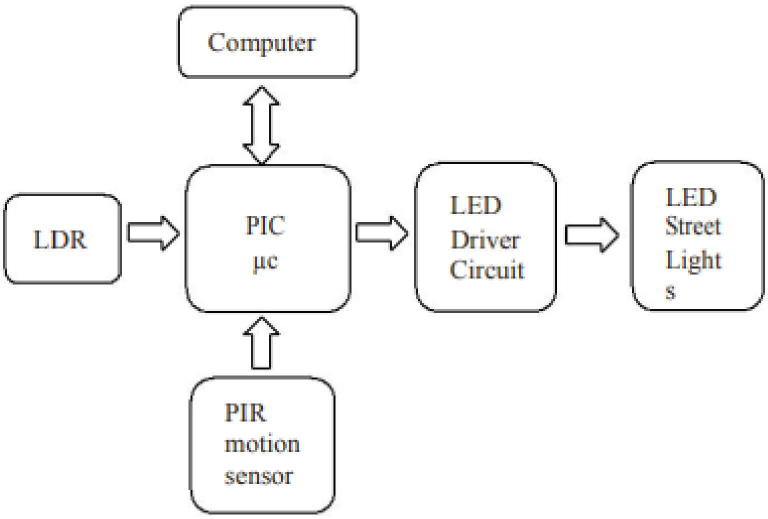

Figure 2.3: Block Diagram Smart Street Lighting [Bha16]

Figure 2.3 depicts a block diagram of the mentioned system, including a PIC microcontroller, LDR sensors, PIR motion sensors, a computer, and the LED driver. The LDR sensor threshold depends on the sunshine which will amount to a minimum when there is enough sunlight and amount to a maximum when the street is dark. The user can set any value to the threshold. The PIC microcontroller turns the lights on when the threshold reaches the maximum. When there is no motion on the street within a user-defined time limit, the lights get dimmed to a minimum value with a PWM (Pulse with Modulation) pulse. The motion sensors can detect up to the range of seven meters. The PIC microcontroller will brighten the lights to their maximum if any motion is detected, this happens again with a PWM pulse [Bha16].

2.2 Smart Lighting in warehouses

The authors of [FGG21] propose a smart lighting system in a warehouse, which appears specifically in order picking. Order picking means that one finds items in the warehouse‘s storage location. They choose warehouse order picking because this task is manually performed in many companies, and there are large parts in a warehouse which need lighting. Besides energy savings, the goal of the propositions was also to improve the light conditions and to improve the work quality. Bad lighting can cause a slower work rate, error-prone, and risk for accidents. The manual order picking process requires efficient routing through warehouses. Therefore, a smart lighting system can improve this process with the following propositions.

The first proposition's primary goal is to increase energy savings in the warehouse. The size of warehouses vary between 144m^2 and 76.600m^2. More than a half of the warehouses are bigger than 10.000m^2 which results in the illumination of large surfaces. Changing traditional lighting sources to LEDs can save more than 50% of the energy. The combination between light and motion sensors can also contribute to energy savings. With the proposed smart lighting system, the light sensor can control the lighting intensity depending on how much daylight illuminates the warehouse and complements it to fulfill recommended lighting value at any time. Another problem with traditional lighting is the illumination of the warehouse‘s large unused areas. With motion sensors, the lights in unused areas can be dimmed and brightened when someone enters the area again.

The steps of warehouse order picking are the following: setup, travel, search and pick. These four steps depend on how the items are stored in the warehouse. A smart lighting system can influence the storage in warehouses. As mentioned in the first proposition, there are large, infrequently used areas in warehouses. Warehouse managers can put frequently asked items together. They can also restructure incoming orders to reduce worker travel to infrequently used areas and contribute more to energy savings, maximizing certain aisles' use.

Picking wrong items and the incorrect number of items or skipping items from an order are all pick errors in warehouses. A smart lighting system cannot eliminate this problem, but it can help to reduce it because one of the pick error‘s sources is bad lighting at warehouses. Bad lighting makes it difficult for workers to read the order and items' information. Workplace regulations define a lighting requirement, and in Germany, the requirement for reading tasks is only 200 lux, but this value should be at least three or four times higher. Smart lighting systems come up to this requirement because they can offer higher light intensity than traditional lighting sources.

Smart lighting systems can improve the well-being of humans and regulate their circadian rhythm which means that smart lighting systems can also enhance the well-being of warehouse workers. LEDs can adjust to their wavelengths and colors temperatures. Therefore, the light can adapt to the workers' needs by influencing their moods, motivation, and productivity. Higher productivity and motivation result as well in fewer pick errors. In summary: smart lighting systems do not only contribute to energy savings but can improve processes by having different functionalities combined with sensors [FGG21].

2.3 Smart Lighting in Residentials

A smart lighting system was introduced to office buildings to save energy and to improve the lighting conditions for the occupants. Usually, the lights are connected with a motion sensor in modern office buildings. When there is no motion after a certain time, the lights turn off. This technique can save energy, but there are other problems. For example, the occupants always have to make a motion that the lights do not turn off, and moreover, not every activity requires lighting. It can also be the case that there is enough natural light. The goal of this system was to provide the occupants a genuinely immersive experience by letting the system understand the current activity and to react accordingly. The proposed system used Artificial Intelligence (AI) planning, particularly the Hierarchical Task Network (HTN) planning. This technique provides a good solution for looking for the "best" action. Additionally, it is easy to adapt, which is also suitable for changes in environments [GNN+17].

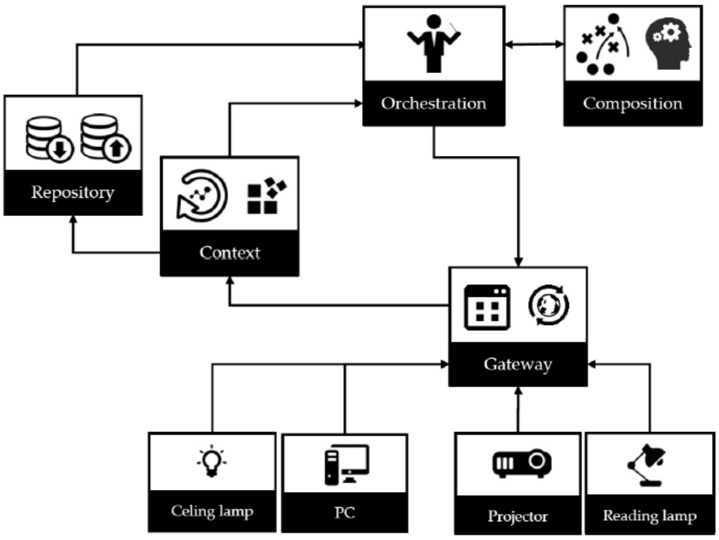

Figure 2.4: Architecture design of the activity recognition service [GNN+17]

