Final Report

Optimizing office space utilization using an Indoor Air Quality sensor network

Jan-Willem Gmelig Meyling

Leon Hoek

Sayra Ranjha

Technische Universiteit Delft

Final Report

Optimizing office space utilization using an Indoor Air Quality sensor network

by

Jan-Willem Gmelig Meyling
Leon Hoek
Sayra Ranjha

in partial fulfillment of the requirements for the degree of Bachelor of Science
at the Delft University of Technology,
to be defended publicly on Tuesday July 2nd, 2019 at 15:00.

Project duration:	April 21st, 2019 – June 25th, 2019	
Supervisors:	Dr. Hayley Hung,	TU Delft
	Dr. Ekin Gedik,	TU Delft
	Ir. Margit Blok,	VTTI
	Margreeth Doornbosch,	VTTI
	Ir. Ruud Timmermans,	VTTI
Bahelor Project Coordinators:	Dr. Huijuan Wang,	TU Delft
	Ir. Otto Visser,	TU Delft

This thesis is confidential and cannot be made public until July 2nd, 2019.

Preface

In front of you lies the bachelor's thesis created by Jan-Willem Gmelig Meyling, Leon Hoek and Sayra Ranjha as part of the Bachelor Computer Science and Engineering at the Delft University of Technology. Over the course of ten weeks, we have explored the possibilities of optimizing office space utilization and employee well-being that emerge with an indoor air quality sensor network and dashboarding solution. This report documents the progression, system architecture, and key findings of the project. The VTTI group commissioned this project.

We want to thank everyone at the VTTI headquarters for their enthusiasm and involvement with our air quality measurements and software. We want to thank Margit Blok (Global Director HSE at VTTI), Margreeth Doornbosch (Management Assistant at VTTI) and Ruud Timmermans (Global Automation Lead at VTTI) in particular for enabling this project and for their coordination of this project. Finally, we would also like to thank Tibor Casteleijn and Felix van den Horst from Clairify for their continuous support and encouragement, and our TU coaches Hayley Hung, Associate Professor of the Intelligent Systems Department at the Delft University of Technology, and Ekin Gedik, Postdoctoral Researcher of the Socially Perceptive Computing Group at the Delft University of Technology, for their valuable input with regard to the data analysis possibilities and feedback on our progress.

Jan-Willem Gmelig Meyling, Leon Hoek and Sayra Ranjha
Amsterdam, July 12, 2019

Summary

Sick Building Syndrome is present in 30% of all office buildings and can cause serious health damage over time. This is an era where sustainability and well-being are becoming dominant aspects of life. As a result, it is becoming increasingly important to businesses to invest in their employees' well-being and health. The VTTI group cares for the well-being of their employees, and is looking for a tool to optimize the utilization of their building for perceived thermal comfort and indoor air quality.

This report documents the development of *Claire*, an indoor air quality dashboard that helps to identify local air quality problems. Using *Claire*, employees can be rearranged throughout the space, learn about the characteristics of their office, and for example switch to another meeting room. *Claire* translates measurements into insights. *Claire* learns about the behavior of the office, and gives recommendations once she notices that the indoor air quality can be improved.

Claire is backed by an indoor air quality sensor mesh network, which has been developed as part of this project. The sensors continuously measure temperature, humidity and carbon dioxide concentrations. The sensors connect to a cloud infrastructure through a local internet gateway. In the cloud the data gets processed. All measurements are displayed real-time in the dashboard.

Claire is different from existing products in several ways. First, the sensors developed measure both dry-bulb and black globe temperature, which gives it a temperature reading that describes human thermal comfort more accurately. This is not done in competing products. Furthermore, the sensors fill the gap for small and medium-sized enterprises (SMEs): the sensor network is able to get fine-grained results due to its high sensor density, whilst still being very easy to setup with no adjustments to the building being required. Finally, the developed data analysis methods translate the measurements from the sensor network to concrete suggestions, sent through a push notification, which enables workers to get involved with improving the indoor air quality in their office space.

Contents

1 Introduction

People spend 87% of their time indoors [1]. The buildings they spend their time in change continuously, which unconsciously affects their health. Sick Building Syndrome (SBS), a medical condition identified by a set of symptoms generally observed among office workers, is estimated to be present in 30% of all office buildings and can cause serious health damage over time [2]. This is an era where sustainability and well-being are becoming dominant aspects of life. Currently, 75% of job seekers mind that their potential employer is engaged in their well-being, and as a result, 57% continues to stay longer with a company if they do [3]. In response, 73% of companies believe it is their responsibility to guarantee that employee health and wellness will grow within the next 3 to 5 years [4]. Good indoor air quality also increases sustainability and corporate social responsibility (CSR) scores for, for example, the Global Reporting Initiative index.

1.1. VTTI

The VTTI group specializes in tank storage, and currently has over 9.2M cubic meters of storage spread over facilities in 14 countries. VTTI cares for the well-being of its employees and is aware of the impact of air quality. VTTI has over 5000 employees, 100 of which work from their headquarters in Rotterdam.

1.2. Problem Definition

VTTI is situated in a flexible office space. This means that the building is delivered as a flat floor, and that office spaces and meeting rooms can be set up flexibly. However, the ventilation systems are usually not adjusted when such changes are made to the floor plan. In Appendix A.2 it is shown that this can be particularly problematic in modern multi-zone HVAC systems. In particular employees that work in the south-west wing of the building report heat complaints, believed to be caused by solar radiation throughout the day. Conversely, employees that work on other sides of the building report cold issues, presumably caused by the ventilation trying to compensate for the heat in an undirected fashion.

Even though air quality complaints are common in offices, they are usually hard to quantify, and when indoor air quality is measured and quantified, the direct impact on employees remains vague. Considering the possibility that there may very well be a structural indoor air quality problem, VTTI would like to gain insight in the distribution and flow of heat, humidity and CO_2 throughout the building, so that they can optimize utilization of their building.

To obtain this insight and provide an opportunity to formalize new experiments on optimization strategies, a continuous real-time measuring system will be proposed, using a network of indoor air quality sensors that report to an online available dashboard. This dashboard will display indoor air quality throughout the building in an intuitive manner (see Section 2.2), and is also intended to give suggestions concerning proper ventilation design of the building.

1.3. Starting point: preliminary tests

In December 2018 initial air measurements were conducted at VTTI using Netatmo Healthy Home Coach CO_2 sensors. During this period surveys were conducted as well. Additionally, the Building Symptom Index was computed, a pseudo standardized scoring based on reported Sick Building symptoms among employees [5]. While the Building Symptom Index showed no worrisome results, there seemed to be room for improvement regarding improving employee personal air quality and comfort. Although a rapid increase to levels above common CO_2 guidelines was found in occupied meeting rooms, the levels found throughout the rest of the building did not explain the symptoms reported in the surveys. Therefore, Particulate Matter (PM) concentrations were measured between January and February 2019, using Alphasense OPC-R1 sensors. These measurements suggested that PM levels in the VTTI office were floating around the guidelines set by the World Health Organization (WHO). The PM measurements, however, had a great variance, and within some intervals the sensors produced completely unreliable data altogether. The accuracy of medium to low budget sensors seems to be insufficient for classifying PM concentrations or detecting PM incidents in the office space. Key learning from this test is that PM sensors could potentially be used for detecting anomalies in variance of indoor PM concentrations. For example, a major decrease in particulate matter concentrations was noticed after the filters of the ventilation system were replaced. This could mean that continuous measurements of PM could predict ventilation system maintenance needs.

This leads to the idea of collecting data at a larger scale, with greater sensor density and being able to read (and act on) these results in real time.

2
Project Plan

To address VTTI's Indoor Air Quality questions, properties of indoor air quality and existing sensor technologies to measure these properties were researched first. The properties and guidelines for indoor air quality are outlined in Appendix A. In order to measure these properties a network of sensors will be deployed at the VTTI office. These sensors will connect to a cloud infrastructure, where the data is collected and processed into periodic digests as well as real-time status reports available through an online dashboard.

The sensor devices will be equipped with temperature, humidity, and CO_2 sensors, because (a) complaints are mostly centered around thermal discomfort (Section 1.2), (b) high CO_2 levels are a good indicator of insufficient ventilation (Appendix A.2), (c) CO_2 levels, in particular in small rooms, were found to rise to levels that are known to affect cognitive performance (Section 1.3 and Appendix A.2), and (d) low-cost temperature, humidity and CO_2 sensors have been proven to be sufficiently accurate (Appendix C),

As (a) pollutants such as Volatile Organic Compounds (VOCs) and Particulate Matter (PM) have proven to be difficult to measure (Appendix C), (b) their contribution to perceived air quality is less known (Appendix A.3 and A.4), and (c) complaints from VTTI employees concentrated around thermal discomfort, the decision was made in consultation with VTTI to not take these pollutants into account for this project. Sensors for these properties will however be considered a nice-to-have inclusion.

The dashboard will display indoor air quality values reported from the sensor network on a map, possibly a heat map, although projecting simple values on the map will suffice. This map should show the distribution of air quality throughout the building and display indoor air quality hot-spots, which will allow VTTI to intervene, for example through the means of re-positioning desks or employees, turning on ventilation in certain rooms or opening a window. The dashboard should allow VTTI employees to obtain insight into historic trends in indoor air quality, and as such provide a means of indoor air quality experimentation. This historic trend will allow VTTI to find a systematic basis for incidental complaints, i.e. address whether or not there is a general indoor air quality problem or whether these problems are very specific.

VTTI has indicated that the tool must be intuitive and thoughtful. Rather than displaying the actual sensor values, these values should be compared to existing norms and guidelines or indoor air quality indexes. One possibility of a more interactive system is the ability to predict significant changes in indoor air quality and proactively sending employees actionable notifications to ventilate the room or suggest an alternative meeting room with fresher air.

2.1. Project requirements

At the start of the project, requirements were defined to establish clear goals for the software itself. These requirements describe the functionality the system should offer. The requirements also allow for the progress of the project to be verified. The requirements set for this project are as follows:

1. Develop a wireless sensor hardware/software combination that reports CO_2, temperature and relative air humidity (RH) levels through a local gateway interface to a central, cloud, infrastructure. A wireless setup eases the installation of the sensors.

2. Setup a back-end infrastructure that will collect and store measurements. This *must* comprise the following components:

- Setup a message broker to which the gateway can deliver its messages over internet;
- Setup a time series optimized database that will store and provide the measurements;
- Develop a time series data processing layer;
- Develop a data processing layer that will normalize sensor values, compute possible air quality indexes and classify values within general guidelines;
- Develop an eventing layer that will notify employees when values go beyond certain thresholds.

3. Develop the dashboard front-end application. This *must* comprise the following components:

   - Show real-time sensor values and derived attributes;
   - Show historic sensor value trends;
   - Ability to select a particular sensor and show its values real-time;
   - User authentication;
   - Ability to submit complaints about thermal comfort or perceived air quality / take part in the survey.

   Next, the dashboard *should* comprise:

   - Ability to select a certain wing of the building;
   - Basic alerts based on thresholds set by guidelines;
   - Better thermal comfort prediction by measuring the air velocity and the mean radiant temperature.

   Finally, the dashboard *could* comprise:

   - Ability to select a certain facility or floor of the building;
   - Advanced alerts and recommendation based on predictive analysis;
   - Ability to send notifications as a push notification to a smartphone;
   - Inclusion of actual Particulate Matter (PM) and total volatile organic compound (TVOC) concentrations.

2.2. Representative qualities

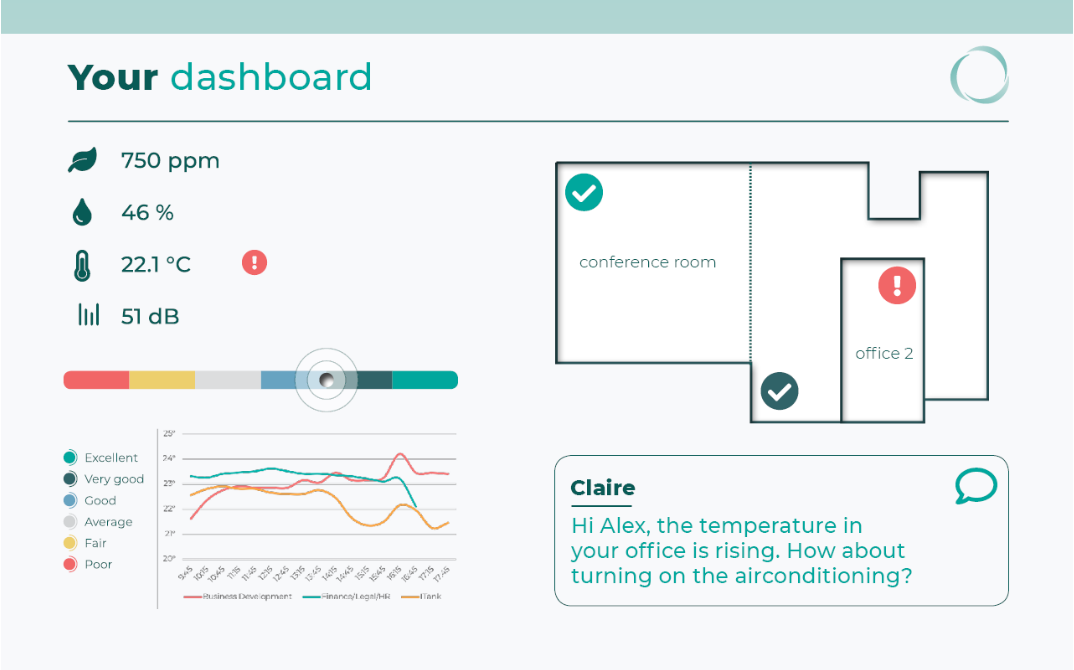
In this section the representative quality goals for this project will be introduced. These quantities will be referred to throughout the design section (Section 6).

2.2.1. User Friendliness

The system should be easy to use. This is twofold. First, the dashboard should be efficient, and any disruption should be minimized. A user of the system should be able to easily navigate through the pages and identify potential problems. Second, the system should be intuitive to use. No prior knowledge about indoor air quality or building ventilation is to be expected from the users. In order to accommodate for this, the system should translate the actual measurements to more indicative indicators (i.e. color scales, index numbers, or the - prediction of - percentage people dissatisfied). No training besides a brief introductory presentation should be required. Consecutively, the system should be performant, in order to minimize disruption caused by delays of the system.

2.2.2. Correctness

The proposed system should provide correct and reliable results. This means that the measurements should be accurate and timely, alerts should be legitimate, and recommendations to improve the indoor air quality should be justified. The accuracy of the system is important for the tool to be as little obtrusive as possible, and only disrupt the user when a significant improvement w.r.t. the indoor air quality can be achieved. The correctness is one of the primary qualities that will be taken into consideration for the selection of sensor hardware and statistical methods used for data analysis.



Hi Alex, the temperature in your office is rising. How about turning on the airconditioning?

Figure 2.1: Mock-up of envisioned dashboard, demonstrating the suggestions in order to improve local air quality

2.2.3. Evolvability, reusability and interoperability

With relative ease, it should be possible to deploy additional sensors at the current office or extend the system to other building facilities of VTTI. Furthermore, it should be possible to introduce sensors to the network with different hardware or firmware, and as a result return a different type of data.

2.2.4. Privacy and ethical implications

Because indoor air quality perception can be affected by health issues such as sleep deprivation, personal user data on air quality perception is considered to be highly sensitive. The questions posed to employees in surveys must be composed with privacy in mind. For example: no questions will be asked about specific symptoms experienced that are believed to be caused by poor indoor air quality, rather questions will be asked whether the indoor air quality is considered to be comfortable. Furthermore, all survey data will be fully anonymized, and only data derived from the surveys in an aggregated fashion will be displayed.

2.2.5. Security

Continuously measuring indoor air quality may expose some actual ventilation issues at VTTI. Obviously, it can be damaging if this information is leaked to the public, in particular without context. The system should therefore be developed with security in mind. Only authorized employees should have access to the dashboard. Defensive programming, a development methodology where security is considered in every step of the development process, will be used to enforce that also the back-end systems are built with security in mind. Furthermore, introducing embedded devices to a company network poses security challenges. Therefore the sensor network should not be reliant on existing company network infrastructure.

2.3. Development Methodology

The project comprises five individual subprojects: indoor air quality research, sensor comparison and development, embedded software development, back-end software development, data analysis and front-end development. Some of these projects are interdependent: without the knowledge about indoor air quality a deliberate decision on sensor hardware cannot be made, and without the sensors being assembled and deployed there will be no data to analyze or display in the dashboard.

The project was divided into sprints of two weeks. Roughly speaking, every sprint worked towards the completion of one of the five subprojects (see also the planning in Section 2.4). At the beginning of each sprint, the component would be broken into several tasks which would be distributed evenly over the group members during the sprint planning. The progress of each sprint was tracked in a KANBAN board.

During this project pair programming was used extensively. This helped familiarizing team members with the used technologies and developed components. For the components developed individually, a pull-based development methodology was used. Here the team member first makes a formal request to merge the code branch into the main code base, which is then reviewed by other team members. After several rounds of feedback and changes the pull request is eventually accepted and merged into the code base.