Figure 2.4 shows the architecture design of the system. At the bottom level there are the devices and gateway, which collect raw data about the building's environment. Sensors are also part of the devices besides the PC, lamps, and the projector. The system‘s sensors consists of light, temperature, motion, and pressure sensors. Additionally, a microphone is used as well. The gateway is the component where the system interacts with the devices. Furthermore, the collected data from the devices and from gateway arrive at the context component. The context component will store all data in the repository and monitor the state of each device. The last part includes the orchestration and the composition. The composition uses HTN planning to create a new plan and to solve problem as accurate as possible. When the orchestration finds a plan, it executes it by communicating with the devices through the gateway. The developed system was tested in an office area and the building‘s restaurant. The system was tested in three rooms of the office area for three days. With the sensors, they could anticipate the user‘s action. Furthermore, they simulated the natural lighting by gaining information about the weather condition and mapping these into fixed lux values. The artificial lighting adjusts to the lux values. The system detected whether the user was working on the PC and adjusted the lighting appropriately to the desired lux value. The system saved energy up to 75.5% and showed success rates of 80.85%, 76.35%, and 99.17% for the different rooms. The number of wrong recognition was 230, 284, and 56. For the experiment in the restaurant, they installed motion and light sensors. The light sensors measured the natural lighting, and the motion sensors detected people in the restaurant. They divided the restaurant space into areas to use the motion sensors‘ information more precisely. The system collected data for three weeks and worked on automated control after the collection of data for two weeks. During automated control, the option of manually controlling the lights was disabled. The energy savings compared to manual control were 89% [GNN+17].

Another instance of smart lighting was used in a smart home for a disabled person, where the system adjusted to the natural lighting which was measured by the light sensors. The system depended on the user‘s input. The user's action was an indicator for the user‘s room where he or she was present. Therefore, the system could decide whether lighting was necessary depending on the sensor's data.

For example, when the user decided to go to sleep, the system turned off the lights, and when the user wanted to wake up, the system checked the environment whether it was necessary to turn on the lights. This sophisticated system used a pretty similar architectural structure which was shown in 2.4 [KWLA13].

3 Background

In this chapter, I will elaborate on my thesis' technical and methodical background. First, I will briefly explain the methodical background as well as provide an understanding of what Machine Learning is. Moreover, I will introduce the different types of Machine Learning and show how to calculate the accuracy of Machine Learning algorithms. The second part will be about the hardware which I used in the experiment, e.g. the Raspberry Pi and the Philips Hue lamps.

3.1 Machine Learning

Machine Learning algorithms are designed to emulate human intelligence and learn from their surrounding environment. These kinds of algorithms are not designed for a specific task. Instead, they adapt to the continuous input data which results in experience for the algorithms and they achieve better results in the task. The adaptation process starts with the training of the algorithm where they get labeled data, i.e. input data with the desired outcome. During the training, the algorithm can autonomously configure itself in order to compute the outcome with unlabeled data i.e. data without the outcome. The adaptation process does not end after the training has been completed. If you have a good algorithm, it will perform "lifelong" adaptation. Machine Learning occurs in three different types: supervised, unsupervised and semi-supervised Machine Learning. These types are divided according to the labeling of their data. In supervised learning, there is labeled data to compute unknown outcomes from new unseen data (e.g. classification and regression). Unsupervised learning algorithms work with unlabeled data i.e. that the input data is given. Unsupervised learning has to find the outcome on its own. Combining supervised learning and unsupervised learning results in semi-supervised learning. In this technique, the input data is partially labeled. The labeled data is used to compute the outcome of the unlabeled data [EM15].

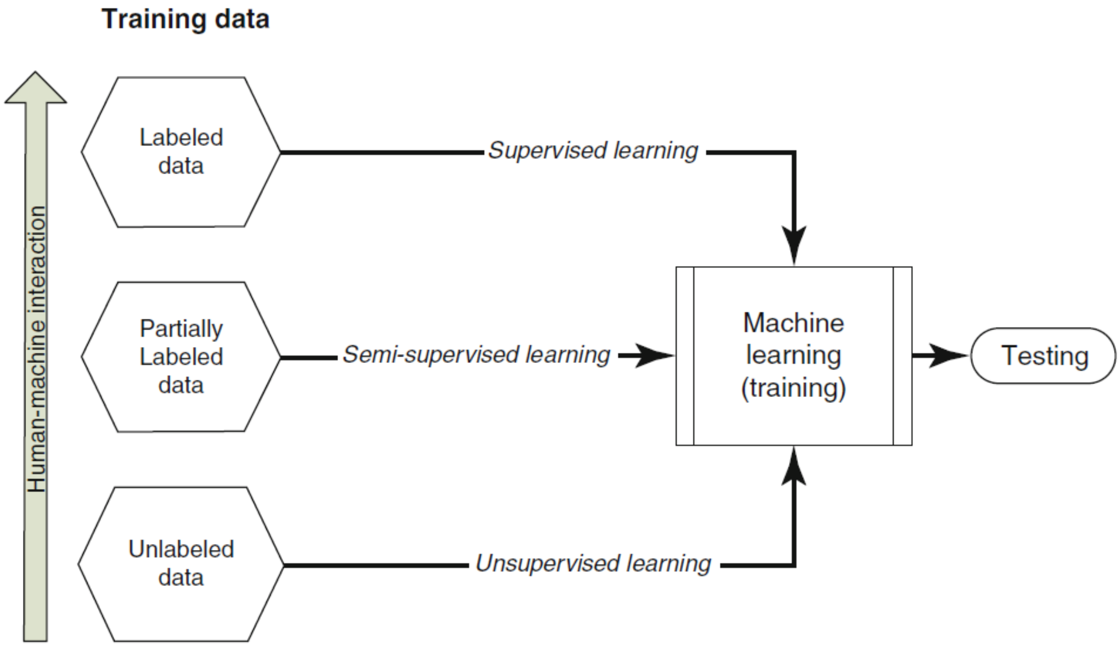

Figure 3.1: Machine Learning categories according to the input data [EM15]

3.1.1 Machine Learning Metrics

There are a few metrics with which we can evaluate Machine Learning algorithms. First, there is accuracy which can also be called classification accuracy. Accuracy represents the ratio of the sample data's number as well as the number of correct predicted outputs. However, it is not the best way to calculate the accuracy because e.g. if we have 98% samples of class A and 2% samples of class B in our sample date, the accuracy could be 98%. But if the sample data contains 60% samples of class A and 40% samples of class B the accuracy would decrease to 60% although the prediction was "accurate".