2.4. Planning

December 2018	Initial CO_2 measurements, research IAQ properties
Januari - Februari 2019	Initial PM measurements, continue research IAQ properties
April 2019	Research sensor hardware and communication protocols
March 7th	Project Plan meeting with VTTI

Week 1 (21 - 28 April)	Research IAQ properties, establish the project plan and draft literature study
April 26th	Project Progress meeting at VTTI
Week 2 (29 April - 5 May)	Approval of project plan, finalize decision on hardware selection based on project plan and literature study, finish research report, sensor delivery
Week 3 (6 - 12 May)	Sensor assembly and deployment at VTTI, development of basic data collection system in order to start data collection, draft design document
Week 4 (13 - 19 May)	Development of data analysis layer and dashboard frontend
May 16th	Sensor deployment and project update meeting at VTTI
Week 5 (20 - 26 May)	Development of data analysis layer and dashboard frontend
May 20th	Sensor network updates at VTTI
May 23th	Sensor network updates at VTTI
Week 6 (27 May - 2 June)	Development of data analysis layer, initial code release for VTTI, initial SIG code submission
May 29th	Project Progress meeting at VTTI
Week 7 (3 - 9 June)	First user feedback, development of data analysis layer, development of virtual assistant, labeling of measurement data at VTTI
June 5th	Project Progress meeting at VTTI
Week 8 (10 - 16 June)	More development, processing SIG feedback, data analysis, labeling of measurement data at VTTI
Week 9 (17 - 23 June)	More development, second SIG code submission, data analysis, final report writing
Week 10 (24 - 30 June)	Working on final report
June 25th	Submit final report
Week 11 (1 - 7 July)	Prepare final presentation at TU Delft, present findings at VTTI
2 - 5 July	Presentations

3 Research question

Ultimately, VTTI is looking for answers regarding their indoor air quality and thermal comfort questions. The research question is as follows: is there an indoor air quality problem at the VTTI headquarters, and if structural discomfort is found, is it possible to identify what is causing this discomfort? To answer the research question, the following sub-questions are considered:

- *What are means of measuring indoor air quality and thermal comfort?*
- *What are known effects of working in an environment with poor indoor air quality?*
- *What are guidelines for indoor air quality metrics?*
- *Are these guidelines met at the VTTI office?*
- *If so, is discomfort reported through surveys even though the models would not indicate that discomfort should occur?*
- *Does the data provide any insight that can lead to concrete recommendations which will improve the well-being among employees?*

4

Research method

After the sensors were assembled and the software was developed, the sensor network was deployed at the VTTI office. Measurements were conducted for an entire month. During this period survey data was also collected. Employees were asked to fill in the survey 3 times a day. In this survey, employees were asked to score the thermal comfort and indoor air quality on a Likert scale. The location of the employee was registered as well in the survey. During the measurement period, the data was also labeled. Although air velocity is not measured with the deployed sensors, air velocity measurements were conducted at the VTTI office during the labeling of the data. Hourly weather data published by the KNMI was also correlated with the measurements.

5 Technology Research

First, it is important to establish which air quality properties exist, how these may affect employee health and well-being and whether these are subject to industrial guidelines. In Appendix A various indoor air quality properties and their effects are outlined. Secondly, it is important to research which of these properties can be adequately measured with accessible sensor technologies. In Appendix C several air quality sensors using different sensing technologies are compared. The insights obtained from these surveys were used for the selection of hardware in the sensors for this project. In Section 5.1 the requirements for the sensor hardware are further developed. Section 5.1.1 compares the various means of communication between sensor devices. Consecutively, Section 5.2.1 describes how the measurements from the sensors can be sent over the network and propagated as an event stream through the systems. Finally, 5.2.2 looks into various means of storing the time series data from the measurements.

5.1. Sensors

Indoor air quality sensors are becoming cheaper and more accurate, making it a viable solution to actively and accurately monitor indoor air quality and control ventilation systems [6, 7]. There is a variety of indoor air quality sensors commercially available, such as the Air Mentor, Netatmo, and Uhoo (shown in Figure 5.1). However, these sensors are (a) limited in their connectivity (e.g. use only Wi-Fi or Bluetooth), (b) make use of a closed system (e.g. the sensor data is only available through the manufacturer's app or API), (c) often do not measure the combination of properties defined in the project requirements (Section 2.1), and (d) do not have the flexibility to expand the system with additional sensors. Therefore, a system needs to be developed comprising a choice of sensors that meet the requirements established in this project.

5.1.1. Communication

Commercially available sensors can be paired to a mobile device through Bluetooth and then added to a local Wi-Fi network, which they utilize to connect to a cloud platform. However, this approach is not very suitable for VTTI. First of all, the sensors cannot use VTTI's existing wireless network due to security concerns. Secondly, due to the relatively low range of Wi-Fi, having to set up a dedicated wireless network for the sensors with sufficient coverage would require the installation of several access points throughout the building. Due to the low range and client-server architecture of Bluetooth only a small number of sensors can be connected to a single device, which is insufficient for the purpose of this project. Both Wi-Fi and Bluetooth also require significant energy, which makes it infeasible to power the sensors using batteries. While battery-powered sensors are not a requirement set by VTTI and there are plenty of wall sockets available in their office, having the option for sensors to run on batteries for a prolonged amount of time will allow for more flexible deployment of sensors throughout the office.

In addition to Wi-Fi and Bluetooth, there is a wide variety of other network technologies available for wireless sensor networks. Usually, these networks communicate to a nearby gateway over radio frequency, and the gateway is then responsible for relaying those messages to the internet. These technologies originate from various industries. For example Z-Wave and ZigBee are often used in home automation as well as industrial sensor networks. In particular the ZigBee network is used a lot in similar Air Quality sensor networks [8, 9, 10, 11, 12]. With the rise of Internet of Things, new network standards have emerged rapidly [13]. The new Blue-

Figure 5.1: The commercially available Netatmo Healthy HomeCoach and Air Mentor indoor air quality monitors

tooth LE standard uses significantly less power. The new Bluetooth standard as the option to operate using a mesh topology, similar to Z-Wave and ZigBee. This remedies its short range and limited number of clients problems, since sensors in a mesh topology connect to each other in a non-hierarchical manner, enabling all sensors in the network to act as a repeater. For example the commercially available bGrid sensor network uses the new Bluetooth Mesh standard. Another new development is LoRaWAN, a low power telecommunication network suitable for long distance communication. LoRaWAN cannot only be used with private gateways, but also on public national networks. LoRaWAN is unique for its very low power draw, which allows sensors to operate on a single battery for months. However, LoRaWAN devices are only allowed to send a relatively short message every once and a while, which restricts the sensor sample rate. All communication technologies mentioned above have a security mechanism built in for encrypted communication between devices. A full comparison between available network technologies is made in Elhadi et al. [14].

For almost all wireless sensor networks there are modules and extension shields available to easily connect them to prototype development boards such as the Arduino. Some development boards even come with communication functionality built in, such as the Particle Xenon BLE+MESH for Bluetooth LE Mesh networks, Arduino MKR WAN 1300 for LoRa or LoRaWAN networks or the Z-Uno for Z-Wave networks.

For this project Bluetooth LE Mesh networks will be used using Particle.io development boards because the additional bandwidth that BLE provides over other technologies such as LoRaWAN allows for a higher sample frequency, which in turn will provide greater flexibility in terms of data analysis options.

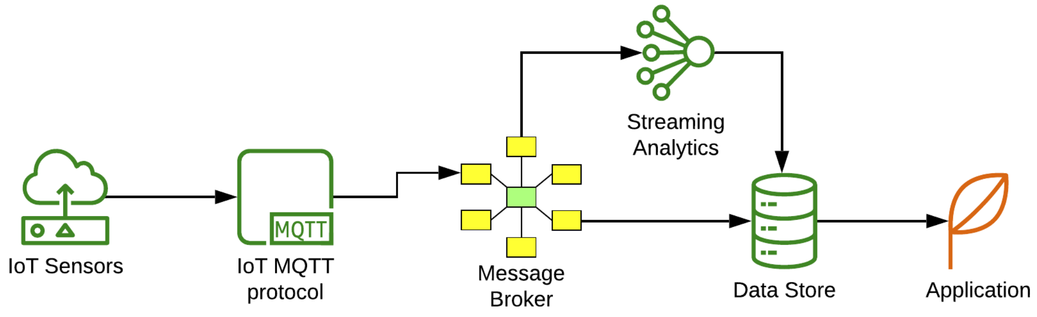


Figure 5.2: A common flow of measurements from sensors through parts of a system

5.2. Component Technologies

The system will contain a variety of aspects: the data flow, the database, and the data analysis. The following sections describe the existing technologies that are considered to be used as a basis for the developed system.

5.2.1. Measurements propagation

Sensors produce a stream of data which has to be propagated through several layers of the system, including the gateway, the network, the database, streaming analytics, notification systems and eventually the dashboard front-end. In essence this stream of data can be sent over a plain network socket connection. However, dealing with a stream of data on this level introduces a variety of problems with regard to durability, scalability (load balancing of messages), availability (failover on disconnect) and the ability to use a more dynamic topology with for example multiple procedures or systems receiving the same message. To address these issues it is common to utilize a message broker that implements the publish-subscribe pattern. The publish-subscribe pattern uses a message queue to which producers can send messages, agnostic of which exact and how many receivers are subscribed to that queue. A common flow of messages through parts of a system is shown in Figure 5.2.

A common publish-subscribe based messaging protocol used in Internet of Things applications is MQTT, which is for example implemented in the freely available and open source message brokers Eclipse Mosquitto, Apache ActiveMQ and RabbitMQ [15]. MQTT is also one of the available protocols for many Infrastructure as a Service (IaaS) providers such as Amazon Web Services IOT, Google Cloud IOT and Microsoft Azure IoT Hub.

The Particle.io Cloud Platform that the sensors connect to (Section 5.1.1) lacks the ability to forward published messages to a MQTT message broker directly. However, one can listen to a stream of events through their API (returned as W3C Server-Sent events), which can then be propagated through the system - possibly by first publishing it to an MQTT message broker. Whilst also lacking support for connecting to an online available (MQTT) message broker, the Particle.io Cloud Platform does have integrations for the aforementioned IaaS platforms from Amazon, Google and Microsoft [16]. Furthermore, there are open source implementations available that allow a Particle.io gateway device to publish messages to a MQTT client [17].

5.2.2. Database

The measurements received from the sensors will have to be stored in a database. Sensor measurements are intrinsically temporal, with data sizes growing fast over time, in particular when sample frequencies increase. Time series databases utilize the temporal nature of the data: by maintaining a temporal order of values, they are able to partition the data by certain intervals, and as a result are able to find data within a certain interval or at the tail of data entries very quickly. Although this can be emulated in relational databases as well (for example PostgreSQL supports timestamp partitioning of tables [18]), performance is often much worse than in a time series optimized database (for example PostgreSQL has to lock all partitions of a table while inserting data, drastically affecting throughput on large datasets [19]). Time series databases also include various time-related functions in their query language, that simplify working with time series. There are many time series database systems available. Currently, InfluxDB, Graphite and Prometheus are popular, open source, time series database implementations [20]. A comparison between these database implementations is made in

Bader, Kopp, and Falkenthal [21].

Time series databases can be connected to a general purpose dashboard and graph composer. A popular tool for this is Grafana, which connects to - amongst others - InfluxDB and Graphite. Using Grafana, one can explore datasets by running queries and plotting the results. These plots can then be combined in a dashboard. For example, Uber is a notable user of Graphite in conjunction with Grafana [22].

The amount of data that will be produced on a daily basis can be approximated by the simple multiplication of (a) the number of sensors, (b) the number of measurement dimensions, (c) length of each dimension in bytes, and (d) the sample rate. As the intention is to deploy 10 sensors each measuring 10 dimensions represented in 4 byte floating point values every 10 seconds, this yields $24 \cdot 60 \cdot 10 \cdot 10 \cdot 4 \cdot 6 = 3.5\,\text{MB}$ per day. From the obviously small amount of data that is produced, it can be concluded that there is no need for a database specialized in dealing with high volumes of data. An increase in the number of deployed sensors, measurement dimensions, or the sample rate will linearly increase the amount of data produced on a daily basis. This must be taken into consideration for future research and development. However, for now it suffices to focus the on ease of use and functionality that a database might provide.

Even though the team members are more familiar with relational database technologies, the better suitability of time series databases is found to be of greater importance. Therefore, time series database InfluxDB will be used in combination with Grafana, where Grafana will be used for data exploration only. Queries will be run on InfluxDB to provide the dashboard with data.

6 Design

This section describes individual components and how they interact with other systems in the system architecture.

6.1. Architecture

The proposed system comprises three main components: the embedded sensor firmware, the back-end system and the dashboard front-end. Each of the components requires software to be developed. The software needed for each component is outlined in the sections below.

6.1.1. Device firmware

The device firmware will be programmed onto the device's microcontroller. The firmware will be responsible for controlling the sensors, taking measurements and offloading these to the cloud platform. Each of the sensors exposes a programmable interface over an I2C wire or UART serial that allows for example control (initiate measurement), calibration (baseline correction), specifying parameters used for normalization (correct the error introduced for elevated temperatures or atmospheric pressure), retrieving diagnostics and taking measurements. The firmware will use this programmable interface to interact with the sensor. For each measurement the firmware will enable the sensors, take measurements and transmit the results to the cloud. The firmware also implements remote procedure calls (RPC) for remote calibration of sensors (useful for cross-calibration).

The firmware is built on top of Particle.io's Device OS operating system. This operating system handles the mesh network connectivity (such as device registration, security and network topology) and connects this network to the Particle.io Device Cloud. This Device Cloud provides the possibility to (a) call a function (RPC), (b) publish or subscribe to events, (c) read or write variables, and (d) perform over-the-air (OTA) firmware updates.

6.1.2. Dashboard front-end

The measurements taken from the sensor network will be displayed in a dashboard. The dashboard will be accessible online. The dashboard will facilitate several basic use cases: (a) view historic trends, (b) view real-time data and suggestions, and (c) alert notifications. The dashboard will also allow employees to report discomfort or submit feedback through a survey. This survey data is continuously analyzed and reported to the manager in the form of regular digests. The use cases are depicted in Figure 6.1.

The dashboard is designed to be user-friendly and productive. Therefore, as much information is visible as possible in the primary screen and not hidden behind menus. The interface comprises four main components: (a) the actual indoor air quality measurements, (b) a heat map showing the sensor locations and predicted thermal comfort and perceived air quality, (c) a line chart showing historic trends, and (d) a form for the thermal comfort survey. A wireframe of the dashboard is depicted in Figure 6.2.

The actual indoor air quality measurements will display the actual temperature, CO_2 concentration, relative air humidity (RH) and Predicted Percentage Dissatisfied (PPD) (see Appendix B). The floor plan allows employees to easily identify the closest sensor node. By clicking one of the sensors on the floor plan, this

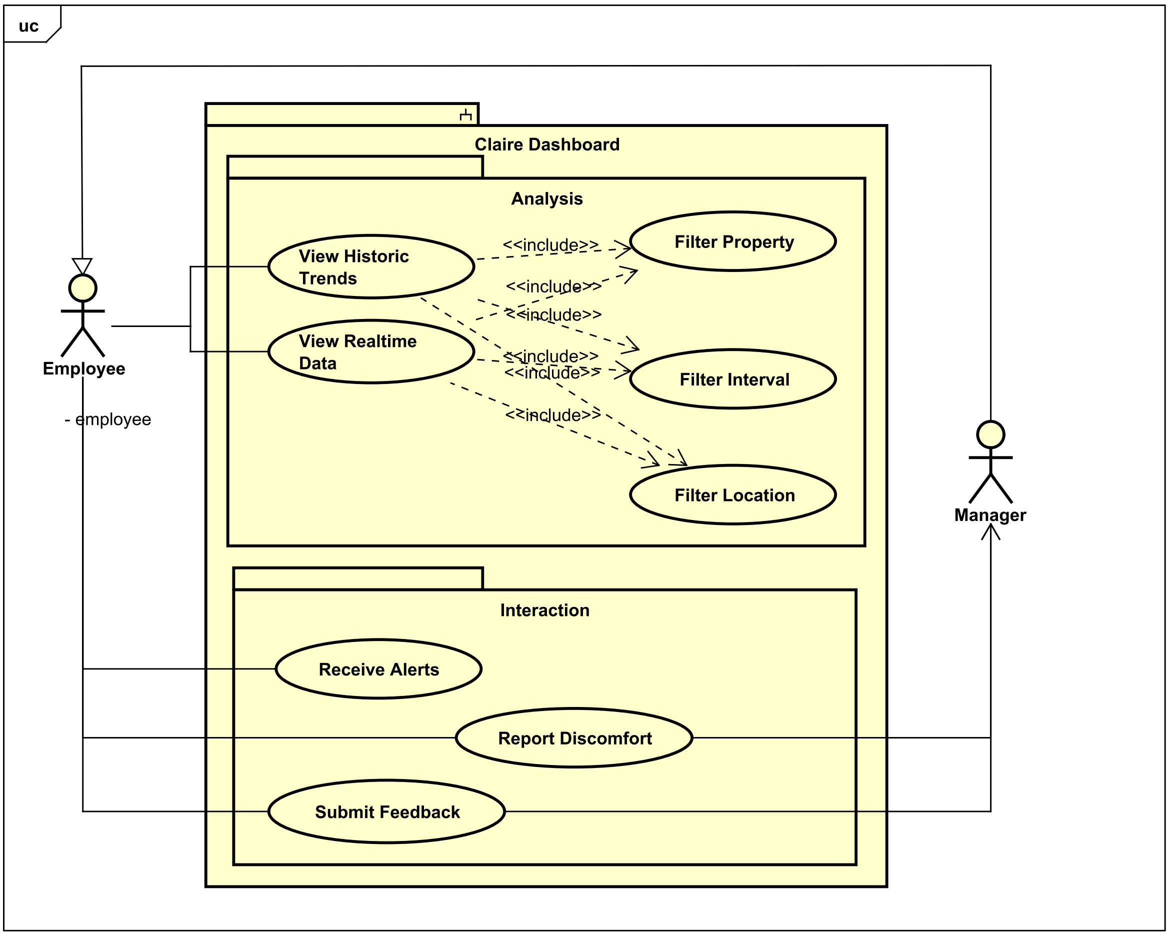


Figure 6.1: Use Case Diagram

sensor will become selected and its values will be presented. If no sensor is selected, the average values throughout the building will be shown.

The section dedicated to historical trends is not intended as a full fetched data analysis tool. Instead, it should allow employees to verify elementary indoor air quality hypotheses such as simple trend comparisons between different hours, or days of weeks. More detailed data analysis will be provided in the form of regular digests.

The thermal comfort survey is a simple tool to collect soft data on the perceived air quality. The survey consists of two simple questions: one score for the thermal comfort (based on the Predicted Mean Vote (PMV) Likert scale) and one score for the perceived air quality (another Likert scale). Both scores are provided through setting a slider. When the survey is submitted on a desktop device, the employee also has to select the closest sensor device. For mobile devices where the application has access to the Bluetooth API, this device will be derived from the signal strength to the surrounding devices. This simplifies taking part in the survey because it eliminates the step of selecting the nearest sensor node.

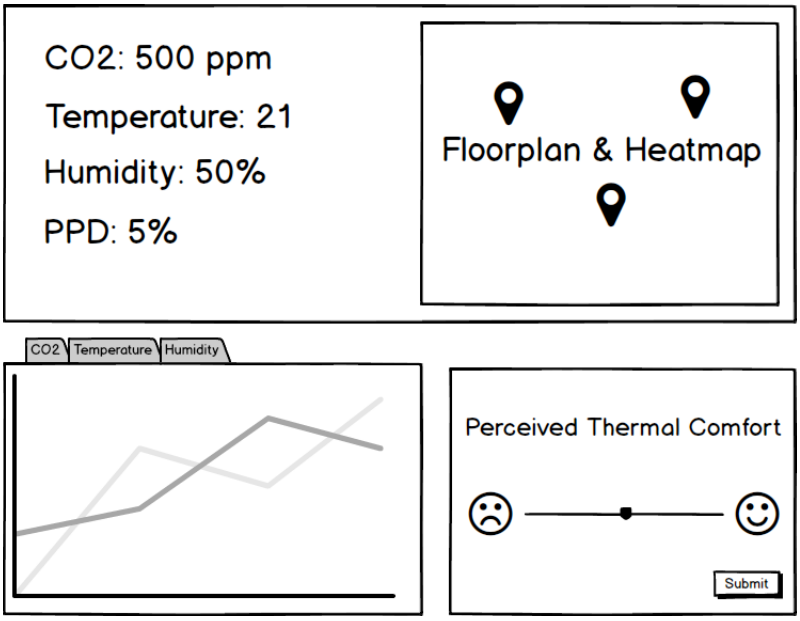


Figure 6.2: Claire Dashboard wireframe. Top: Actual measurements, bottom left: historical trends, bottom right: thermal comfort survey.

6.1.3. Back-end system

The back-end provides access to the measurement data stored in the database to the front-end. The back-end is also responsible for checking whether a user is authorized to see particular data and ensuring the user is presented the right data, in case there are multiple buildings connected to the system. The communication between the front-end and back-end follows the request-response model: the back-end exposes a REST API. Communication through this API will be in the JSON content type.

The back-end also exposes a WebSockets endpoint. Clients may subscribe to this WebSocket to be notified about new measurements immediately and update the dashboard accordingly in real-time.

Information that may be retrieved from the back-end includes:

1. Floorplan geometry;
2. Actual sensor locations;
3. Actual sensor measurements;
4. Query historical sensor measurement data.

Clients may also submit survey and feedback data. The back-end will make sure this data gets persisted into the database. Besides communicating with the front-end, the back-end also performs any required post-processing for incoming measurements.

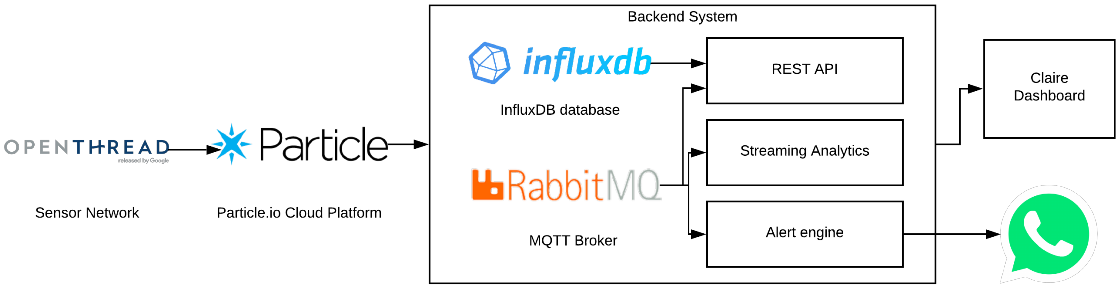


Figure 6.3: Data Flow Overview

6.2. Choice of programming language and frameworks

Several programming languages were considered for this project: Java, C++, JavaScript and Python. For the embedded software code programming in C++ is the most practical option, because all example code from the hardware is provided in C++ and the Particle.io development boards only support firmware written in C++.

As the front-end dashboard is developed as a web application, where JavaScript is the standard, which all team members have experience with, there was no reason to consider any other programming language for the front-end. The dashboard will be developed as a responsive single page application. For the data processing and back-end technology, however, there were many options. For this decision personal experience with the programming language was taken into consideration, as well as the availability of open source frameworks and libraries to connect to the technologies that will be used. It was also preferred to use the same programming language throughout the entire stack as much as possible, and the team members were open towards learning a new programming language if this could reduce the total number of languages involved in the project. Because of the large portion of data analysis in this project, and the popularity of Python among data scientists, the decision was made to use Python as the programming language for all other software developed.

7 Implementation

This section describes how the designed system is implemented. In Section 7.1 the implementation of the individual sub-systems is described, Section 7.2 outlines the implementation of the predictive analysis, Section 7.3 describes how servers and development systems are managed and provisioned, and Section 7.4 describes how the implementation was changed based on the feedback received on the code submission to the Software Improvement Group (SIG).

7.1. Components

This section describes the implementation for each of the individual sub-systems. Section 7.1.1 describes the specifics of the sensor firmware, Section 7.1.2 describes the implemented cross-calibration mechanism, Section 7.1.3 describes the front-end implementation, Section 7.1.4 describes the back-end implementation, and finally, Section 7.1.5 and 7.1.6 describe how the measurement data is processed and stored in the system.

7.1.1. Device firmware

Embedded microcontroller firmware usually comprises fairly simple programming routines. Microcontrollers usually only have a single core and no multi-threading capabilities, so one can only do a single operation at a time. This usually boils down to a single procedure that gets invoked in a continuous, looping manner. It is, however, possible to suspend the main program using a system interrupt. Such an interrupt can, for example, be triggered by a timer or an input signal change. The main program will be halted, the appropriate interrupt service routine (ISR) will be invoked, and after returning from the ISR the main program will resume. Because ISRs will temporary disable any further interrupts, ISRs should be fast and are not intended to perform blocking operations or operations that are dependent on other ISRs (asynchronous operations).

The sensors connect to the microcontroller over I2C wire and UART serial communication buses. Messages received on these buses (in this case the measurements) are also handled by ISRs. The ISRs write the messages to a buffer for later use. Publishing a message to the cloud platform is an operation that intrinsically requires communication (IO), as such is reliant on system interrupts being enabled, and therefore cannot be performed within another ISR. Therefore, after the measurements have been collected through their respective ISRs, it is the main loop / program that is responsible for publishing the aggregated data to the device cloud. After all measurements have been taken, the device will turn off the sensors and wait a certain amount of time in order to save power while still accomplishing the desired sample frequency. As a result, the same frequency is used for all measurement dimensions. The process of taking measurements is visualized in Figure 7.1.

7.1.2. Cross-calibration

The CO_2 sensors use an internal automatic baseline correction, where the lowest point within a certain measurement period is expected to be equivalent to a concentration of 400 ppm, as this is the Earth's current atmospheric concentration. This automatic baseline correction can lead to unexpected calibration events and will also lead to various sensors using a different baseline. Having multiple sensors installed within the same room allows for manual cross-calibration to be performed. After people gradually leave the office between 17:00 and 18:00, the CO_2 concentration drops gradually until it finds an equilibrium throughout the

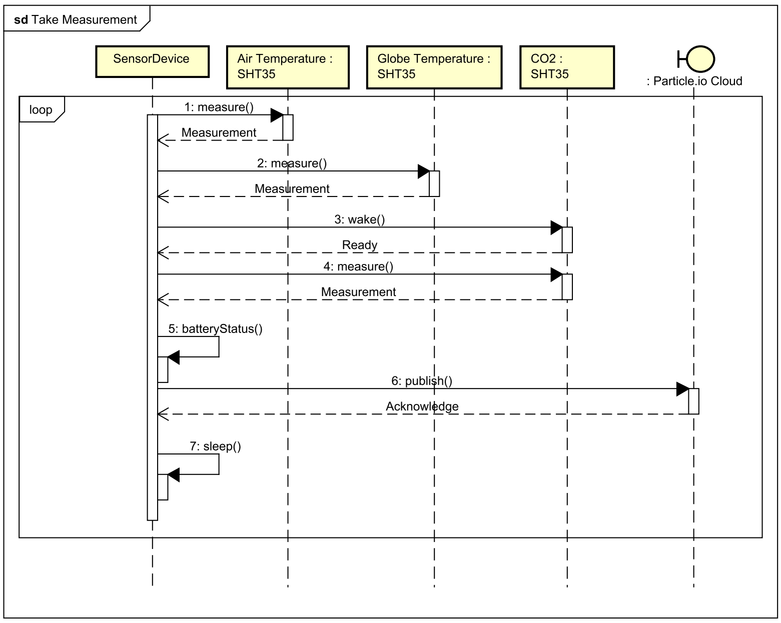


Figure 7.1: A sequence diagram that shows how the various measurements are taken and then published to the cloud

building, with the lowest concentration being reached around 19:00 after which it remains stable until the ventilation system turns back on at 6:00 (see Figure 7.2). The measurements within this time window are used to detect whether sensors have drifted (i.e. whether they report higher values than expected). If this is the case the drifted sensor is re-calibrated: the back-end system will trigger a remote calibration with a specified concentration offset.

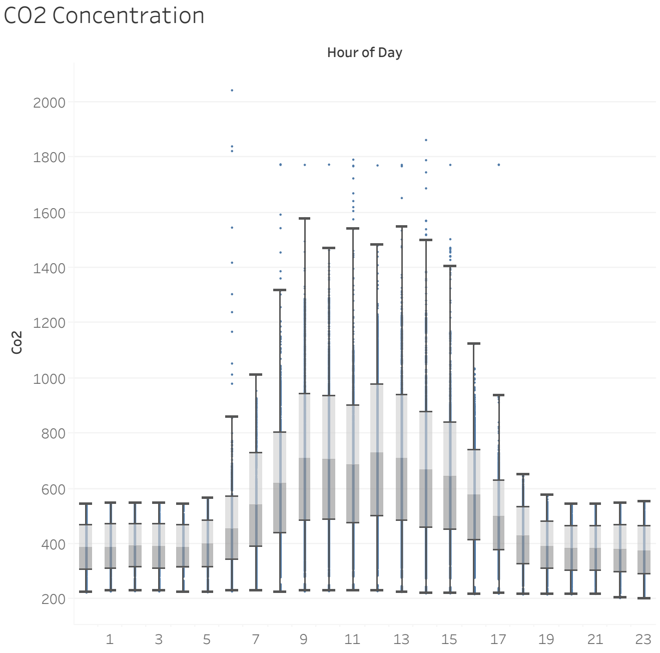
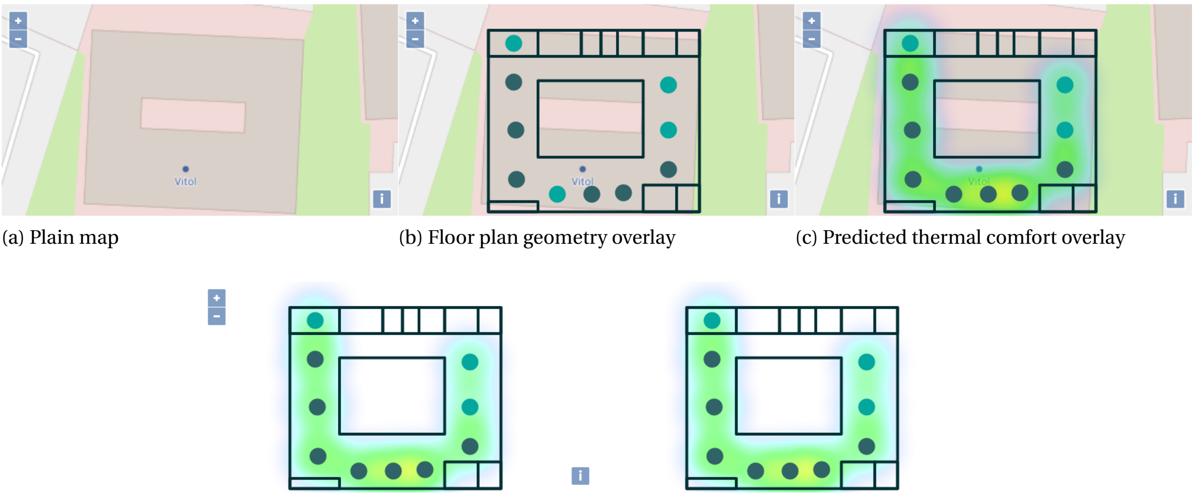


Figure 7.2: Average CO_2 concentration per hour of day

7.1.3. Dashboard front-end

The dashboard front-end is built on top of the JavaScript templating library React. This has the potential benefit that some of the developed code can be reused when potentially using React Native for building smartphone applications in the future - React Native being React's counterpart for smartphone app development. The dashboard fetches data from the back-end both through the means of making REST API requests to the back-end and being continuously updated through messages received over a WebSocket connection. The state of these actual sensor data has to be managed locally in the frontend. Redux is used to manage this and ensures that all UI components referencing this state will consistently show the updated data. There are various JavaScript libraries available for rendering charts in the browser. Most notably are D3, Chart.js, and Highcharts. Chart.js was chosen as chart library for the dashboard because there exists a specific Chart.js wrapper library for React, which alleviates integration between the two frameworks.

To overlay sensor values on an image of the floor plan of VTTI, OpenLayers was used, a map rendering library, with a custom made map that represents the floor plan. The map can then be enhanced with markers representing the sensors, or for example, a heat map overlay showing the temperature throughout the building (see Figure 7.3).



(a) Plain map

(b) Floor plan geometry overlay

(c) Predicted thermal comfort overlay

(d) Removal of the map background

(e) Removal of the map controls

Figure 7.3: Construction of the heatmap.

One of the challenges was to generate a GeoJSON file with the geometry of the floor plan. The approach used was to draw the geometry based on a picture of the emergency evacuation floor plan. However, this drawing was not perfectly square and the walls of the rooms did not align with the walls of the building. To solve this, a grid of coordinates was made to which each of the coordinates was then snapped. Another challenge was that the coordinates produced by the drawing tool were in a different coordinate system (WGS) than the map (EPSG:4326). The coordinates were transformed to the proper coordinate system using the popular library Proj4.

7.1.4. Back-end API

The back-end API is implemented using the Flask Python library. Flask makes building request-response APIs very easy in Python. Flask is also capable of WebSocket communication through the Flask-Socket.io extension.

For authentication JSON Web Tokens (JWT) are used. These are self-contained authentication tokens, which are signed using a private key stored on the server. As a result, these authentication tokens can only be produced by the server itself. It is a safe mechanism for authentication that requires practically no session state to be stored at the server. JWT tokens have a built-in mechanism of session invalidation: upon creation of the token, an expiration date can be set. The Flask-JWT is used as the implementation of JWT on top of the back-end API. Flask-JWT implements a request interceptor: through means of adding a simple annotation to a resource method, authentication will be mandatory and the implementation will look for a valid JWT token to be provided with the request.

7.1.5. Data ingest

Data enters the system as a stream of Server-Sent events. This stream is consumed by the data collection service. This service is responsible for computing derived properties, persisting the measurement in the database, and propagating the measurement through the system. This propagation includes broadcasting the measurement in a specific WebSocket room topic. Rooms are a group of WebSocket listeners that subscribe to a set of specific messages. In this case, every tenant can subscribe to their own room. By broadcasting the measurement to this room, it is ensured that all active dashboards are updated, whilst still isolating measurement data from different tenants. The process of collecting messages is shown in Figure 7.4.

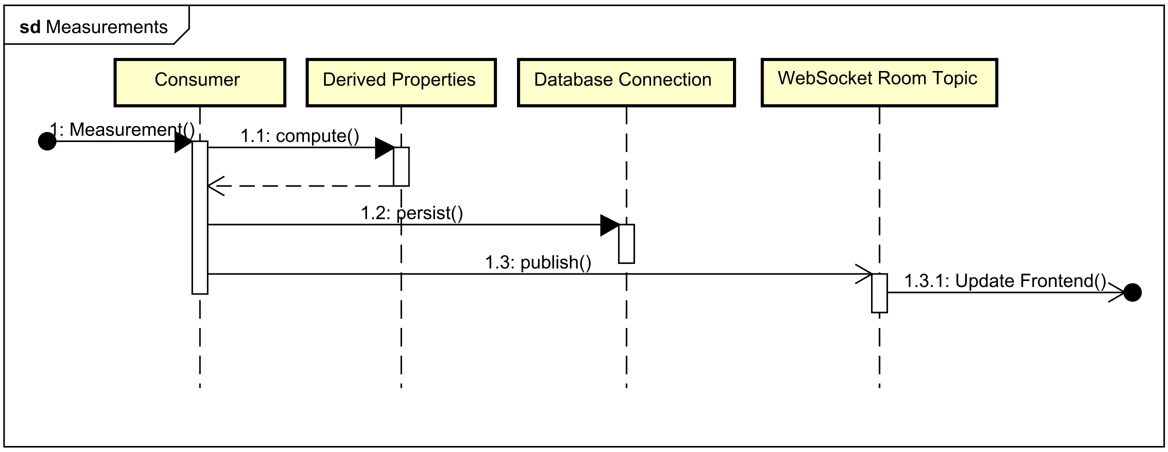


Figure 7.4: Sequence diagram of measurement collection

7.1.6. Database data model

The data is stored in InfluxDB. The data model comprises five entities: measurements, data collected through user surveys, feedback, the KNMI weather data and an entity that contains the data labels. The entity diagram is depicted in Figure 7.5.