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions made}}$$

To define a "better" accuracy for Machine Learning we have to introduce some new terms: True Positive, True Negative, False Positive and False Negative. To simply explain it I will use the outcomes "YES" and "NO".

- If the algorithm predicts "YES" and the actual outcome is "YES" then this is called True Positive.
- If the algorithm predicts "NO" and the actual outcome is "NO" then this is called True Negative.
- If the algorithm predicts "YES" and the actual outcome is "NO" then this is called False Positive.

- If the algorithm predicts "NO" and the actual outcome is "YES" then this is called False Negative.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total number of samples}}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{F1 Score} = 2 * \frac{1}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}$$

Combining these new metrics results in the new formula of accuracy. Other metrics are precision and recall. There is also the F1 Score, which is the harmonic mean of precision and recall. The F1 Score tells you how precise the Machine Learning classifier functions [MLMETRICS18].

3.2 Technological Background

3.2.1 Philips HUE

The light bulbs of the Philips Hue are wireless controllable color-changing LED lamps first released in October 2012. During that time it was exclusive to Apple's ecosystem. The Philips Hue light bulbs were also the first smart lamps on the market. The communication with the Philips Hue was fulfilled by an internet connection with the Hue Bridge which was connected to the network via Ethernet connection. The Hue Bridge is also the central controller for the communication and is connected with every lamp in the same network. Since June 2019, it has been possible to use products of the Philips Hue without using the Hue Bridge because of the lately added Bluetooth connection into the lamps. The user has to install the Philips Hue Bluetooth app to connect with his or her lamps. However, the user has to enable location services to use the Bluetooth service. The addition of the connectivity via Bluetooth does not change the fact that every light bulb can be connected to the Hue Bridge. The first lamps, which were introduced, produced 600 lumens while the newer ones produce 800 lumens. Furthermore, the lamps are all dimmable and there are 3 different types of lamps: White, White Ambiance, and Color & White Ambiance [HUE21].

Type	Color	Temperature
White	white light	2700K
White Ambiance	warm soft white to bright day light	2700K - 6500K
Color & White Ambiance	White Ambiance + 16 million colors	2000K - 6500K

Table 3.1: Different types of Hue Lamps and their color and color temperature [HUE21]

Figure 3.2: The Philips Hue Bridge from June 2018 [HUEFIG21]

API

The Hue Bridge has a RESTful interface with which developers can communicate via HTTPS calls in the same network. The base URL for this is https://<Bridge IP Address>/api. There are several ways to find out the Bridge IP Address. With an mDNS discovery app it is possible to find Philips Hue in the network, to look in the routers DHCP table, to visit https://discovery.meethue.com, or to connect the Philips Hue app with the Hue Bridge and look up the Bridge IP Address in the app. The next step is to create a new user and with the newly created username, the developer can send HTTPS requests getting information about all the lights connected to the Hue Bridge, and even to control them by turning them on or off or to change the color, the intensity or the brightness. There is also an API Debug tool from Hue in the webbrowser when visiting https://<bridge ip address>/debug/clip.html [API21].

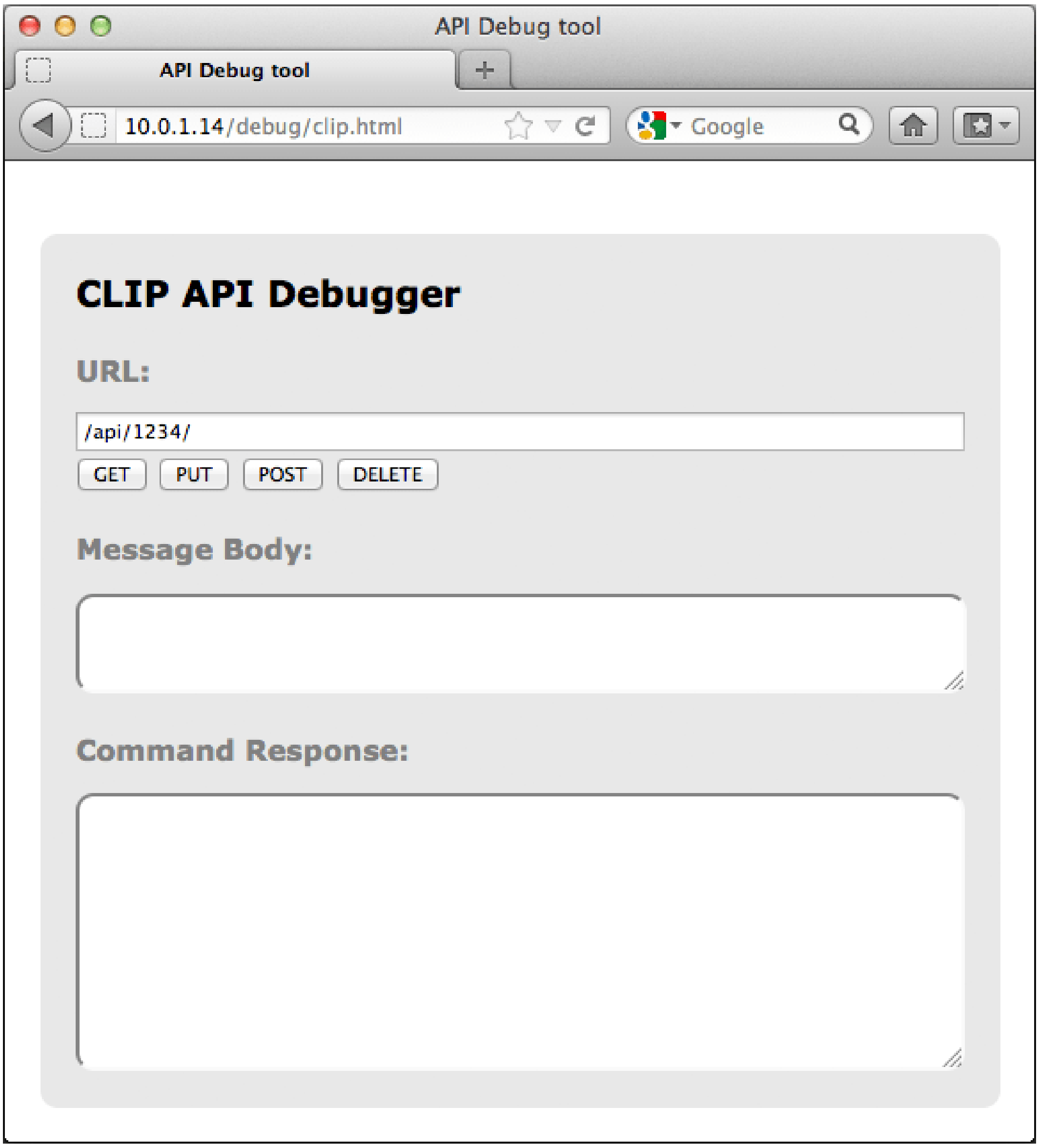

Figure 3.3: API Debug tool in the browser [APITOOL21]