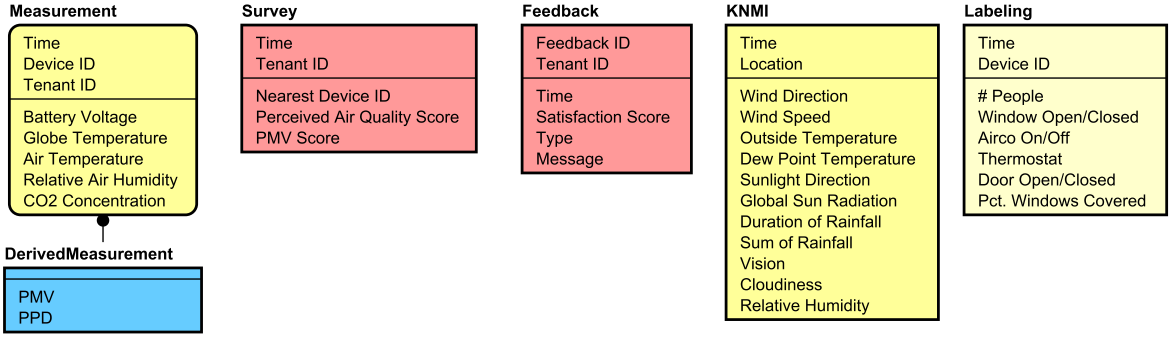


Figure 7.5: Entity diagram

7.2. Prediction models

7.2.1. Smoothing and derivative

One of the goals of this project is to notify employees when CO_2 concentrations are rising. However, when waiting until the threshold is already reached, the notification is already too late and it will be difficult to ventilate the CO_2 away. Computing the derivative of the CO_2 concentration provides an idea of how fast the CO_2 concentrations are rising, which is a good predictor for how many people are in the room. The second derivative can then be thought of as an indicator of how many people enter the room. Given a certain rise of CO_2 concentration over a particular amount of time, a likelihood that the concentration will approach the threshold can be computed. The exact value for the second derivative needs to be optimized for the size of the room, the amount of ventilation in that room and the measurement interval. After this derivative is found for a particular room, it will be configured in the database. Furthermore, the data needs to be smoothed, for example using a moving average, to prevent false alerts for outliers (for example, people blowing into the sensor). The moving average and second derivative are computed as one of the streaming analytics. Therefore this data can be used in real-time to automatically generate alerts which CO_2 levels are rising. Figure 7.6 shows the CO_2 concentration of a meeting room during the day together with the computed second derivative.

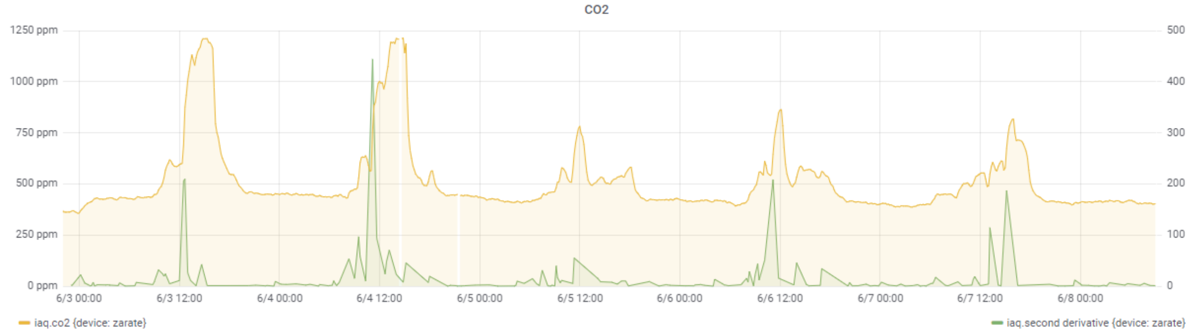


Figure 7.6: CO_2 and its second derivative for a meeting room. The spikes in the chart clearly correspond to larger meetings, and larger spikes are seen when higher concentrations are reached in the room consecutively.

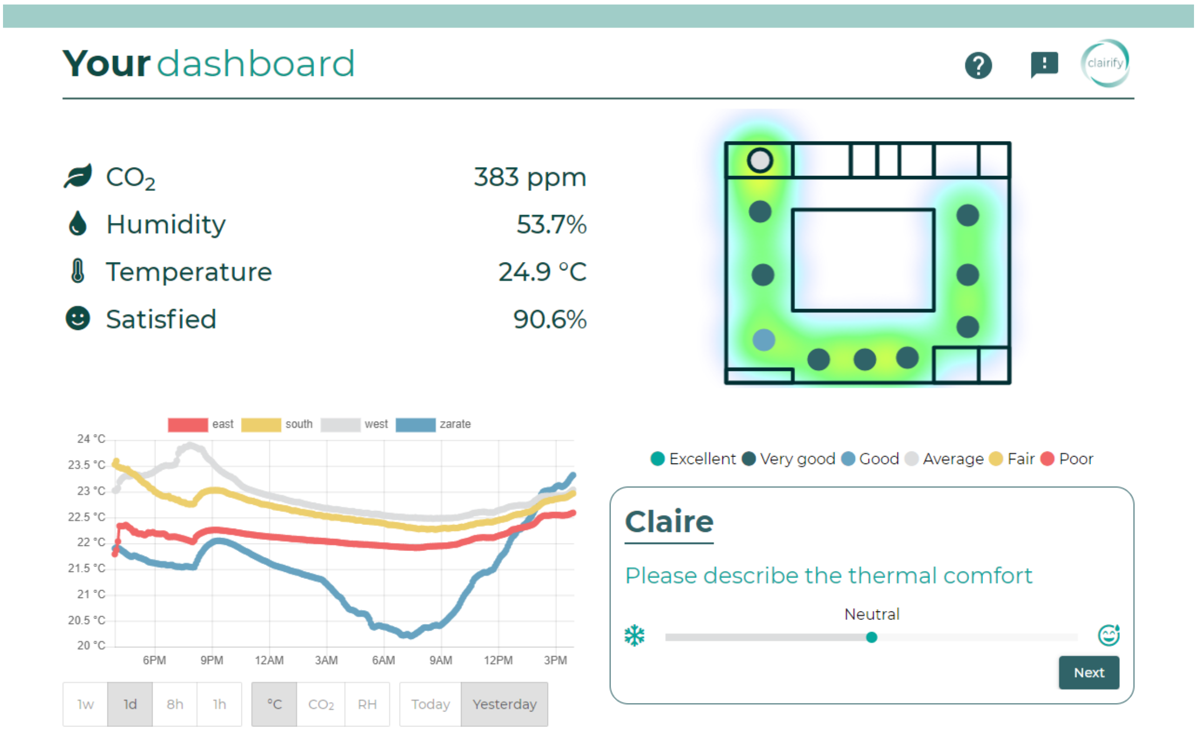
7.3. DevOps and containerization

The back-end software runs in a Python application servers and is dependent on the availability of an InfluxDB database instance and a (RabbitMQ) MQTT server. The front-end software needs to be hosted from a web server such as NGINX or Apache Httpd. These are quite a few services to roll out and deploy. To make deployment easier and to make software behave the same way on a developer's machine as well as a production machine, the vitalization software Docker will be used. Docker virtualizes virtual machines with a shared kernel, eliminating much of the virtualization overhead. Furthermore, Docker has the ability to build containers from a simple setup script. Container snapshots can also be exchanged through Docker repositories. Docker containers run isolated from the host system, meaning that if a container gets corrupted, this can easily be cleaned up and won't necessarily affect the host system or any other containers running on the host system.

7.4. Software Improvement Group Feedback

During the project the code was submitted for evaluation by the Software Improvement Group (SIG). The findings of SIG are included as Appendix G. The scores for the development were found to be around the market average. SIG identified Unit Interfacing and Unit Size as areas for improvement. The code responsible for the low score on unit interfacing was the implementation for the PMV computation (full equation listed in Appendix B), which takes a lot of parameters. Both JavaScript and Python have the possibility to define default values for unset parameters. These parameter defaults were used to provide constant values to the computation. The PMV implementation was later swapped out for an existing open source implementation that was available through the package manager. This improves the maintainability of the code: there is less code that has to be maintained, and any potential bugs with regard to this specific computation can now be addressed and resolved within the open source community. The unit size was reduced by breaking out some logic into separate utility methods. The unit size in the Python code however was not addressed: the code was merely formatted in a sparse way to improve readability (and increase maintainability) while the fragment only contained a couple of actual code statements. SIG also addressed the lack of tests to be an issue. Because the SIG submission was at a very early stage of development, most of the code actually concerned little code snippets to integrate with a particular sensor or data service. In essence, these were all small tests. It was also quite hard to test some of these components in an automated fashion because they were written in C++ and dependent on Particle OS system calls. The more fundamental components of the systems (such as the back-end and dashboard front-end) are more suitable for testing, however, these were not developed at the time. As the project progressed, more tests for all the individual subsystems were added to the project gradually.

7.5. Final product



Your dashboard

CO_2	383 ppm
Humidity	53.7%
Temperature	24.9 °C
Satisfied	90.6%

● Excellent ● Very good ● Good ● Average ● Fair ● Poor

Claire

Please describe the thermal comfort

Neutral

Figure 7.7: Final dashboard

8
Prediction model

Although the derivative method works well, it requires tweaking for a specific sensor location and the time window used. This makes the method difficult to apply to other sensors within the network or even at different locations. Therefore, a more generic approach is required. A neural network might be able to learn the characteristics of an irregular raise of CO_2 concentrations. A neural network can learn to predict the progression of a time series. When the neural network predicts a concentration above the threshold, this would mean that an alert has to be generated.

8.1. Neural networks

One of the advantages of neural networks over the derivative method is that they might be able to learn patterns unrelated to the derivative of the measurements, and use this additional information to make better predictions. Another advantage is that a neural network does not require any specific domain knowledge or manual tweaking for individual sensors or locations, as the network is able to extract the relevant features automatically [23]. Neural networks do have a couple of disadvantages. First of all, neural networks require a large amount of data to train on, or otherwise risk overfitting on the data. When a network overfits it learns the specific mappings in the training data set, but performs poorly on validation data and unseen data, as the model is unable to generalize well [24]. Furthermore, the hyperparameters (e.g. the number of epochs to train, the number of layers, and the number of neurons) of a network have a significant effect on its performance [25], and can require a lot of tweaking to get satisfactory results.

Recurrent neural networks (RNNs), and in particular long short-term memory networks (LSTMs) are often used in time-series forecasting, due to their ability to capture temporal dependencies [26, 27, 28, 29, 30, 31]. RNNs contain recurrent connections that propagate previous information, enabling the network to base its decisions on the current data as well as past data [27]. However, RNNs have no way to control how (long) the past states' information is propagated, and as such suffer from exploding and vanishing gradient problems. When suffering from the exploding or vanishing gradient problem, the updates to the network's weights respectively become very large or infinitesimal, making the network very unstable [32]. As a result of this instability, RNNs are inhibited in their ability to learn long-term dependencies [26].

LSTMs are RNNs that deal with this problem by introducing input, output, and forget gates to the cells that control the cell's state and the propagation of previous states' information, enabling the LSTM to control which past states to forget [26]. As a result, LSTMs are able to capture both short- and long-term dependencies, which makes them capable of forecasting time-series data [27].

Other neural networks, such as multilayer perceptrons, are not able to capture these temporal dependencies at all, and are as such outperformed by RNNs when it comes to time-series prediction [31]. Convolutional neural networks, on the other hand, are able to identify short-term dependencies [29].

Due to the nature of the measurement data, it is deemed necessary to be able to learn long-term dependencies as well as short-term dependencies. Therefore, due to the LSTM's ability to capture both long- and short-term temporal dependencies in contrast to other neural networks, the decision has been made to use an LSTM for predicting measurement data.

8.2. Model design

This section describes how the prediction model will be designed in terms of what the training process will look like and how the model will be applied in the final system.

8.2.1. Training the model

Training the prediction models consists of several steps: acquiring the data set, preprocessing the data set such that it can be used by the model, constructing the model, and then training the model. Each of these steps is discussed below.

1. The data set to use for the model will be retrieved from the database and stored in a suitable format that can be loaded by the data processing framework.
2. The data set will be preprocessed such that it adheres to the format expected by the Tensorflow model. The data will be preprocessed according to the following steps:
   (a) The data is resampled in order to limit the number of redundant data points. The current sampling frequency (10s) is rather high, and using all data points will slow down the model's training process and performance, while not adding anything meaningful to the predictions.
   (b) The data is split into separate data sets for training, validation, and testing. The test data should be kept entirely separate during the training to ensure that no training occurs on any test data.
   (c) Each resulting data set is split into several fragments using a sliding window approach in order ensure there is sufficient data for the model to train on. The fragments are made after the data is split to avoid any overlap between training, validation, and test data. The size of the windows is determined by the time span used for training data and the time span that should be predicted.
   (d) The data is normalized. Normalization takes place after splitting the data sets, as the data may only be normalized based on training data characteristics. This is to make sure that the test data is completely separated from the training process. The normalization ensures that the values are small and within a limited range, which is necessary to make converging easier for the model. Using the original (large) values may result in large network updates that make it difficult for the model to converge. Additionally, normalization may also result in faster model training since it is easier to converge.
   (e) Missing data is handled appropriately, either by imputing missing data points or not using fragments with missing data points at all.
   (f) For each data set (i.e. training, validation, and test data set), the data fragments will be split into input and target data fragments. The model will train on the input data to predict the target data.
3. The prediction model is constructed by specifying the layers, input and output data shape, and other remaining parameters (such as the loss function to be used, and for how many epochs the model should be trained). The optimal model layout and parameters need to be established by comparing a number of different models.
4. The model is trained by providing it with the training and validation data sets.
5. The test data set is used to assess the model's performance on data it has never seen yet.
6. Finally, the model is saved in its entirety, including the training and validation losses and test results for future use and comparison between models.

8.2.2. Applying the model

Applying the trained models to real-world data in the application will happen as follows:

1. The data to use for the prediction is retrieved from the database directly.
2. The data will be preprocessed as follows, similar to the process during training:
   (a) The data is resampled.
   (b) The data is normalized.

   (c) Missing data is imputed or discarded.

3. The model is provided with the preprocessed data and outputs a prediction.
4. The predicted data is denormalized to convert the values back to the original measurement unit.
5. The predicted data is then propagated for further use (e.g. for display in the frontend).

8.3. Neural network design and construction

The RNN model is constructed using Tensorflow's implementation of the Keras API specification. In order to accelerate the training process, the decision was made to use the Tensorflow package with GPU support based on NVIDIA's CUDA Toolkit. Keras includes a variety of RNN layer implementations, including RNN, simpleRNN, LSTM, and CuDNNLSTM. The CuDNNLSTM is an LSTM implementation optimized for GPU use, resulting in significantly shorter training times compared to the normal LSTM implementation. Therefore, the model was constructed using CuDNNLSTM layers.

A number of models were constructed in order to find a model with sufficient prediction performance, using the CO_2 data to train the models. To be able to determine whether the performance is sufficient, and whether using an LSTM has added value over some simpler models, a very simple prediction method was implemented to use as a baseline. This method uses the input window's mean as the predictions, and the resulting mean absolute error (MAE) of these predictions is used to compare with the MAE of the trained LSTMs.

Using the CuDNNLSTM layer as a basis, models were created with varying numbers of layers with varying numbers of neurons, resulting in everything from simple single layer LSTMs to complex multilayer LSTM Autoencoders. For further comparison, variations were made for other parameters as well, such as loss functions, window sizes, and resampling intervals. All models with their corresponding parameters, training history, evaluation results, and layer diagrams can be found in Table H.2, Figure H.1, Table H.3, and Figure H.2 respectively. Model 19 was found to have the lowest mean absolute error in all cases, with an MAE of ~4.4 ppm on the test data set. As such it is deemed to be the best model to use for the prediction of CO_2 levels. As can be seen in its training history in Figure H.1s, after about 125 epochs the model seems to start overfitting, so training was stopped at that point for the final model, a technique sometimes referred to as early stopping.

The search for the best model was done using the CO_2 measurement data, as predicting the CO_2 values had the highest priority due to CO_2's large variability. Due to time constraints the individual models were not each tested on the remaining measurement data. Instead, models were trained using the same parameters and layout as model 19 on the data for humidity, air temperature, and globe temperature, to create prediction models for those measurements. This unfortunately means that suboptimal models might be used for the predictions of those measurements. However, all trained final models beat their baseline in terms of MAE, as can be seen in Table 8.1, and as such they are deemed to be performant enough.

The final models and their evaluation results and training history can be seen in Table 8.1 and Figure 8.2, and a diagram of the model layers in Figure 8.1.

Although a large variety of models was constructed, it has to be noted that the chosen final model is likely not the optimal model for the predictions of measurement data. The search space for the hyperparameters is simply too large to be able to do an exhaustive search for the optimal set of parameters. However, with some more time and more hyperparameter tweaking it should be possible to find a model that performs even better than the current one, even though it might not be an optimal model either. Another aspect that can be improved is the amount of data used to train the neural networks. The models were trained on a relatively small data set, comprising of measurement data spanning about three to four weeks. This is very little data for a model to generalize on, and as such better performance is to be expected when training the models again at some point in the future when more data has been gathered.