1. Raspberry Pi with sensors as subchapters

3.2.2 Raspberry Pi

The Raspberry Pi is a series of single-board computers. The Raspberry Pi computers are used in many areas. That's because of the low cost, modularity, and open design of the Raspberry Pi [PI22]. Some areas where the Raspberry Pi is used are to learn programming skills, build hardware projects or do home automation. Another aspect of the Raspberry Pi is the general purpose input/output (GPIO) pins which allow the user to control components for physical computing (e.g. temperature sensor). The Raspberry Pi project is part of the open source ecosystem. The main operating system is Pi OS, which is also a Linux distribution [OPENPI22].

Figure 3.4: Raspberry Pi 4 [PIMODEL22]

4 Design

The following experimental design is to create a predictive control system for indoor lighting. There are some variables in this experimental design. First, there is the experiment's independent variable, which can be manipulated by the user. The independent variable will be the state of the light because this is the only value that can be changed and has no dependencies on other variables. The variables depend on the ones that we control. When the user turns the light on or off depends on physical measurements, e.g., the room's brightness. The control variable of the experiment is the place of the experiment, including its environment. This variable must be constant because it can negatively affect the experiment's credibility when it changes. You can rerun the experiment multiple times to control and compare the results to the control variable. Last but not least, there are also confounding variables, which we cannot control. In our case, confounding variables can be the weather or the duration of daylight [EXDESIGN22]. The room, where the experiment takes place, has only one window where the sunlight can enter. The experiment will be divided into two steps with different duration. The first part is about collecting usage data of the light with additional parameters (e.g., date and time). This part of the experiment should at least last four weeks. The second part of the experiment uses the collected data to create a predictive control system for indoor lighting. This system should be able to predict the light state and to accordingly change it. The final goal of the experiment is to see whether it is possible to create a predictive automation system for controlling home lighting. I will test the resulting system for one week with a time metric to evaluate how correctly it will predict the desired state of light. Finally, I will note all results in a log.

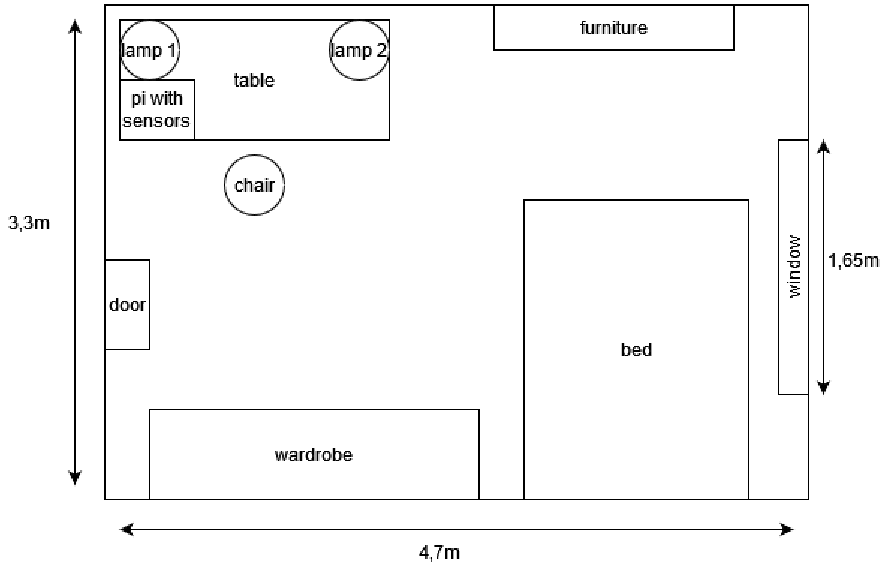

Figure 4.1: Map of the room

5 Implementation

The experiment will be split into two parts to create a predictive control system for indoor lighting. The first part is collecting user data on the light related to the sensors. The second part of the experiment is to train a machine learning algorithm with the collected samples of user data and to use the same setting which was used for collecting the data in a real-life environment to predict light usage. Figure 2.3 was a template for the general setup. I used a Raspberry Pi as the main computer with all the sensors connected to the Raspberry Pi. The PIR motion sensor mentioned in [FGG21], [JSE+18], and [Bha16] was also part of the sensors I have used for my experimental setup. The authors of [DH14] tried to predict Smart Home Lighting Behavior. The essential pieces of information in their database were time, temperature, light sensor, and light switch value. Therefore, I also used a temperature sensor and a light sensor to measure the lux value of the room. In addition, I used two Philips Hue White and Color Ambiance Play Lightbars for the lighting. The Hue Bridge, which was connected to my home network, was also connected with the Philips Hue lamps. The Raspberry Pi collected the data for the first part of the experiment with a python script that logged the date, time, lux value, temperature value, and the light state in a CSV file. The Raspberry Pi retrieved this information with REST calls (GET requests) to the Hue Bridge to get the light state. When the PIR sensor detected any motion then the Raspberry Pi logged every sensor's current state, including time and day. When there was not any motion detection from the PIR sensor, the python script triggered the data collection after a time limit of one minute. The light state changed via the Philips Hue App for smartphones. The experiment setup had the Raspberry Pi and the sensors on the left corner of my table. There, the PIR sensor pointed to the room's door. The light sensor lay flat that the sensor area could point to the top. The temperature sensor itself did not need to touch any surface because otherwise, the surface's temperature could negatively influence the data and the experiment. The Philips Hue Lamps were positioned in the table's left and right back corners. For the second part of the experiment, the collected data was implemented to train a machine learning algorithm. In paper [DH14], they used the Very Fast Decision Tree (VFDT) algorithm to correctly predict the information, but they calculated its accuracy. That means that they did not test the algorithm in a real-life setting. They compared multiple machine learning algorithms to each other besides VFDT, like Naive Bayes and Artificial Neural Networks. Their experiment tested these algorithms in a non-real-time and real-time switch activity learning. The results were that the VFDT algorithm performed well in both settings. Therefore, they recommended the usage of the VFDT algorithm in the smart home lighting context. Another similar system design was proposed where they used a Raspberry Pi to track the user's behavior in Smart Home. They also used a Decision Tree algorithm to predict the user's behavior. Their use case of the system was the prediction of the TV device, but you could replace this with a music device or lights. They used a Python script to collect the data and logged the device's status for every minute. The system was used in a real-life setting [AA21]. For the second part of the experiment, I used the VFDT algorithm to predict lighting behavior. I converted some of the data sets (date to weekday, time to seconds passed) and then I have trained and calculated the algorithm's accuracy. I wanted to eliminate one case for the light prediction. That is the prediction when there is no presence in the room. I only let the algorithm collect the data from sensors and predict whenever the PIR sensor detects any motion. After every prediction, the Raspberry Pi checks the current light status, and only if the predicted state differs from the actual state, the Raspberry Pi will send the predicted state to the Hue Bridge, to turn the lights on or off. The script logs the change of the light state in a CSV file with the date and time as well as the current sensor values. Additionally, the user will control manual changes outside of the system with the help of the Philips Hue smartphone app. These manual changes will be noted in a log journal.