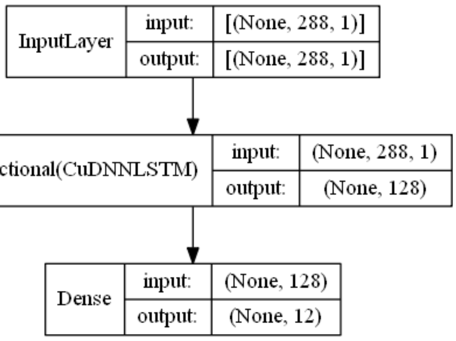
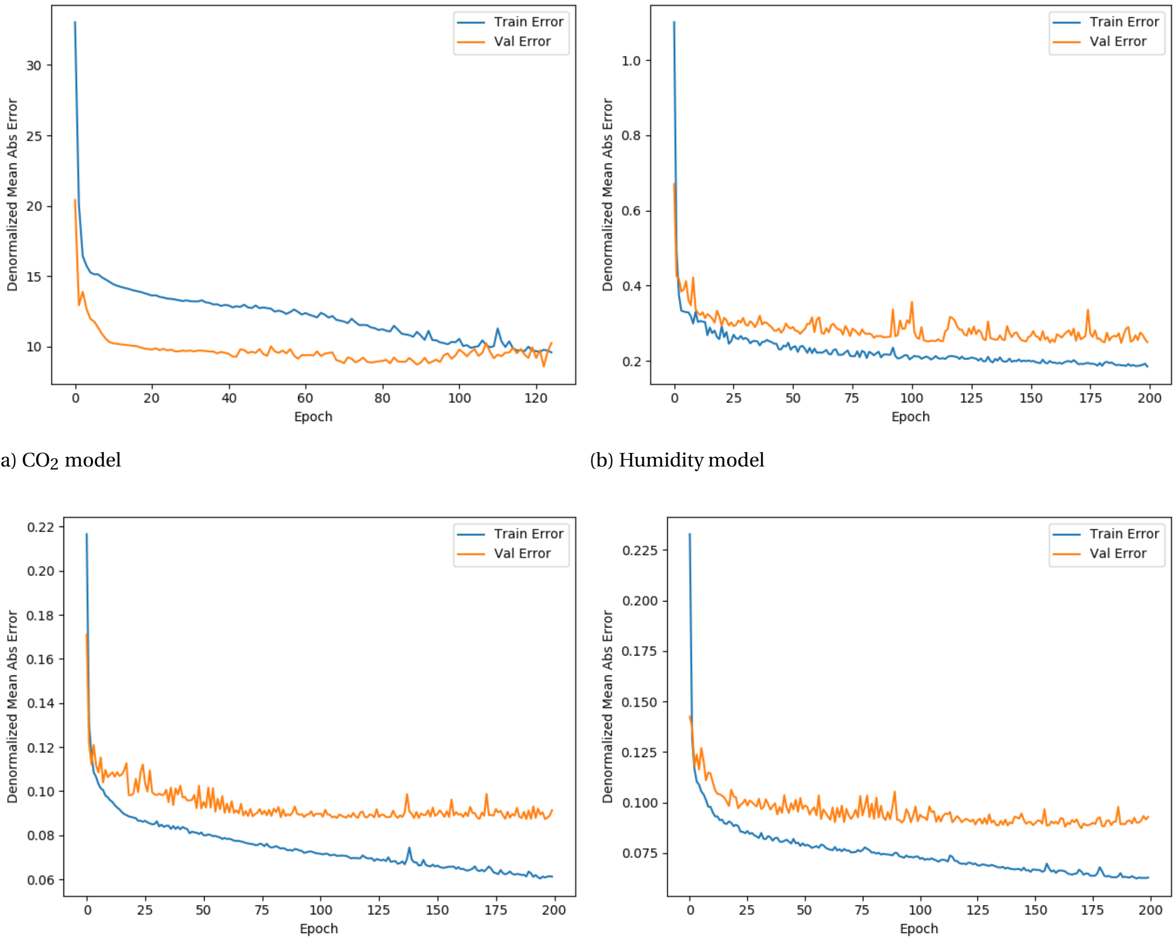


Figure 8.1: Diagram of the final LSTM

Table 8.1: Final evaluation results for each model

Model	Loss	Normalized MAE	MAE	Normalized MSE	Baseline normalized MAE	Baseline MAE
CO_2	0.0362	0.0362	4.6155	0.0050	0.5602	71.5220
Humidity	0.0598	0.0598	0.3512	0.0084	0.6399	3.7586
Air temperature	0.1241	0.1241	0.1103	0.0460	0.6038	0.5365
Globe temperature	0.1297	0.1297	0.1200	0.0475	0.6054	0.5601



(a) CO_2 model

(b) Humidity model

(c) Air temperature model

(d) Globe temperature model

Figure 8.2: Training and validation losses for the final models

8.4. Model implementation

This section describes the implementation details of the model's data preprocessing and the implementation in the final system.

8.4.1. Data preprocessing

The data preprocessing was done using the widely used `pandas` data analysis library for Python. Pandas DataFrames simplify time series data analysis and manipulation as they provide a large variety of operations on time-indexed data, such as sample frequency conversion, filtering, joining data sets, and calculating moving averages based on time windows.

The data processing is implemented as follows:

1. All available measurement data is retrieved from the InfluxDB database and saved to a comma-separated values (csv) file, as this is the file format supported by pandas to read data into DataFrames.

2. The saved data is loaded into a pandas DataFrame (using the measurements' timestamp as an index for easy resampling) for further processing.

3. The data is then preprocessed separately for each device:

   (a) The data is resampled to a frequency of 5 to 15 minutes (varying resampling frequencies were used between models for comparison) using the mean of that time window. Resampling to larger time spans might result in too much information loss, while using smaller time spans might result in too much noise in the data.

   (b) The data is split into training, validation, and test data sets using a ratio of 70:20:10. The data is split in place, meaning that the first 70% of the data will be used for training, the following 20% for validation, and the final 10% for testing. Given that the data is time series data, no shuffling of the data occurs in order to prevent the model from learning on data in the future.

   (c) The data is split in windows using a sliding window that slides over the dataset and creates a data set consisting of windows containing (in case of most models) 25 hours of data.

   (d) All training, validation, and test data sets are accumulated into larger data sets comprising all devices' data.

4. The data is normalized based on the training data set's characteristics only, since the test data may not be involved in the training process in any way. The data sets are normalized using the training data set's z-score, such that the resulting data set's mean is equal to 0 and its standard deviation equal to 1. (Although in practice the validation and test data sets' mean and standard deviation will be approximately those values, since they do not contain the same exact data as the training data set).

5. Windows with too many missing data points are discarded (i.e. windows where more than 30% of data is missing), including windows with missing data in the last section of the data (the data to be used as target data). Not all windows with missing data are discarded as the model should learn to deal with missing data [33], given the expectation of missing data when using the model in the real-world application due to the occasional sensor failure or lost measurements. Otherwise, when faced with missing data in the application, the model will not know what to do with the missing data [33]. The remaining missing data points are imputed by replacing the data points with 0, which corresponds to the mean of the data set. Other approaches suggest filling in the missing data points by using a neural network to predict the missing data. However, with the relatively little data that has been gathered at this point, and the occasional large gaps in the data, this approach was deemed to be unsuitable at this point in time. Training a neural network to fill in the missing data point does not only take extra time, but with the little data currently collected also might do more harm than good to the data set and thus the subsequent training of the prediction model.

6. Finally, each window is split into input data and target data to create an input data set and a target data set (for the training, validation and test data sets each).

8.4.2. Model application

The models consist of CuDNNLSTM layers, which are only GPU compatible and are therefore not suitable to use on the current back-end server directly. In order to solve this, the models are first converted back to CPU-compatible models by constructing a model with the same layers and parameters, only replacing the CuDNNLSTM layers with normal LSTM layers, and subsequently transferring the CuDNN model's weights to the new CPU-compatible LSTM model. These models are then saved and used in the back-end. This conversion process does not affect the performance of the models, as the CuDNNLSTM is simply a GPU-optimized LSTM implementation and works fundamentally the same as a normal LSTM layer, the only difference being the accelerated training on GPUs.

The trained LSTMs are then applied as follows:

1. The trained model and the training data characteristics are loaded.
2. Data from the last 24 hours is retrieved from the InfluxDB database and loaded into a pandas DataFrame directly.
3. The data is preprocessed as follows, similar to the process during training:
    (a) The data is resampled to a sampling frequency of every 5 minutes, taking the mean of each 5-minute window.
    (b) The data is normalized using the mean and standard deviation of the original training data set.
    (c) Missing data is imputed.
4. The model is provided with the preprocessed data and outputs the predictions for the next hour as 12 data points.
5. The predicted data points are denormalized to convert the values back to the original measurement unit.
6. The predicted data is then pushed over the WebSocket for further use (e.g. for display in the frontend).

9 Results

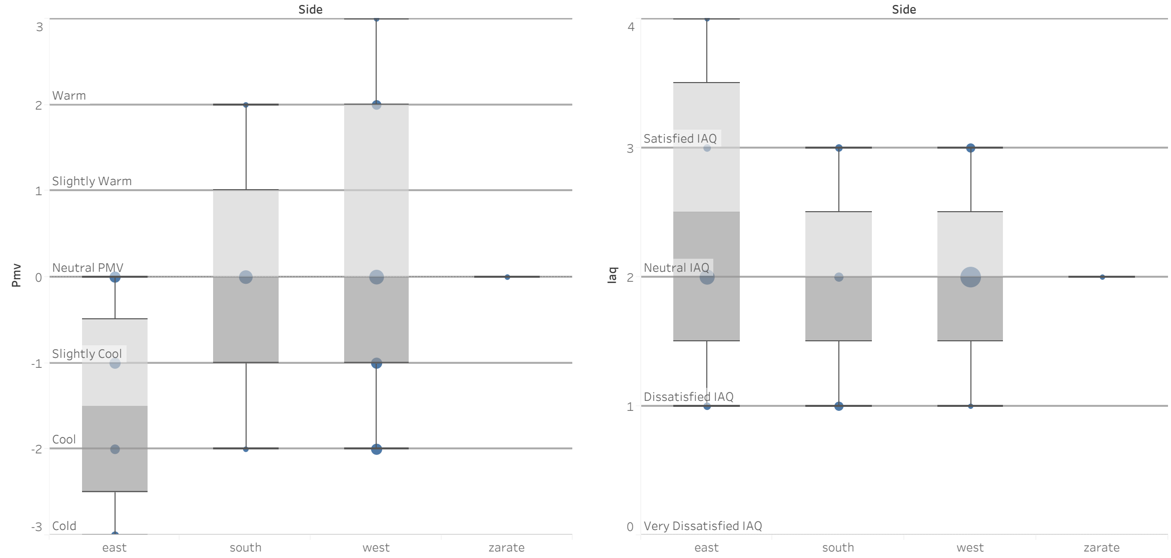
9.1. Thermal comfort

Based on the measurements the computed PMV lies well within the comfortable zone ($[-0.5, 0.5]$) for the vast majority of the time. The reported thermal comfort based on the surveys is found to also average around this comfortable zone (see Figure 9.1a). However, the deviation is quite larger than anticipated. Whilst on average the temperature is perceived comfortable, there are still occupants that complain about cold or heat. No explanation for this *local* discomfort was found. The discomfort is reported on all sides of the building, even though the temperature has been fairly constant throughout the entire measurement period. Both heat and cold complaints occur on the same side of the building under the same circumstances. A common thought is that the airflow from the ventilation system is responsible. However, air velocities measured at the ventilation exhausts did not substantiate this. Note that temperature is a combination of not only air temperature and air velocity, but also absorbed radiant heat on the body. Throughout the measurement period there had been relatively neutral, cloudy weather. Even though cloudy, not all blinds were opened on all sides of the building. The absence of direct sunlight and a bit of the radiant heat emitted with that might play a role in the perceived cold. A recommendation is to actively control the blinds and open them as soon as they are not required anymore. The temperature was noticed to be the lowest and also most constant on the east side of the building (see Figure 9.2). This can be explained by the fact that the thermostat is not controlled on that side of the building (in fact, it is fixed with tape). Based on the survey, employees on the east side perceive the most cold, and it is recommended reconsidering whether this thermostat could use readjustment.

Finally, it is obvious from the temperature readings throughout the day that in the early morning the system actively ventilates and cools the building. This concerns the west side of the building in particular, which then consecutively has to be heated up when the employees arrive in a reportedly rather cold building. It is possible that the temperature sensor that controls the thermostat on this side of the building has drifted over time, leading to unwanted cooling on this side of the building. It can be worthwhile to leave the thermostat at a slight offset for a while and see whether unnecessary cooling can be prevented in order to save energy as well as improve thermal comfort. It is however also possible that the west side of the building is subject to more cooling because it is simply better ventilated.

9.2. Indoor air quality

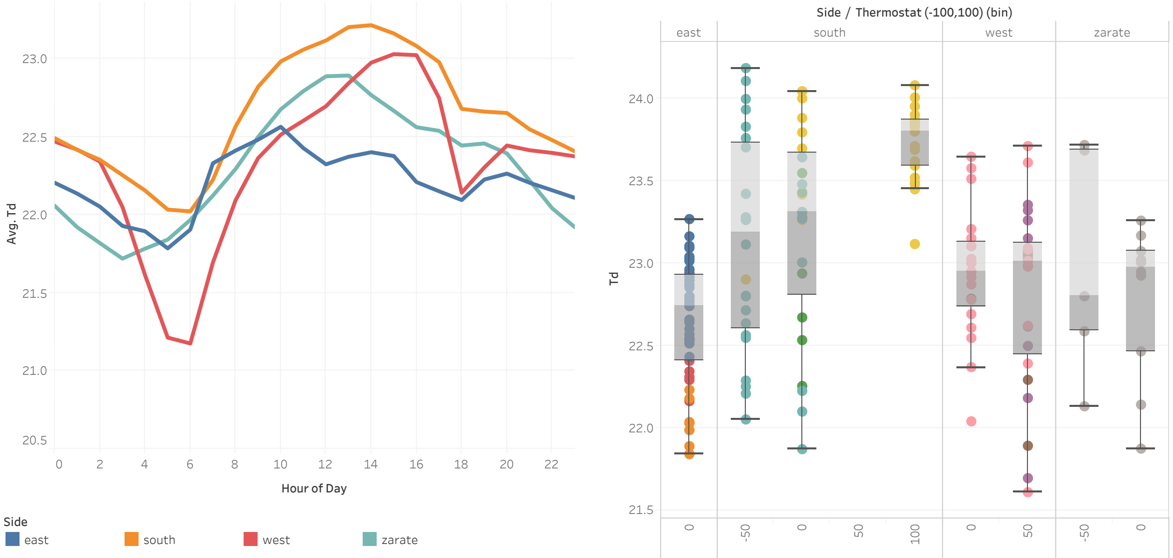
CO_2 concentrations throughout the office are generally acceptable. On average, the concentrations are well below guidelines (1000ppm) and the recommendation of 850ppm (see Figure 9.3), at which a cognitive impact becomes significant (see Appendix A.2). This is also confirmed in the survey: very few employees complained about indoor air quality (see Figure 9.1b). In the meeting rooms, however, both the guideline and recommendation norms are sometimes exceeded. Between 9:00 and 14:00 in particular, concentrations above the recommendation were measured about 10% of the time. Although the air in the meeting room is not polluted on average, it can surely be said that productivity in especially longer and crowded meetings will be affected by this. It is observed that CO_2 concentrations are quite local: CO_2 has a tendency to stay where it is produced and the ventilation is only able to remove that to some extent. Obviously, higher CO_2 concentrations occur with a higher occupancy around the sensor. In particular, when an island of desks is occupied



(a) Reported thermal comfort per side

(b) Reported IAQ per side

Figure 9.1: IAQ Survey results



(a) Temperature throughout business hours

(b) Temperature measurements per setting of the thermostat

Figure 9.2: Temperature measurements

with over 8 employees, elevated CO_2 concentrations were observed. It is recommended to spread out employees evenly across the office space to some extent. Furthermore, it is observed that the CO_2 concentrations on the east side of the building are generally higher (see Figure 9.3). This can be explained by the fact that the east side is simply more crowded (see Figure 9.4b). In the afternoon the concentrations sometimes exceed the recommendation of 850ppm. Concentrations are rarely observed to exceed the guideline of 1000ppm here. During the labeling, it was noticed that elevated CO_2 concentrations in this area mostly come from crowded stand-ups. During these stand-up meetings, air gets polluted very locally, but the CO_2 will stay there for a while. It is recommended that, if multiple of these stand-ups occur during the day, to move these to different sides of the wing rather than concentrating these events on the same spot throughout the day.

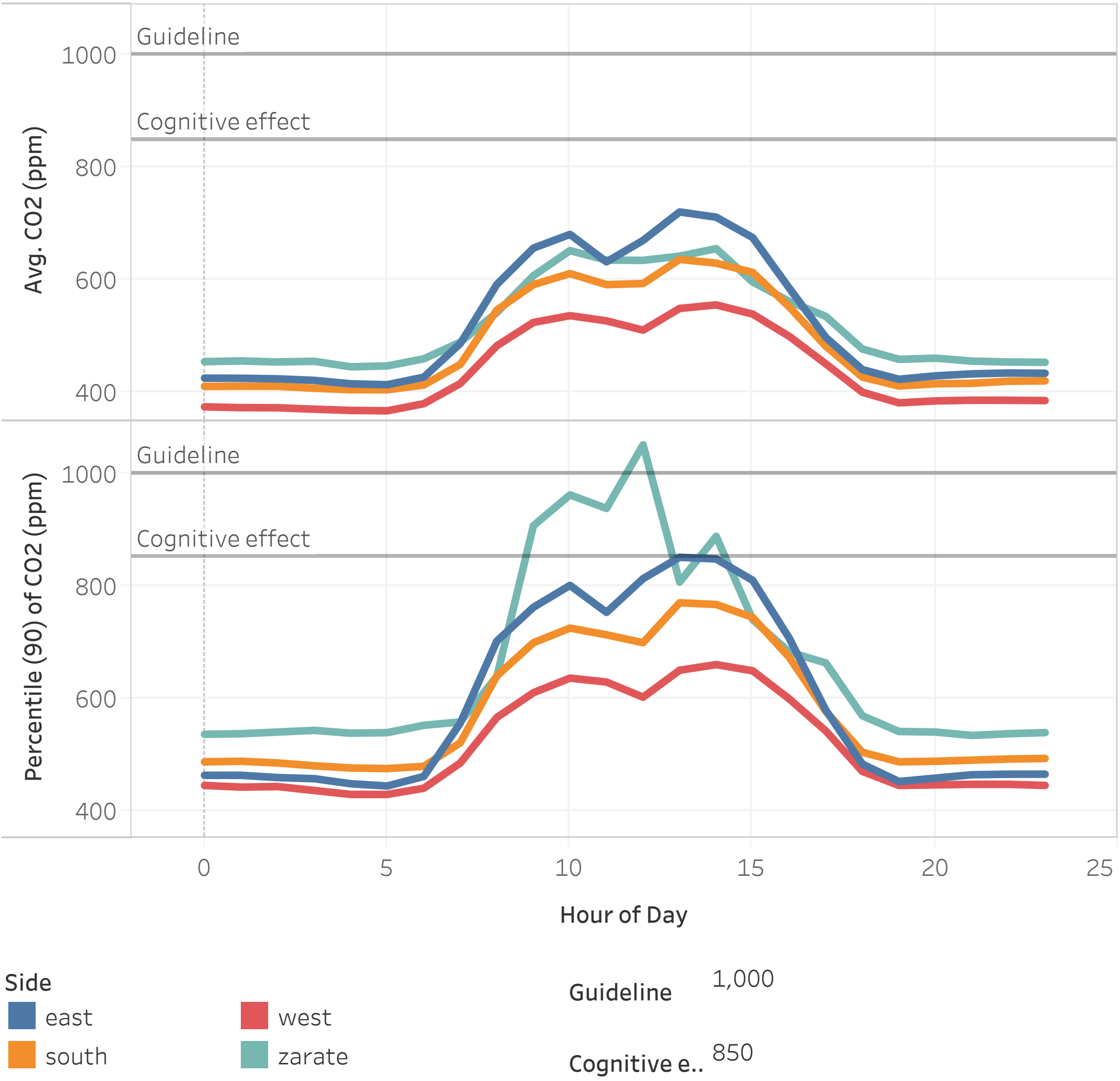
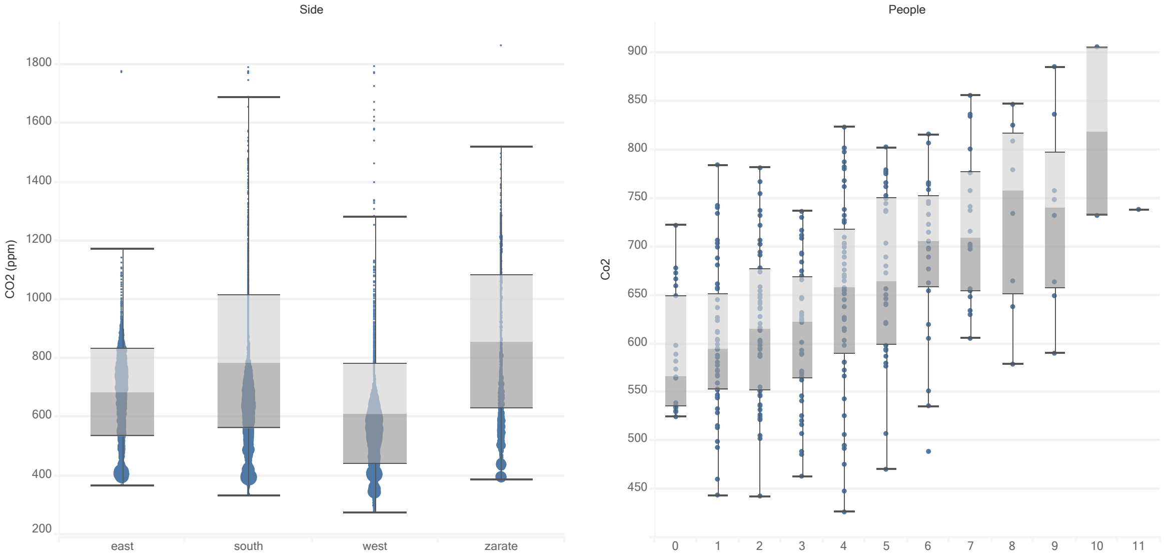


Figure 9.3: CO_2 concentrations per hour of day



(a) CO_2 concentration per side of the building box-plot

(b) Relation between number of nearby employees and CO_2 concentration

Figure 9.4: CO_2 concentration

10

Recommendations

Thermal comfort

It is recommended to install thermostats with a clear temperature indicator. The target temperature cannot be read from the current thermostats, which leads to over-enthusiastic cooling and heating, which then has to be compensated, and consecutively leads to thermal discomfort and waste of energy. It also contributes to a sense of loss of control over the temperature, leading to the belief that not using the thermostat at all is ideal, where this may not be the case due to sensor drift. It is not recommended to fix thermostats in a certain setting for prolonged amounts of time. From the data can be observed that the thermostats are effective after all, and allowing employees to control the thermostat can alleviate the cold complaints among employees. When setting the thermostat, one should not just consider which temperature is *right*, but realize that thermal comfort is based on many factors. If the majority agrees that they feel cold, the thermostat should probably be turned higher. The surveys from the dashboard can be utilized as a means of democratizing the thermostat setting. Furthermore, it is also recommended to actively control the blinds. The radiance from the sun plays an important role in thermal comfort. Its recommended to employees to embed opening the blinds in their workflow: whenever one closes the blinds, one should check every once and a while whether the blinds are still necessary. In particular, whenever one leaves the office is a good moment to open the blinds again and reset the thermostat.

Indoor air quality

Elevated CO_2 concentrations were measured in the Zarate meeting room and on the east side of the building. The standard ventilation in Zarate is not sufficient to keep the room within guideline levels for more crowded and longer meetings. It is therefore recommended to increase the ventilation in Zarate for meetings with more than two people or that will take longer than 30 minutes. For meetings with more people (approximately eight or more) or that take very long (two hours or more) CO_2 concentrations may still reach levels that will cause weariness (above 1000 ppm). It is recommended to open a window when this occurs, as this will drastically decrease the CO_2 concentration in the room. For ad hoc meetings it is recommended to quickly compare the indoor air quality of different rooms in the *Claire* dashboard in order to decide which room to use. For the east side, it was observed that quite a few stand-up meetings may happen throughout the day, which are usually held at the northeast side of the building. This leads to moderately high CO_2 concentrations on that side of the building, which may lead to loss of productivity well after the stand-up meeting. It is recommended that if multiple of these stand-up meetings occur throughout a single day, to organize these meetings at varying sides of the east wing. In general it is also recommended to spread out employees evenly across the office space as much as possible, but with the aforementioned recommendations in place, it is not deemed required to actively spread out workplaces.

Measurement continuation

The findings and recommendations of this report are limited by the data that is available. It is therefore recommended to keep using the sensor network, in particular throughout the summer and winter season, where it is anticipated to measure some more extreme temperatures. Although ventilation exhausts were not conclusively found to be a cause of thermal discomfort, it is recommended to keep using the surveys to collect employee satisfaction data, in particular when there are heat or cold complaints. Perhaps additional data will provide new insight into the role of the ventilation system in the thermal comfort among employees.

Increasing employee engagement

In the busy day-to-day work, the presence of an indoor air quality platform can be easily forgotten, even though employees still want to stay engaged in realizing a better indoor air quality. It is therefore recommended creating a place for the Claire dashboard on a tablet on the wall, so that employees have a quick reminder of the existence of Claire on their way to the coffee corner.

Key user interaction

Due to time constraints, it has not been possible to crystallize out the user notification feature fully. Even though the framework is there, it has still to become clear whether it is productive, for example, to notify employees to increase the ventilation when they enter a meeting room with a certain amount of people. If this method turns out to be too disruptive, an alternative could be to notify the office manager instead. The same applies to notifications regarding overcrowded office areas or changing the thermostat setting based on survey complaints. It is recommended to test each of the implemented notification on a small group of test users, to collect the required insight on how to roll this out effectively.

Inclusion of PM and VOC sensors

Although not considered in this project, Particulate Matter (PM) and Volatile Organic Compounds (VOCs) play a significant role in perceived indoor air quality. High PM concentrations and certain VOCs are also known to cause serious health issues. Therefore, it is recommended to explore the possibility of including PM and VOC sensors in the existing sensor devices, in particular if complaints about the indoor air quality remain that cannot be explained by thermal comfort and CO_2 concentrations alone.

Installation of additional sensor devices

The experiment was limited to 10 devices only, which were deployed throughout the office. As a result, some of the desk islands and the majority of the meeting rooms have no sensors installed yet. Installing sensors in all meeting rooms will allow indoor air quality intervention notifications for these meeting rooms. It also allows employees to chose a meeting room based on its indoor air quality. Installing additional sensor devices at the minority of the desk islands that do currently not have a sensor installed might provide more insight into local discomfort. It will also ensure that every employee always has a sensor within eyesight, which may contribute to a sense of ownership of the project among employees. Lastly, increasing the sensor network density, in general, will improve the underlying network structure and therefore, the reliability of the system. With the current distance between the sensors, the technology is relatively pushed to its limits, which leads to sensors disconnecting from the network incidentally.

11
Conclusion

In this project, the possibilities of utilizing an indoor air quality sensor network to optimize office space utilization for employee health and well-being were explored. As the first step, the properties of indoor air quality, their effects on employee health and well-being, and current industry guidelines were studied. This lead to a selection of properties that were chosen for the measurements for the case study for VTTI. Sensor hardware and firmware, a cloud back-end infrastructure, a reporting solution, and several data processing mechanisms have been developed and lead to the final product *Claire*: a digital assistant and dashboard that monitors indoor air quality. Claire provides VTTI with continuous, real-time and fine-grained insights regarding their indoor air quality and is even able to detect very local and temporal anomalies. This allows VTTI to investigate employee complaints or concerns regarding thermal discomfort or indoor air quality.

The system monitors thermal comfort by measuring dry-bulb temperature, globe temperature and relative air humidity and computing the Predicted Mean Vote. Indoor air quality is determined by measuring CO_2 concentrations. The system can be improved by adding additional air quality sensors to the sensor devices. Predicted Mean Vote and CO_2 concentrations are common methods of estimating thermal comfort and indoor air quality respectively and form the basis of air quality guidelines such as ASHRAE or the Dutch Arbogids. These guidelines were generally met during the measurement period, however, recommended values were sometimes exceeded in the meeting rooms and the east side of the building. These findings aligned with data reported through employee surveys. This lead to a set of concrete recommendations to improve indoor air quality at VTTI.

The system is not only able to classify the current environment, but is also able to predict the progression of the indoor air quality in the short future by looking at the change of variable. One method was implemented that uses the second derivative of the smoothed time series data. This approach was shown to be very effective when the parameters are adjusted correctly for a given room. An alternative method was proposed to predict the progression of the time series data using a Recurrent Neural Network. The latter approach has the potential benefit that it is able to learn the specifics of a certain sensor location.

During the course of this project, all design goals and requirements (as defined in Section 2) have been met. As of the time of writing, *Claire* is still being used use at VTTI.

12 Discussion

Even though very strict planning allowed for the collection of data for as long as four weeks prior to starting the data analysis, the total amount of data was still restrictive. The indoor climate was very constant throughout the measurement period. As a result, the Predicted Mean Vote was practically constant as well and a correlation with the reported thermal comfort could not be found.

The variance in the thermal comfort values reported through the surveys was relatively high. It is possible that these values are amplified by the fact that relatively friendly icons were used to illustrate the *cold* and *hot* thermal comfort levels, which have a meaning that should be interpreted very seriously. Due to the friendliness of the icons, people are unaware of the severity of the meaning of the used comfort levels. From the data cannot be concluded that airflow from ventilation exhausts is responsible for *local* discomfort. It is however possible that our method of measuring temperature and air velocity was not accurate enough to appropriately take the effect of airflow on the thermal comfort into account.

Although the neutral Predicted Mean Vote aligns with the average reported thermal comfort, this finding alone is not sufficient to conclude whether the Predicted Mean Vote model still holds and whether it is an appropriate estimator of thermal comfort. In order to further research this, more tests should be conducted, in particular over a longer time span and in a variety of buildings with different thermal properties. Measuring for a longer time span is also required for observing the influence of the weather. During the measurement period there was relatively moderate weather, which means that the HVAC system did not have to work at full capacity in order to heat or cool the building to the desired temperature and that extreme temperature values within the office have likely not been reached during this period. Any potential HVAC capacity limitations could therefore not be derived from the data. For the same reason, it was also not fruitful to isolate the influence of the local weather on the indoor climate.

The aforementioned have constrained the project in the use of statistical methods such as clustering and neural networks. Nonetheless, interesting trends have been found that can be further researched. Furthermore, the software developed allows for these measurements to be continued. The recommendations provided in this report can be followed through the means of several indoor air quality experiments, which can be benchmarked through the software, with both the hard data from the sensors as well as the soft data from the employee survey reports.

A

Indoor Air Quality

The air quality within buildings and structures is referred to as Indoor Air Quality (IAQ). IAQ can be affected by the concentration of gasses, such as volatile organic compounds (VOCs), carbon dioxide (CO_2), as well as particulates (PM) in the air. Perceived IAQ (or PAQ) is an umbrella of reported descriptors like temperature, presence of odor/smell, and experience of stuffy, dry or wet (humid) air [34]. The perception of temperature, often referred to as thermal comfort, is subject to relative air humidity (RH), air velocity and temperature. PAQ complaints are common and IAQ concerns are ever so often raised, and as a result the subject is of wide interest among scholars.

Wyon [35] has shown that (a) poor indoor air quality can reduce the performance of office work by 6-9%, (b) this measured performance reduction is approximately linearly related with the percentage dissatisfied with indoor air quality, and (c) that these effects were accompanied by negative effects on general symptoms such as headache and concentration.

Poor IAQ has been linked to Sick Building Syndrome (SBS) [36, 37], which is a medical condition identified by a set of symptoms generally observed among office workers, including "(a) eye, nose and throat irritation; (b) a sensation of dry mucous membranes and skin; (c) erythema (skin redness); (d) mental fatigue; (e) headache; (f) a high frequency of airway infections and cough; (g) hoarseness; (h) wheezing, itching and non-specific hypersensitivity; and (i) nausea and dizziness" (WHO 1983, cited from [38]).

In the sections below the components of Indoor Air Quality and its perception are described in more detail.

A.1. Temperature and Humidity

Thermal comfort is the "condition of mind that expresses satisfaction with the thermal environment and is assessed by subjective evaluation" [39]. The body exchanges heat with immediate surroundings through several means of heat exchange: sensible heat flow from the skin (through conduction, convection and radiation), latent heat flow from evaporation of sweat and moisture diffused through the skin, and heat flow during respiration [40]. Task performance is highest when thermal comfort occurs, i.e. when the body temperature is within a certain range, the skin moisture is low, and the physiological effort of regulation is minimal. Research has shown that performance decreases as temperature deviates above or below a thermal comfort temperature range. Specifically, at a temperature 8K higher than optimal, average office task performance decreased with 9% [40]. Conversely, within the thermal comfort temperature range there is a range in which performance does not increase any further.

Thermal comfort is primarily affected by temperature, air speed, humidity, a person's metabolism and clothing insulation [40].

Thermal comfort is frequently modeled using the Predicted Mean Vote (PMV) model. The PMV is a Likert scale scoring from -3 (cold) to +3 (hot), based on heat-balance equations and empirical studies about skin temperature. Values between -1 and 1 are considered satisfactory. The PMV value can be computed given a set of measurements about the temperature, humidity, air velocity, clothing insulation, and type of work. The method to compute the PMV value is described in Appendix B.

Using the value for PMV, the Predicted Percentage Dissatisfied (PPD) can be derived. This is the predicted percentage of people with a score lower than -1 or higher than +1. Fanger and Berg-Munch [41] found the following distribution for PPD:

$$PPD = 100 - 95^{-(0.03353PMV^4 + 0.2179PMV^2)} \tag{A.1}$$

The guideline for PMV is any value between -0.5 and 0.5, which corresponds to a PPD of 10%.

Note that there is no optimal temperature and that some temperature complaints will remain even at an optimal PMV value. There are several explanations for this. One of these explanations is radiant asymmetry, which is the difference in the radiant temperature of the environment on opposite sides. People are sensitive to this asymmetry, in particular when an overhead surface is significantly warmer or a vertical surface is significantly colder. If the gradient is sufficiently large, local discomfort can occur at some body parts. Furthermore, thermal comfort is also subject to many secondary factors such as: day-to-day variations, age, sex, weight, metabolism, adaptation or seasonality [40].

Another complaint that office workers often voice is that an office has 'dry air'. This often shows in symptoms of dry eyes or dry throats. However, humans also often confuse a high concentration of dust particles in the air for low humidity. Raising humidity to higher levels positively alleviates some of these symptoms, as well as increases the perceived IAQ (PAQ) [34]. On the other hand, high indoor humidity can also result in discomfort, while also stimulating the growth of fungus in spaces. In regions with especially hot and humid climates it is shown that in practice, without a desiccant-cooling system, low ventilation rates are used to combat high humidity, therefore causing a buildup of *stale air* and high pollutant levels indoors, leading to a permanent sick building [42].

For the purposes of this research, the PMV-PPD method is used to approximate thermal comfort.

A.2. Carbon Dioxide

High levels of CO_2 concentration are commonly associated with Sick Building Syndrome, which is in turn associated with symptoms such as fatigue, nausea, loss of concentration and tiredness [38, 43, 44].

Because occupants are the dominant source of indoor CO_2, the difference between outdoor and indoor CO_2 concentration (δCO_2) is considered an approximate for indoor concentrations of bioeffluents [38]. Erdmann, C. Steiner, and Apte [45] have shown elevated lower respiratory SBS symptoms with increasing δCO_2, and suggests that an increase in ventilation rate would reduce SBS prevalence of SBS symptoms. Zhang, Wargocki, and Lian [46] found that exposure to human bioeffluents in conjunction with high concentrations of metabolically generated CO_2 cause physiological effects expected to reduce cognitive performance.

Myhrvold, Olsen, and Lauridsen [47] identified a 23% improvement in schoolwork due to a reduction in CO_2 concentrations in classrooms (from above 1500 ppm to below 999 ppm) by increasing ventilation in a field study. Allen et al. [48] have shown that cognitive function scores were 15% lower for the moderate CO_2 day (~ 945 ppm) and 50% lower on the day with CO_2 concentrations of ~ 1,400 ppm than on the two regular days (~ 500 ppm) in a Green+ certified building with artificially controlled CO_2 concentrations independent of ventilation. The same research found that on average a 400 ppm increase in CO_2 was associated with a 21% decrease in a participant's cognitive scores across all domains.

CO_2 concentration is included in some industrial guidelines and standards, such as the ANSI/ASHRAE 62-1989, ASTM D 6245-98 and Dutch Arbocatalogus, which recommend a maximum acceptable indoor CO_2 concentration of 1000 ppm. This guideline is however not mandatory. In general, the required ventilation rate is considered to be sufficient for the CO_2 concentration to stay within reasonable levels. However, this relation between CO_2 concentration and ventilation rate is complicated by the transient nature of multizone systems as well as temporal and spatial variations in occupancy (i.e. meeting rooms).