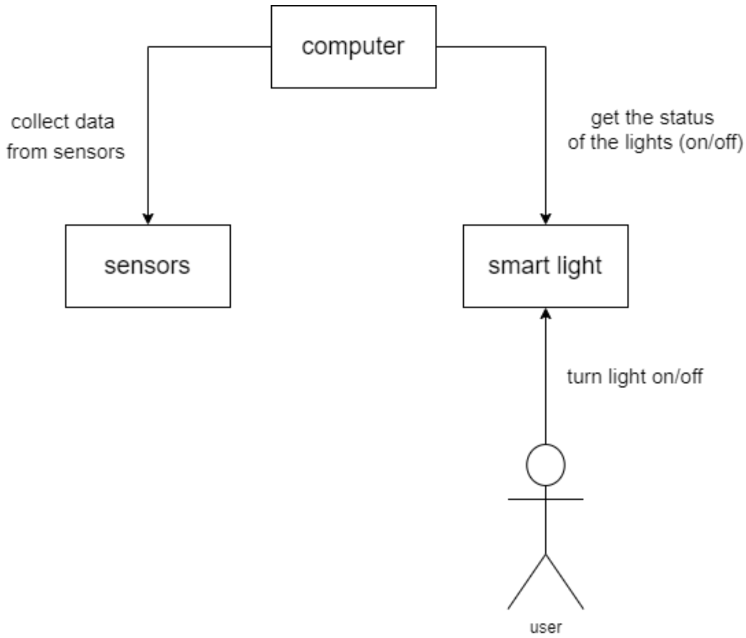

Figure 5.1: Setup for data collection

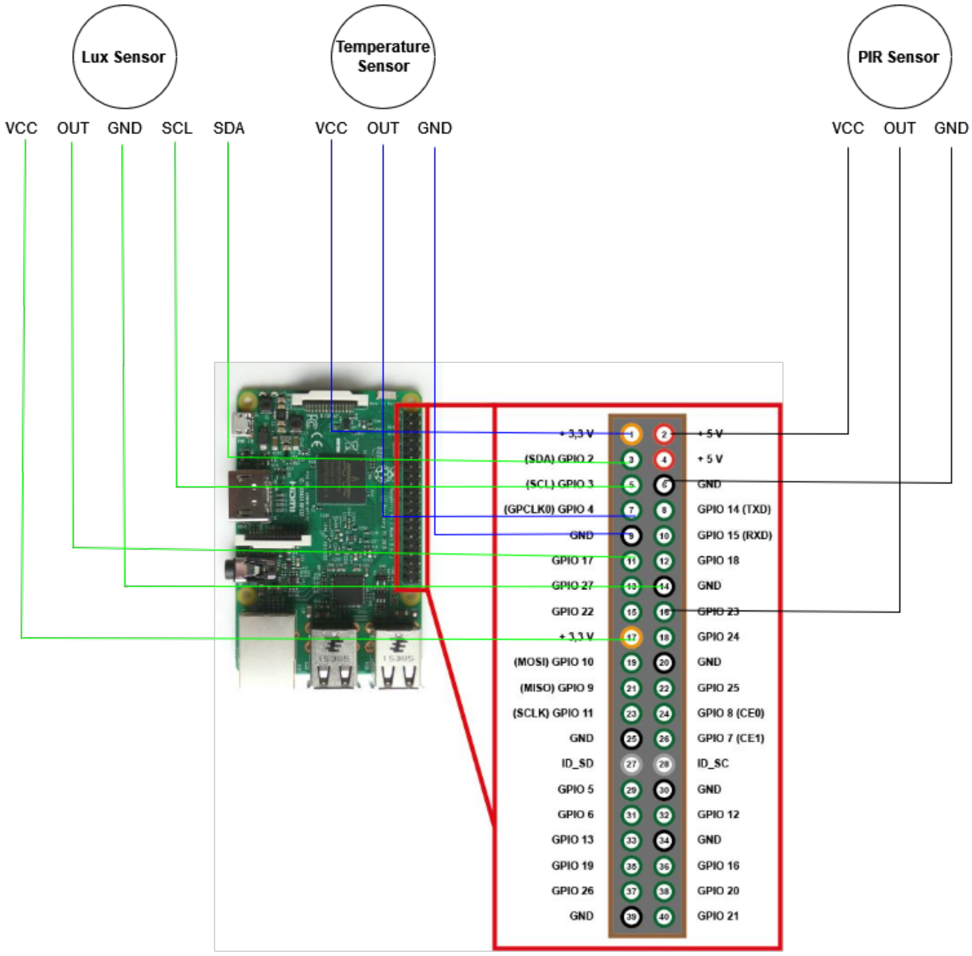

Figure 5.2: Raspberry Pi GPIO Setup with sensors [PIGPIO22]

```
import csv
import json
import os
import sys
from collections import namedtuple
from datetime import datetime

import requests
import RPi.GPIO as GPIO

libdir = os.path.join(os.path.dirname(os.path.dirname(os.path.realpath(__file__))), 'lib')
if os.path.exists(libdir):
    sys.path.append(libdir)
from waveshare_TSL2591 import TSL2591

# PIR motion sensor config
SENSOR_PIN = 23
GPIO.setmode(GPIO.BCM)
GPIO.setup(SENSOR_PIN, GPIO.IN)
```

```
#light sensor config
sensor = TSL2591.TSL2591()

#temp sensor config
def getTempSensor():
    for i in os.listdir('/sys/bus/w1/devices'):
        if i != 'w1_bus_master1':
            ds18b20 = i
    return ds18b20

def read(ds18b20):
    location = '/sys/bus/w1/devices/' + ds18b20 + '/w1_slave'
    tfile = open(location)
    text = tfile.read()
    tfile.close()
    secondline = text.split("\n")[1]
    temperaturedata = secondline.split(" ")[9]
    temperature = float(temperaturedata[2:])
    celsius = temperature / 1000
    return celsius

def my_callback(channel):
    today = datetime.today()
    time = today.strftime("%H:%M:%S")
    today = today.strftime("%d-%m-%Y")
    light1 = requests.get('http://192.168.178.44/api/HEoGGZfp-zwpm26aFEIq49-TvaHGpJdPPSTnnxpF/
lights/1')
    light2 = requests.get('http://192.168.178.44/api/HEoGGZfp-zwpm26aFEIq49-TvaHGpJdPPSTnnxpF/
lights/2')
    lux = sensor.Lux
    temperature = read(tempSensor)

    newRow = [today, time, lux, temperature, str(json.loads(light1.text)['state']['on']), str(
json.loads(light2.text)['state']['on'])]
    with open('user-data.csv', 'a') as f:
        writer = csv.writer(f)
        writer.writerow(newRow)

try:
    print("Start collecting data...")
    tempSensor = getTempSensor()
    GPIO.add_event_detect(SENSOR_PIN , GPIO.RISING, callback=my_callback)
    while True:
        time.sleep(100)
except KeyboardInterrupt:
    print("End collecting data...")
    sensor.Disable()
GPIO.cleanup()
```

Listing 5.1: Data Collection [TEMPSENS17] [PIRSENS17] [LUXSENS21]

```
def my_callback(channel):
    pi_today = datetime.datetime.today()
```

```
    pi_time = (pi_today.hour * 3600) + (pi_today.minute * 60) + pi_today.second
    lux = sensor.Lux
    temperature = read(tempSensor)

    new_light_state = vfdt_tree.predict([[pi_today.weekday(), pi_time, lux, temperature]])
    print(new_light_state[0])
    bool_value = False
    if(new_light_state[0]):
        bool_value = True
    else:
        bool_value = False

    requestobj = {'on': bool_value}

    light_state = requests.get('http://192.168.178.44/api/HEoGGZfp-zwpm26aFEIq49-
TvaHGpJdPPSTnnxpF/lights/1')
    if(json.loads(light_state.text)['state']['on'] != bool_value):
        light1 = requests.put('http://192.168.178.44/api/HEoGGZfp-zwpm26aFEIq49-
TvaHGpJdPPSTnnxpF/lights/1/state', json = requestobj)
        light2 = requests.put('http://192.168.178.44/api/HEoGGZfp-zwpm26aFEIq49-
TvaHGpJdPPSTnnxpF/lights/2/state', json = requestobj)

        newRow = [pi_today, lux, temperature, bool_value]
        with open('prediction-log.csv', 'a') as f:
            writer = csv.writer(f)
            writer.writerow(newRow)

# convert date and time values
df = pd.read_csv('user-data.csv')
print(datetime.datetime.strptime(df.date[0], '%d-%m-%Y').weekday())
for index, row in df.iterrows():
    df.at[index, 'date'] = datetime.datetime.strptime(row['date'], '%d-%m-%Y').weekday()
    df_time = datetime.datetime.strptime(row['time'], '%H:%M:%S')
    df.at[index, 'time'] = (df_time.hour * 3600) + (df_time.minute * 60) + df_time.second

X = df.drop(columns=['light1', 'light2'])
print(X)
y = df['light1']

# Configure VFDT algorithm
title = list(df.columns.values)
features = title[:-2]
vfdt_tree = vfdt.Vfdt(features, delta=0.01, nmin=100, tau=0.5)

#Train VFDT
vfdt_tree.update(X, y)

try:
    print("Start prediction system...")