A.3. Volatile Organic Compounds

Volatile Organic Compounds (VOCs) are emitted from (i) biological sources such as humans, molds, bacteria, indoor plants; (ii) outdoor sources such as: traffic, industry, agriculture; (iii) furniture; (iv) consumer products such as air fresheners; (v) building materials; (vi) office equipment such as printers; and (vii) household solvents [49]. VOCs have been associated with (a) activation of the immune system (eye, nose and throat irritations); (b) exacerbation of asthma; (c) carcinogenic effects; (d) headaches; (e) loss of coordination; (f) nausea; and (g) damage to organs such as the liver, kidneys or central nervous system [49], however, the effects with respect to regular indoor VOC concentrations is still unclear.

VOCs can be regulated through source control. In the past decades, there has been a "continued effort to develop and use low emitting building materials and consumer products during the last decades, e.g. by implementation of national labeling schemes for emission testing (Wolkoff, 2003), a change to lower room concentrations of VOCs by lower material emissions (Tuomainen et al., 2003), and use of less volatile organic compounds (Weschler, 2009)" (cited from Wolkoff [34]). Indoor plants are also known to reduce VOCs [50].

VOCs are commonly associated with Sick Building Syndrome (SBS), although the role of VOCs in SBS complaints is far from being fully understood. Normal indoor air concentrations of VOCs cannot explain indoor air complaints [51]. It should be noted that the diffusion of VOCs is affected by temperature and humidity [52]. Consecutively, air conditioners and filters also release VOCs and are themselves associated with SBS complaints [49]. This might offset results from studies.

Many different types of VOCs are present in typical indoor environments, which makes it practically impossible to identify and quantify every compound individually. As a result, a uniform procedure to measure total volatile organic compound (TVOC) was proposed. Whilst there are no TVOC regulations in most countries, the WHO includes VOCs such as formaldehyde and benzene in their Indoor Air Quality (IAQ) guidelines, and, for example, Germany has a TVOC guideline of 300 μg/m^3 [49].

A.4. Particulate Matter

Particulate Matter (PM) is one of the most dangerous air pollutants, and can cause serious health problems [53]. The effect of particulate matter on health is dependent on the size of the particles [54]. Therefore, a distinction is made by defining two categories of particulate matter, namely $PM_{2.5}$ and PM_{10}. $PM_{2.5}$ and PM_{10} are defined as particles with a diameter up to 2.5 μm and 10 μm respectively. $PM_{2.5}$ is most harmful, as these particles can penetrate into the lungs and bloodstream, causing asthma attacks or cardiovascular problems [55].

A.5. Airflow

An important factor in improving the thermal comfort and the gas composition of a building is the (natural) ventilation and infiltration. Several processes might take place that affect the movement of air in a building.

Stack effect takes place when there is a difference in air density between the indoors and outdoors and a site with air leakage [56]. A site of air leakage could, for example, be an open window, a porous exterior material or a ventilation pipe. A difference in air density might naturally occur when air is blowing past a building or when either inside or outside has a higher air temperature. The difference in air density causes either exfiltration or infiltration of gases and contaminants, which could be either beneficial or detrimental to IAQ.

A proper analysis of the thermal comfort of a building requires measuring the airflow. There is various software available on the market to simulate airflow dynamics and/or do energy analysis on a building. These might be helpful to calculate the effect that measures such as opening windows and doors may have on the temperature, humidity and IAQ. Other measures, such as the effect of installing window blinds on temperature, can also be simulated.

A drawback of simulating the airflow is that a model needs to be built of the building. This model needs some specifications of the building, such as the materials used in the building exterior or the HVAC, that might not be obtainable or possible to model.

B

Predicted Mean Vote

Thermal comfort is frequently modeled using the PMV model. The PMV is a Likert scale scoring from -3 (cold) to +3 (hot). Values between -1 and 1 are considered satisfactory. Its formula is based on heat-balance equations and empirical studies about skin temperature. In this project the PMV scale is used to measure thermal comfort through surveys and as a means of predicting the thermal comfort based on the sensor measurements. The PMV cannot be measured directly and is dependent on many variables, some of which cannot be measured. For other parameters it suffices to derive the value from another known equation, estimator or constant value.

The PMV value is expressed using the following equations [41]:

$$PMV = (0.303e^{-0.036M} + 0.028)L \tag{B.1}$$

where M is the metabolic rate and L is the thermal load, which is defined as follows:

$$L = (M - W) - H - E_c - C_{res} - E_{res} \tag{B.2}$$

Where E_c is the evaporation through diffusion and sweat, E_{res} is the heat loss through vapour from respiration, C_{res} is the heat loss through temperature of respiration, and H is the radiation minus convection, which can be derived from:

$$H = \varepsilon\sigma\frac{A_r}{A_{Du}} \cdot f_{cl} \left\lfloor (t_{cl} + 273)^4 + (t_r + 273)^4 \right\rfloor - f_{cl} \cdot h_c \cdot (t_{cl} - t_a) \tag{B.3}$$

$$H = K_{cl} = \frac{t_{sk} - t_{cl}}{I_{cl}} \tag{B.4}$$

Combining the above two equations, gives the following equation for the clothing temperature t_{cl}, which can be solved iteratively:

$$t_{cl} = t_{sk} - I_{cl} k_l f_c l \left\lfloor (t_{cl} + 273)^4 + (t_r + 273)^4 \right\rfloor - I_{cl} f_{cl} h_c (t_{cl} - t_a) \tag{B.5}$$

With:

$$k_1 = \varepsilon\sigma\frac{A_r}{A_{Du}} = 3.96 \cdot 10^{-8} \tag{B.6}$$

$$h_c = \begin{Bmatrix} 2.38(t_{cl} - t_a)^{0.25}, & 2.38(t_{cl} - t_a)^{0.25} > 12.1 \cdot \sqrt{v_a} \\ 12.1\sqrt{v_a}, & 2.38(t_{cl} - t_a)^{0.25} \leq 12.1 \cdot \sqrt{v_a} \end{Bmatrix} \tag{B.7}$$

$$f_{cl} = \begin{Bmatrix} 1.00 + 1.29 \cdot I_{cl}, & I_{cl} < 0.078\,\mathrm{m^2{}^\circ C/W} \\ 1.05 + 0.645 \cdot I_{cl}, & I_{cl} \geq 0.078\,\mathrm{m^2{}^\circ C/W} \end{Bmatrix} \tag{B.8}$$

H can now be written as a function of t_{cl}:

$$H = 3.96 \cdot 10^{-8} \cdot f_{cl} \cdot \lfloor (t_{cl} + 273)^4 + (t_r + 273)^4 \rfloor - f_{cl} \cdot h_c \cdot (t_{cl} - t_a) \tag{B.9}$$

Furthermore:

$$E_c = 3.05 \cdot 10^{-3} \cdot [5733 - 6.99 \cdot (M - W) - P_a] + 0.42 \cdot [(M - W) - 58.15] \tag{B.10}$$

$$E_{res} = 1.72 \cdot 10^{-5} \cdot M \cdot (5867 - p_a) \tag{B.11}$$

$$C_{res} = 1.4 \cdot 10^{3} \cdot M \cdot (34 - t_a) \tag{B.12}$$

This yields the following final equation for L, that can be solved for metabolic rate M (well known for common activities), external work W (usually 0), clothing factor I_{cl} (well known for regular seasonal clothing), air temperature t_a, mean radiant temperature t_r, air velocity v_a and relative humidity rh:

$$\begin{aligned} L = (M - W) - 3.05 \cdot 10^{-3} \cdot [5733 - 6.99 \cdot (M - W) - P_a] \\ -0.42 \cdot [(M - W) - 58.15] - 1.7 \cdot 10^{-5} \cdot M \cdot (5867 - P_a) - 1.4 \cdot 10^{3} \cdot M \cdot (34 - t_a) \\ -3.96 \cdot 10^{-}8 \cdot f_{cl} \cdot \lfloor (t_{cl} + 273)^4 - (t_r + 273)^4 \rfloor - f_{cl} \cdot h_c \cdot (t_{cl} - t_a) \end{aligned} \tag{B.13}$$

The water vapour pressure p_a can be approximated as follows:

$$p_a = rh \cdot 10 \cdot \exp(16.6536 - \frac{4030.183}{t_a + 235}) \tag{B.14}$$

The mean radiant temperature t_r is the "uniform temperature of an imaginary enclosure in which radiant heat transfer from the human body equals the radiant heat transfer in the actual nonuniform enclosure [40]". Measurements of the globe temperature t_g, air temperature t_a, and air velocity V_a can be combined to estimate the mean radiant temperature using equation B.15:

$$t_r = [(t_g + 273)^4 + \frac{1.10 * 10^8 V_a^{0.6}}{\varepsilon D^{0.4}} \cdot (t_g - t_a)]^{1/4} - 273 \tag{B.15}$$

where V_a is the air velocity, t_g is the globe temperature, t_a is the air temperature, D is the globe diameter and ϵ is the emissivity (0.95 for a black globe) [40, 41].

C

Sensors

In this section a variety of indoor air quality sensors will be discussed, namely CO_2, VOC, Particulate Matter, temperature, and humidity sensors. For each of these sensors the different types of sensor technologies available will be outlined, together with their advantages and disadvantages. Furthermore, a list of commercially available sensors will be provided that are considered as options for the system. Sensors in these lists were chosen based on presence in research papers comparing the performance of various sensors, and usage in consumer IAQ monitors, such as the Foobot and Air Mentor.

C.1. CO_2 Sensors

There are two types of sensors most commonly used for CO_2 detection, namely chemical sensors [57] and Non-Dispersive Infrared (NDIR) sensors [58].

Chemical sensors are sensors with polymer-based coatings. CO_2 reacts with the polymer coating, which then changes the mass and dielectric properties of the coating [59]. Advantages of polymer-based chemical sensors are as follows: (a) high sensitivity [59], (b) short response time [59, 60], (c) low cost [59, 60], (d) simple structure [59, 60], and (e) low power consumption [59]. However, these sensors suffer from (a) long-term instability [59], (b) poor selectivity [59], and (c) short lifetime of the coating [60].

NDIR sensors are based on molecular absorption spectrometry [59]. The sensors measure CO_2 concentration by detecting which wavelengths are absorbed by the gases present in the sensor's gas chamber, and whether these correspond to the wavelengths absorbed by CO_2 [59]. NDIR sensors have a large number of advantages, namely: (a) long-term stability [61], (b) high accuracy [61], (c) low power consumption [61], (d) high selectivity [61], (e) high sensitivity [61], (f) long lifetime [59], and (g) insensitivity to change in environment [59]. The drawback of NDIRs however, is their high cost [59].

Table C.1 contains the CO_2 sensors that were taken into consideration for the system. The selection of sensors for that was used in tests for this project are depicted in Figure C.1. It comprises of NDIR sensors exclusively, as these seem to be the most commercially available. CO_2 sensors often require some calibration in order to produce accurate measurements. The MH-Z14A, MH-Z19B, LP8, and K30 do not ship calibrated, but they all have Automatic Baseline Correction, which means that over a certain calibration period the lowest detected CO_2 concentration will be considered to be 400 ppm (as these sensors are not able to detect anything lower). However, issues arise when the CO_2 concentration in the sensor environment does not drop to 400 ppm or lower because then the sensor will take a much higher concentration as a reference point for 400 ppm, which will produce very inaccurate results. Furthermore, the MH-Z14A and MH-Z19B require a zero-point calibration for at least 20 minutes in an environment with a CO_2 concentration of 400ppm before first use.

The SCD30, K33 ELG and S8 LP all come readily calibrated and as such need no additional maintenance, which gives them an advantage over the aforementioned sensors that are not precalibrated. Out of these three sensors, the SCD30 and K33 ELG are the most accurate (±30 ppm), and the SCD30 is the most affordable. Therefore, the decision is made to use the Sensirion SCD30.

The ELT T-110 and Alphasense IRC-A1 are not readily available and therefore not an option due to time constraints.

Table C.1: CO_2 sensors

Manufacturer	Model	Measurement range	Accuracy	Measurement method	Power draw	Cost
Alphasense	IRC-A1 [62]	0-5000 ppm	±50 ppm	NDIR	20-60 mA	?
SenseAir	K30 [63]	0-5000 ppm	±30 ppm	NDIR	40 mA	98 €
SenseAir	K33 ELG [64]	0-5000 ppm	±30 ppm	NDIR	60 mA	144 €
SenseAir	S8 LP [65]	400-2000 ppm	±40 ppm	NDIR	18 mA	98 €
SenseAir	LP8 [66]	0-2000 ppm	±50 ppm	NDIR	31-225 μA	98 €
ELT	T-110 [67]	400-10000 ppm	±50 ppm	NDIR	20 mA	?
Winsen	MH-Z14A [68]	0-5000 ppm	±50 ppm	NDIR	60-150 mA	36 €
Winsen	MH-Z19B [69]	0-5000 ppm	±50 ppm	NDIR	60-150 mA	33 €
Sensirion	SCD30 [70]	0-40000 ppm	±30 ppm	NDIR	19-75 mA	42 €
Telaire	6703 [71]	400-5000 ppm	±75 ppm	NDIR	25 mA	116 €

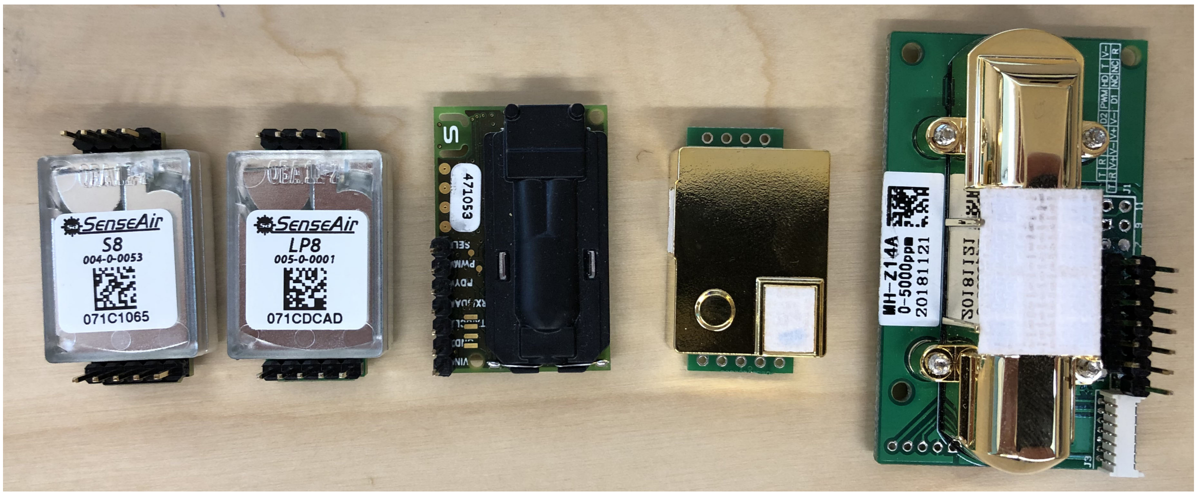


Figure C.1: From left to right: SenseAir S8, SenseAir LP8, Sensirion SCD30, Winsen MH-Z19B and the Winsen MH-Z14A CO_2 sensors

C.2. VOC Sensors

There is a variety of VOC sensors, including spectrometers, gas chromatographs, photo-ionization detectors, and metal oxide semiconductors. Spectrometers and gas chromatographs are very accurate and able to distinguish between different gases [72]. However, these methods are not only costly but are also not portable, have a high power consumption and low throughput [73], which makes them not suitable for use in embedded devices. Photo-ionization detectors (PIDs) and metal oxide (MOx) sensors, on the other hand, are more suitable for such use cases. PIDs detect VOCs by ionizing gas molecules and measuring the current produced by the created ions. By using UV light for the ionization, only molecules with low ionization energy (thus molecules of VOCs) are ionized, and not other common components in the air, such as oxygen or nitrogen [72]. According to Spinelle et al. [72], the advantages of PIDs are as follows: (a) portability, (b) sensitivity, and (c) low power consumption. Spinelle et al. [72] also mention a couple of disadvantages, namely: (a) despite their sensitivity, PIDs are not able to detect very low levels of VOCs, (b) poor gas selectivity, as everything below the ionization energy threshold is ionized, and (c) high cost.

MOx sensors are sensors with metal oxide semiconductors on their surface. Molecules of VOCs react with the metal oxide, changing the conductivity of the surface, which is measured by the sensor [72, 59]. MOx sensors have the following advantages: (a) high sensitivity to VOCs [73], (b) portability [73], (c) short response time [73], (d) low cost [73], (e) low power consumption [73], and (f) ability to sense VOCs that are not detectable by PIDs (e.g. some chlorofluorocarbons (CFCs)) [72]. However, MOx sensors also have a couple of drawbacks: (a) poor gas selectivity (partially due to response to some inorganic gases) [73, 72], and (b) inability to detect low levels of VOCs (≤100 ppb) [72].

Both PIDs and MOx sensors are not able to distinguish between individual gases [72].

Table C.2 contains the VOC sensors taken into consideration for the system in this project. The AMS iAQ-Core-C is used in the Foobot [6] and Air Mentor [74] monitors. Moreno-Rangel et al. [6] researched the precision and accuracy of the sensors present in the Foobot, and found the iAQ-Core-C to provide reliable measurements when compared to scientifically validated instruments. The iAQ-Core-P is the pulsed equivalent of the iAQ-Core-C, with the only difference being power draw, and as such can be expected to perform similarly. The Figaro TGS 2602 is only able to detect high concentrations of VOCs, making it not very suitable for measuring IAQ, as the indoor levels are usually much lower.

Although the iAQ-Core-C performed very well according to Moreno-Rangel et al. [6], it turned out to be calibrated using a proprietary calibration algorithm from Foobot, meaning that out of the box the iAQ-Core-C will not perform as well. MOx sensors have a poor selectivity for which gases are detected [72]. As a result, these sensors only provide accurate results when they are specifically calibrated for a composition of gases of the environment they are placed in. Furthermore, VOC sensors are often calibrated against a specific composition of gases, which will skew results in environments with different gas compositions. A solution for this would be to use a combination of VOC sensors that are all sensitive to different VOCs and to combine the results, though this is a very involved process [75]. Another issue common in VOC sensors is that their lower limit of detection is too low to be able to detect the VOC levels for a typical indoor environment. With these disadvantages of VOC sensors in mind, the decision was made not to measure VOCs for now.

Table C.2: VOC sensors

Manufacturer	Model	Measurement range	Measurement method	Power draw	Cost
Figaro	TGS 2602 [76]	1-30 ppm	MOx	51-61 mA	22 €
AMS	iAQ-Core-C [77]	125-600 ppb	MOx	20 mA	17 €
AMS	iAQ-Core-P [77]	125-600 ppb	MOx	2.7 mA	17 €
Sensirion	SGP30 [78]	0-60 000 ppb	MOx	48 mA	23 €
Sensirion	SGPC3 [79]	0-60 000 ppb	MOx	1 mA	11 €

C.3. PM Sensors

Particulate Matter (PM) is often measured through photometers and optical particle counters (OPCs). Photometers and OPCs are two types of particulate matter sensors based on light scattering [80]. In light scattering sensors the LED emits light which is scattered by the particles present in the detection volume. The scattered light is focused by a lens and subsequently measured using a photodetector. The intensity of light measured by the photodetector is correlated with the concentration of particles present [81].

Photometers are cheaper and more compact than OPCs because they measure all particles present in the detection volume as a group [81, 53]. However, this also limits their measurement performance, as only the mass concentration can be determined using this method [80, 81]. Another consequence of measuring all particles as a group is that the sensors are not size selective, meaning that they are not able to determine the size of the particles [82]. Additionally, the performance of photometers is affected by particle composition and size. The refractive index and light absorption of materials influences the light scattering and light intensity respectively, meaning that particles with different properties can produce varying inaccurate results [81].

OPCs have a detection volume smaller than the ones in photometers, such that only a single particle is lit each time. This enables OPCs to detect individual particles and their size, as the light intensity measured by the photodetector now indicates the size of the particle [53]. As a result, OPCs can provide number concentrations as well as mass concentrations, where concentrations can also be determined for certain particle sizes. Furthermore, OPCs can measure particle size distributions as well [54, 80].

Table C.3 contains the Particulate Matter sensors that were taken into consideration for the system. The selection of sensors used in tests for this project are shown in Figure C.2. All these sensors are based on light scattering. The SHARP GP2Y1010AU0F and Shinyei PPD42NS are photometers while the others are OPCs.

Moreno-Rangel et al. [6] evaluated the precision of the Sharp GP2Y1010AU0F present in the Foobot. They found the sensors overestimated PM mass concentrations, yet had good agreement with the scientifically validated instruments. However, the sensors were calibrated using a proprietary Foobot calibration algorithm, and therefore the sensors will not produce the same results out of the box. Sousan et al. [80] measured the

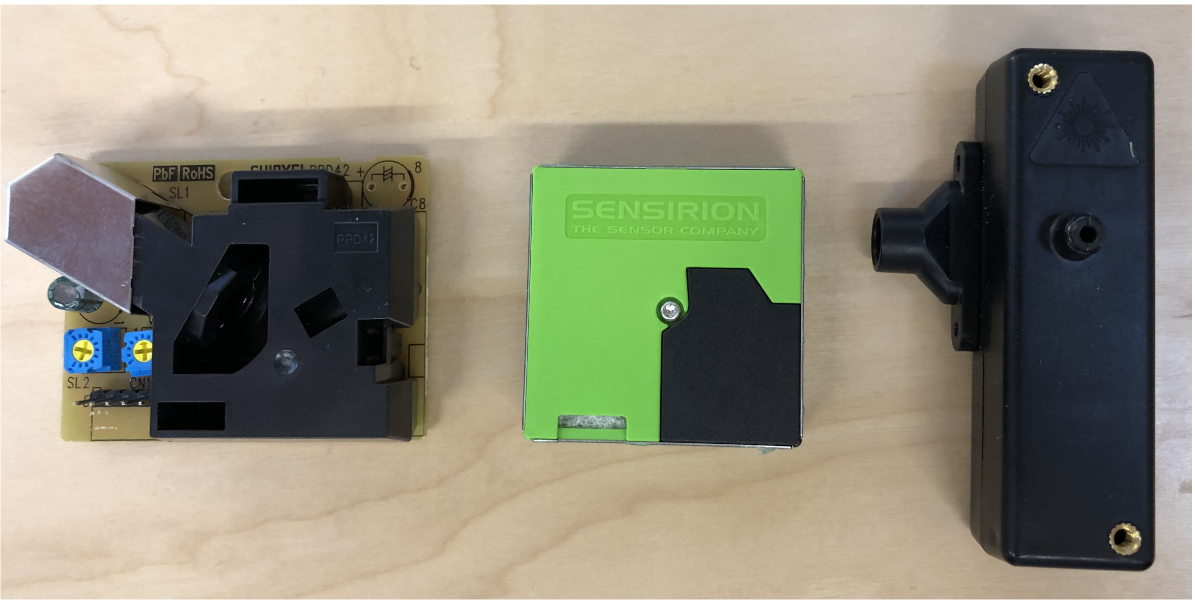


Figure C.2: From left to right: Shinyei PPD42NS, Sensirion SPS30 and AlphaSense OPC-R1 PM sensors

precision of the Sharp GP2Y1010AU0F, and found that the output between multiple sensors varied significantly, caused by the fact that the sensors are not calibrated in the factory to ensure they produce the same results. Nevertheless, Sousan et al. [80] stated the SHARP GP2Y1010AU0F would produce reliable mass concentrations when calibrated specifically for the site it will be used in to account for the particle composition present in the environment. Later, Sousan et al. [83] also evaluated the GP2Y1010AU0F present in the Foobot, however in contrast to Moreno-Rangel et al. [6], came to a similar conclusion as before, namely that it needed additional site-specific calibration in order to produce reliable results.

The Shinyei PPD42NS is used in the Uhoo [84] and Air Mentor [74] air quality monitors. Prabakar, Mohan, and Ravisankar [55] found the Shinyei PPD42NS to have a good correlation with the Dylos DC1100 pro, a high-end laser particle counter that is able to detect individual particles and their size. Qian, Pang, and O'Neill [85] had the same results when comparing the Shinyei PPD42NS to the Dylos DC1100 pro, also taking the Sharp GP2Y1010AU0F into consideration and concluding that the PPD42NS significantly outperformed the Sharp GP2Y1010AU0F.

Both the SHARP GP2Y1010AU0F and Shinyei PPD42NS, being photometers, do not separate particles on their size, meaning that they only detect the presence of particles and not their size. This lack of size selectivity can be a problem when calculating the mass concentration for different size classes of particulate matter, such as $PM_{2.5}$ and PM_{10} concentrations. The sensors from Alphasense do make this distinction. The Alphasense sensors however, are both expensive and not easily available as they need to be ordered from the manufacturer directly, and as such have long delivery times.

Table C.3: PM sensors

Manufacturer	Model	Particle size	Detectable concentration	Power draw	Cost
Sharp	GP2Y1010AU0F [86, 87]	0.1-100 μm	0-500 μg/m^3	20 mA	10 €
Shinyei	PPD42NS [88]	≥ 1 μm	0-28 000 pcs/L	90 mA	10 €
Alphasense	OPC-N3 [89]	0.35-40 μm	0.01 μg/m^3 - 1500 mg/m^3	45-180 mA	327 €
Alphasense	OPC-R1 [90]	0.35-12.4 μm	0.01 μg/m^3 - 1500 mg/m^3	5-95 mA	123 €
Honeywell	HPM [91]	0-10 μm	0-1000 μg/m^3	80 mA	35 €
Sensirion	SPS30 [92]	0.3-10 μm	1-1000 μg/m^3	60 mA	42 €

C.4. Temperature and Humidity Sensors

Table C.4 contains the temperature and humidity sensors taken into consideration for this project's system. Most sensors are able to measure both temperature and humidity, except for the Bosch BMP180 and Maxim Integrated DS18B20 that are only able to measure temperature.

The Foobot air quality monitor contains the Sensirion SHT20 for measuring temperature and humidity. Moreno-Rangel et al. [6] found the sensor to have a very significant agreement with the scientifically validated instruments used for both temperature and humidity, and as such deemed it to be reliable enough to be recommended. The SHT31 and SHT35 are the successors of the SHT20 [93]. As the SHT35 is the high-end model of the SHT3x series with the best accuracy for both temperature and humidity, the decision was made to use it for the system. Especially for temperature the accuracy is important because of the limited measurement range (namely room temperature) and small variations in temperature already have a significant effect on thermal comfort, so it is important to measure that as accurately as possible.

Table C.4: Temperature and humidity sensors

Manufacturer	Model	Measurement range	Accuracy	Measurement method	Power draw	Cost
Sensirion	SHT31-DIS-B [93]	0-90 °C, 0-100 %RH	±0.2 °C, ±2%RH	Thermistor, capacitive humidity [94]	1.5 mA	17 €
Sensirion	SHT35-DIS-B [93]	20-60 °C, 0-80 %RH	±0.1 °C, ±1.5%RH	Thermistor, capacitive humidity [94]	1.5 mA	12 €
Bosch	BME280 [95]	0-65 °C, 20-80 %RH	±1.0 °C, ±3%RH	Unknown	3.6 μA	24 €
Bosch	BMP180 [96]	0-65 °C	±1.0 °C	Unknown	0.65 mA	5 €
Maxim Integrated	DS18B20 [97]	-55-125 °C	±0.5 °C	Digital thermometer	1.5 mA	10 €
Adafruit	DHT22 [98]	-40-80 °C, 0-100 %RH	±0.5 °C, ±5%RH	Thermistor, capacitive humidity	2.5 mA	10 €

D
Open Data

D.1. KNMI

Because of the hypothesis that thermal discomfort may be caused by the radiance from the sun (or absence thereof) or the difference between the inside and outside temperature, some weather data needs to be collected. Fortunately, KNMI publishes local weather measurements on a daily basis on its website [99]. The closest KNMI station (Rotterdam) is at a 5 km distance from the VTTI HQ. The KNMI data contains the properties listed in Table D.1.

Resource	Abbreviation	Measurement	Frequency	Measurement size
KNMI	WIND.DD	Wind Direction (0-360 degrees)	1 / hour	2 bytes (small int)
KNMI	WIND.FH	Hourly Average Wind speed (m/s)	1 / hour	4 bytes (float)
KNMI	WIND.FF	Average Wind speed of last 10 minutes (m/s)	1 / hour	4 bytes (float)
KNMI	WIND.FX	Highest Gust (m/s)	1 / hour	4 bytes (float)
KNMI	TEMP.T	Outside Temperature in Celsius	1 / hour	4 bytes (float)
KNMI	TEMP.T10N	Minimal temperature in last 6 hours	1 / hour	4 bytes (float)
KNMI	TEMP.TD	Dew Point Temperature	1 / hour	4 bytes (float)
KNMI	SUNR.SQ	Sunlight Duration	1 / hour	4 bytes (float)
KNMI	SUNR.Q	Global Sun Radiation (J/m2)	1 / hour	4 bytes (float)
KNMI	PRCP.DR	Duration of Rainfall	1 / hour	4 bytes (float)
KNMI	PRCP.RH	Hourly Sum of Rainfall	1 / hour	4 bytes (float)
KNMI	VICL.VV	Vision	1 / hour	1 byte (enum)
KNMI	VICL.N	Cloudiness	1 / hour	1 byte (enum)
KNMI	VICL.U	Relative Humidity	1 / hour	4 bytes (float)
KNMI	WEER.M	Mist	1 / hour	1 byte (boolean)
KNMI	WEER.R	Rainfall	1 / hour	1 byte (boolean)
KNMI	WEER.S	Snow	1 / hour	1 byte (boolean)
KNMI	WEER.O	Thunder	1 / hour	1 byte (boolean)
KNMI	WEER.Y	Ice Formation	1 / hour	1 byte (boolean)

Table D.1: KNMI data available properties

E

Info sheet

General Information

Title of the project: Optimizing office space utilization using an Indoor Air Quality sensor network
Name of the client organization: VTTI
Date of the final presentation: July 2nd, 2019 15:00

Description

Sick Building Syndrome is present in 30% of all office buildings and can cause serious health damage over time. This is an era where sustainability and well-being are becoming dominant aspects of life. As a result, it is becoming increasingly important to businesses to invest in their employees' well-being and health. The VTTI group cares for the well-being of their employees, and is looking for a tool to optimize the utilization of their building for perceived thermal comfort and indoor air quality.

This report documents the development of *Claire*, an indoor air quality dashboard that helps to identify local air quality problems. Using *Claire*, employees can be rearranged throughout the space, learn about the characteristics of their office, and for example switch to another meeting room. *Claire* translates measurements into insights. *Claire* learns about the behavior of the office, and gives recommendations once she notices that the indoor air quality can be improved.

Claire is backed by an indoor air quality sensor mesh network, which has been developed as part of this project. The sensors continuously measure temperature, humidity and carbon dioxide concentrations. The sensors connect to a cloud infrastructure through a local internet gateway. In the cloud the data gets processed. All measurements are displayed real-time in the dashboard.

Claire is different from existing products in several ways. First, the sensors developed measure both dry-bulb and black globe temperature, which gives it a temperature reading that describes human thermal comfort more accurately. This is not done in competing products. Furthermore, the sensors fill the gap for small and medium-sized enterprises (SMEs): the sensor network is able to get fine-grained results due to its high sensor density, whilst still being very easy to setup with no adjustments to the building being required. Finally, the developed data analysis methods translate the measurements from the sensor network to concrete suggestions, sent through a push notification, which enables workers to get involved with improving the indoor air quality in their office space.

Members of the project team

Name: Jan-Willem Gmelig Meyling
Interests: Software Engineering, Database Systems, Distributed Computing
Contribution and role: dashboard frontend and backend development, sensor assembly

Name: Sayra Ranjha
Interests: Multimedia, Artificial Intelligence, Computer Vision, Quantum Computing
Contribution and role: embedded firmware programming, data analysis layer, sensor testing and selection

Name: Leon Hoek

Interests: Data Mining, Pattern Recognition, Software Testing
Contribution and role: data analysis layer, software integrations, databases

Contact Information

Client Supervisor: Margreeth Doornbosch - Management Assistant at VTTI
Client Supervisor: Ruud Timmermans - Global Automation Lead at VTTI
Client Supervisor: Margit Blok - Global Director Health, Safety & Environment (HSE) at VTTI
TU Delft Coach: Hayley Hung - Associate Professor Intelligent Systems Department Delft University of Technology
TU Delft Coach: Ekin Gedik - Postdoctoral Researcher Socially Perceptive Computing Group Delft University of Technology

The final report for this project can be found at: `http://repository.tudelft.nl`

F

Project description

Time series data analysis for measuring indoor air quality in buildings
Keywords: indoor air quality, data processing, forecasting, time series data

Context

According to the World Health Organization, as many as 30% of all buildings suffer from a phenomenon known as Sick Building Syndrome (SBS). Bad Indoor Air Quality (IAQ) is known to cause a drop in productivity up to 50%, for instance caused by fatigue and headaches. Clairify is a start-up company in the indoor air quality industry. At Clairify we tackle unhealthy indoor air quality with smart software and IoT that optimizes offices and assists employees in taking action to improve the IAQ, resulting in higher productivity and enhanced cognition. We want buildings to breathe.

Together with VTTI we have put our heads together to measure indoor air quality and comfort within their offices. VTTI is a company that specializes in tank storage, and currently has over 9.2M cubic meters of storage spread over facilities in 14 countries. VTTI is very involved with the well-being of their employees and realizes how important clean air is. Therefore, Clairify has deployed sensor hardware at VTTI and is conducting measurements as well as consulting VTTI on indoor air quality. During these measurements the opportunity became apparent of collecting data at a larger scale, with greater sensor density and being able to read (and act on) these results in real time.

VTTI is looking for a tool that will engage their employees with the air quality within the building. For example, it would be particularly useful to see actual air quality standards of available meeting rooms, so that you can pick a meeting room with fresh air. Or to receive a notification that a particular room is polluted and requires ventilation. Furthermore, employees have indicated thermal discomfort, which is believed to be caused by the zoning of the HVAC system and positioning of the interior with regard to the ventilation vents, so VTTI would like to know whether measuring air quality could provide valuable insights how to optimally use their building.

The project

1. Develop a backend system that collects and processes data from the deployed sensors, based on an open protocol that will allow for custom extensions (i.e. integrations with building management systems or HVAC)

2. Post-process the data, for example: normalize sensor values for their deviation, correlate different types of sensor values, compute derived metrics (human readable scale), predict future sensor data based on periodic patterns and detection of anomalies

3. Research which indoor air quality metrics should be displayed in the dashboard

4. Develop a dashboard front-end that displays real time data of a building

5. Develop a mechanism of notifying VTTI employees (e.g. Whatsapp, smart watch, dashboard, etc)

During the project VTTI will take on the role of product owner and stakeholder. There will be regular project meetings at the VTTI headquarters and the possibility to come by the office any time to either work on the project, conduct some tests, or obtain user feedback as the project progresses. There also will be regular meetings with Clairify. We at Clairify will assist the team in connecting to our sensor architecture as well as provide you with information as to how sensor data should be interpreted.

We are looking for enthusiastic students with a strong interest in time series data processing, databases and front-end applications who are willing to take the challenge of developing a new platform to improve indoor air quality within buildings.

G

Software Improvement Group feedback

Initial feedback

De code van het systeem scoort 3.2 sterren op ons onderhoudbaarheidsmodel, wat betekent dat de code marktgemiddeld onderhoudbaar is. We zien **Unit Interfacing** en **Unit Size** vanwege de lagere deelscores als mogelijke verbeterpunten.

Voor **Unit Interfacing** wordt er gekeken naar het percentage code in units met een bovengemiddeld aantal parameters. Doorgaans duidt een bovengemiddeld aantal parameters op een gebrek aan abstractie. Daarnaast leidt een groot aantal parameters nogal eens tot verwarring in het aanroepen van de methode en in de meeste gevallen ook tot langere en complexere methoden. Dit kan worden opgelost door parameter-objecten te introduceren, waarbij een aantal logischerwijs bij elkaar horende parameters in een nieuw object wordt ondergebracht. Dit geldt ook voor constructors met een groot aantal parameters, dit kan een reden zijn om de datastructuur op te splitsen in een