    tempSensor = getTempSensor()
    while True:
        channel = GPIO.wait_for_edge(SENSOR_PIN, GPIO.RISING, timeout=5000)
        if channel is None:
```

```
            print('No motion detected')
        else:
            print('Edge detected on channel', channel)
            my_callback(channel)
except KeyboardInterrupt:
    print("End prediction system...")
    sensor.Disable()
GPIO.cleanup()
```

Listing 5.2: Predict indoor lighting

6 Experiment Design and Evaluation

The first part of the experiment lasted from January 15 to February 14, 2022, to collect the user data. I used the Philips Hue Lightbars as an additional light source in contrast to the existing room lighting. I used the same settings on both lamps (same color, dimming, etc.). For the first part, I turned the lights on when there was not enough sunlight in the room. In addition, I turned on the room's lights when there was no sunlight. At the end of this period, the CSV file had 69.448 entries. Before starting the second part of the experiment, I tested the accuracy and F1 Score of the VFDT algorithm. I used 80% of the data for training and 20% for testing. The results were an accuracy of 0.991 and an F1 Score of 0.988, being pretty similar to the results in [DH14]. Here, the accuracy is the ratio between the sum of correct predictions for turning the light on and off and the total number of the data set. The F1 Score consists of precision (the ratio between correctly turning on the light and the sum of predictions turning the light on and turning the light off where it should have been on) and recall (the ratio between correctly turning on the light and the total number of predictions turning the light on). The second part of the experiment lasted from March 17 to March 24, 2022. The metric which I have used for the light prediction was to wait for a time period of 5 minutes whenever the desire of turning the lighting on or off appeared or to check whether the prediction fits to the current environment. After 5 minutes, when the system did not match with the desired outcome, I manually switched the lights and noted the switch in a log. I regularly checked the log file of the system to control whether it turned the lighting on or off while there was no presence in the room. For this part of the experiment, the number of manual switches was nine. On March 17 and 18, the system turned the lights on at 15:44 and 15:11. I was not expecting to turn on the lights, but I accepted the decision because it was pleasant. On March 17, I had to turn the light off at two separate times (18:05 and 18:24) because I left the room, and the system did not turn the lights off. From 19:00 to 8:00, the system did not have any problems predicting the correct decision. Moreover, the system has never decided when there was no presence in the room because the PIR sensor did not measure any motion. On March 18 at 8:07, the system turned on the lights, as I expected, and I only had to wait for a few seconds for this decision. However, at 8:50, I had to turn off the lighting because I left the room. It was interesting to see that the system worked correctly on March 19 at 3:44 and 3:45, where the lights were on and off correctly, since I have not used the lighting very often from 12:00 - 6:00. On the same day, at 14:25, the system falsely turned the lights on while I was in the room. After checking the logs, I manually turned the lighting off at 14:27. After that, the system worked correctly for the rest of the day. On March 20, the system worked correctly for the whole day. On March 21, 22, and 23, I manually turned the lighting off at 14:22, 15:03, and 15:48 because I left the room and expected the lighting to turn off. However, on these days, the system did turn on the lights for the first time between 14:10 and 14:37. On March 24 at 18:13, the lighting correctly turned on at when I had entered the room, but I had to manually switch off the lighting because I left the room at 18:20.

date	**time**	**switch**	**by who?**
March 17	18:05	on ->off	user
March 17	18:24	on ->off	user
March 18	8:50	on ->off	user
March 19	14:25	off ->on	system
March 19	14:27	on ->off	user
March 21	14:22	on ->off	user
March 22	15:03	on ->off	user
March 23	15:48	on ->off	user
March 24	18:20	on ->off	user

Table 6.1: Log data of user switches and wrong switches from the system

The system worked most of the time correctly, but the most issues appeared between 14:00 and 16:00 (see Table 6.1). The explanation for this could be that the machine-learning algorithm's data was collected between January and February. Therefore, the days have been shorter than they have been in March. However, when we compare the lux values to the systems log and the collected user data, we can see that the values in March without lighting are almost the same as the collected data with lighting between 14:00 and 16:00. Another reason for the difference in lux value could also be the weather. During the second part of the experiment, the weather was not as cloudy as collecting the data in the first part. The difference in the lux value could also explain the wrong switch of the system on March 19 at 14:25. Another drawback of the system is that it can not detect when the user leaves the room to turn off the lights. Instead, the system waits for the PIR sensor to send an event so the VFDT algorithm can predict with the data from the sensors. Therefore, the system does not predict anything when there is no presence in the room. But that is also why I turned the light off manually on March 18 at 8:50. The system had no issues when there was no sunlight because the room's lighting was on. The room's lighting supported the system to understand when you need to turn the lights on/off. The predictions without sunlight are of a different kind compared to the predictions during daylight because these are more of a pattern. The reason for the pattern is that the room's lighting adds a constant value of lux.

At the end of the experiment, there are a few things to improve the system. First, it would have helped to measure the natural lighting by either positioning a light sensor near the window or simulating the natural lighting by mapping the weather status to specific values like the authors of [GNN+17] did. The current light sensor measured the lighting value in the room, which got influenced when the user/system turned on the room's lighting. Therefore, the system would be more accurate with an additional light sensor during daytime. Generally, using multiple sensors could also improve the detection of human presence, placing multiple PIR sensors to locate the user‘s presence in a certain area of the room. Additionally, letting the system learn while using it can improve its prediction because it adapts to environmental changes, e.g. longer/shorter daytime, or when the user changes his or her behavioral patterns.

7 Conclusion and Outlook

According to the Bachelor Thesis, there have already been enough methodical and technological backgrounds to create a predictive control system for smart lighting and smart homes. The system which was shown in this Bachelor Thesis is one example of creating a smart lighting system in human behavior prediction and smart home. The system's accuracy was pretty high. The user had to manually overrule when the lights should not be used because there was no presence in the room. The system could easily predict behavior patterns, like light usage during nighttime. This accuracy was just achieved with motion, temperature, and light sensors. The vital part of these systems is to decide which values of the environment we want to measure and how to measure these values. An excellent example of this would be the light sensor's presentation in the presented system of the Bachelor Thesis. When we made a sensitivity analysis, this sensor got falsely influenced by longer daytime and less cloudy weather, which led to a false prediction of the system. Therefore, we can let the system constantly learn or use more sensors to measure values in order to get more insight into the environment's context. If we get more insight into the environment's context the system will be less sensitive against unexpected values. Nevertheless, the system should not be fully automatic because of the constant changes in the environment. Last but not least, the user should be given a choice of overruling decisions.

Outlook

In my Bachelor Thesis, we have learned that lighting goes beyond illumination. More applications with new functionalities and use-cases arise. However, at the moment, most smart lighting systems mainly focus on illumination and dimming the lights to an appropriate level, and reducing energy consumption. But we have learned how light can also affect productivity, well-being, and emotions. Therefore, I am convinced that this is just the start of the smart lighting systems. Moreover, I believe with more and different sensors, it will be possible to automate more functionalities and to recognize the emotions and actions of the user. One important criterion is that the system should constantly learn. Unfortunately, it is still inconvenient for the user to use smart lighting systems with sensors. The developers should consider this inconvenience when they develop systems like these. Their installation and use should come after the plug-and-play approach.

A Zusammenfassung

Neue Lösungen von Licht wurden schlauer, indem sie mehr Energie gespart haben und wegen der Einführung von Technologien wie der Leuchtdiode (LED). LEDs bieten viele neue Möglichkeiten an wie man mit dem Licht interagieren kann. Dies öffnet die Tür zu Smart Lighting Systemen. Diese Systeme können mit Sensoren kombiniert werden und automatisieren den Beleuchtungsprozess. Dabei geht es nicht nur um die Beleuchtung, sondern das Licht kann auch eine Wirkung auf die Produktivität, das Wohlbefinden und auf Gefühle haben. Es existieren bereits mehrere Anwendungsfälle wie in Smart Cities, in Lagerhallen und sogar beim Wohnen. Das Ziel dieser Bachelorarbeit ist, die aktuellen Methoden zu analysieren und ein prädiktives Kontrollsystem für Innenbeleuchtung zu erstellen. Dabei werden Smart Lights von Philips Hue in Kombination mit einem Raspberry Pu und mehreren Sensoren (Bewegungssensor. Temperatursensor und Lichtsensor) verwendet. Wir werden auch auf den technischen und methodischen Hintergrund dieser Thesis schauen. Für das Experiment verwenden wir Benutzerdaten die wir selber sammeln. Anschließend verwenden wir diese für einen Machine Learning Algorithmus (Very Fast Decision Tree) um die Beleuchtung in einem realen Umfeld vorherzusagen. Anschließend werden wir das entstehende System mit Zeitmetriken nach der Genauigkeit bewerten. Mit einer Sensitivitätsanalyse können wir sagen, dass das System falsche Vorhersagen gemacht hat, aufgrund von Änderungen des Umfelds wie längere Tageszeiten oder auch veränderte Wetterzustände (weniger bewölkt, mehr Sonnenlicht).

Bibliography

[AA21] A. F. Abbas, M. Z. Abdullah. "Design and Implementation of Tracking a user's Behavior in a Smart Home". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 1094. 1. IOP Publishing. 2021, p. 012008 (cit. on p. 31).

[AD08] M. Aiello, S. Dustdar. "Are our homes ready for services? a domotic infrastructure based on the web service stack". In: *Pervasive and Mobile Computing* 4.4 (2008), pp. 506–525 (cit. on p. 17).

[API21] Philips Hue Developer Program. *Get Started*. 2021. URL: https://developers.meethue.com/develop/get-started-2/ (cit. on p. 27).

[APITOOL21] Philips Hue Developer Program. *API Debug tool*. 2021. URL: https://developers.meethue.com/wp-content/uploads/2018/02/clip.png (cit. on p. 27).

[Bha16] V. Bhangdiya. "Low power consumption of LED street light based on smart control system". In: *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*. IEEE. 2016, pp. 619–622 (cit. on pp. 19, 31).

[DH14] I. B. P. P. Dinata, B. Hardian. "Predicting smart home lighting behavior from sensors and user input using very fast decision tree with Kernel Density Estimation and improved Laplace correction". In: *2014 International Conference on Advanced Computer Science and Information System*. IEEE. 2014, pp. 171–175 (cit. on pp. 31, 37).

[EM15] I. El Naqa, M. J. Murphy. "What is machine learning?" In: *machine learning in radiation oncology*. Springer, 2015, pp. 3–11 (cit. on pp. 23, 24).

[EOE12] H. van Essen, S. Offermans, B. Eggen. "Exploring the role of autonomous system behavior in lighting control". In: *Proceedings of Designing Interactive Lighting workshop at DIS*. Vol. 12. Citeseer. 2012 (cit. on pp. 16, 17).

[EXDESIGN22] Statistics How To. *Experimental Design*. 2022. URL: https://www.statisticshowto.com/experimental-design/ (cit. on p. 29).

[FGG21] M. Füchtenhans, E. H. Grosse, C. H. Glock. "Smart lighting systems: state-of-the-art and potential applications in warehouse order picking". In: *International Journal of Production Research* 59.12 (2021), pp. 3817–3839 (cit. on pp. 13, 15, 19, 20, 31).

[GNN+17] I. Georgievski, T. A. Nguyen, F. Nizamic, B. Setz, A. Lazovik, M. Aiello. "Planning meets activity recognition: Service coordination for intelligent buildings". In: *Pervasive and Mobile Computing* 38 (2017), pp. 110–139 (cit. on pp. 20, 21, 38).

[HUE21] Wikipedia The Free Encyclopedia. *Philips Hue*. 2021. URL: https://en.wikipedia.org/wiki/Philips_Hue (cit. on pp. 25, 26).

[HUEFIG21] Wikipedia The Free Encyclopedia. *Philips Hue Bridge - June 2018*. 2021. URL: https://upload.wikimedia.org/wikipedia/commons/thumb/e/e4/Philips_Hue_Bridge_-_June_2018_%5C%281986%5C%29.jpg/1024px-Philips_Hue_Bridge_-_June_2018_%5C%281986%5C%29.jpg (cit. on p. 26).

[JSE+18] E. Juntunen, E.-M. Sarjanoja, J. Eskeli, H. Pihlajaniemi, T. Österlund. "Smart and dynamic route lighting control based on movement tracking". In: *Building and Environment* 142 (2018), pp. 472–483 (cit. on pp. 18, 31).

[KWLA13] E. Kaldeli, E. U. Warriach, A. Lazovik, M. Aiello. "Coordinating the web of services for a smart home". In: *ACM Transactions on the Web (TWEB)* 7.2 (2013), pp. 1–40 (cit. on p. 22).

[LUXSENS21] CQRobot Wiki. *TSL2591X Light Sensor*. 2021. URL: http://www.cqrobot.wiki/index.php/TSL2591X_Light_Sensor (cit. on p. 34).

[MLMETRICS18] Aditya Mishra. *Metrics to Evaluate your Machine Learning Algorithm*. 2018. URL: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234 (cit. on p. 25).

[OPENPI22] opensource.com. *What is a Raspberry Pi?* 2022. URL: https://opensource.com/resources/raspberry-pi (cit. on p. 28).

[PI22] Wikipedia The Free Encyclopedia. *Raspberry Pi*. 2022. URL: https://en.wikipedia.org/wiki/Raspberry_Pi (cit. on p. 27).

[PIGPIO22] Elektronik Kompendium. *Raspberry Pi: Belegung GPIO*. 2022. URL: https://www.elektronik-kompendium.de/sites/raspberry-pi/fotos/raspberry-pi-15b.jpg (cit. on p. 33).

[PIMODEL22] Raspberry Pi. *Raspberry Pi 4*. 2022. URL: https://images.prismic.io/rpf-products/3a15d4da-46e3-4940-8be6-9fc7d201affe_RPi_4B_FEATURED.jpg?ixlib=gatsbyFP&auto=compress%5C%2Cformat&fit=max&q=50&w=799&h=533 (cit. on p. 28).

[PIRSENS17] PIDDLERINTHEROOT. *PIR Sensor*. 2017. URL: https://www.piddlerintheroot.com/pir-sensor/ (cit. on p. 34).

[TEMPSENS17] PIDDLERINTHEROOT. *Temp Sensor (ds18b20)*. 2017. URL: https://www.piddlerintheroot.com/temp-sensor-ds18b20/ (cit. on p. 34).

[VMP13] D. Vavilov, A. Melezhik, I. Platonov. "Smart home user's behavior prediction". In: *2013 IEEE Third International Conference on Consumer Electronics ¿ Berlin (ICCE-Berlin)*. IEEE. 2013, pp. 1–4 (cit. on p. 13).

All links were last followed on May 6, 2022.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature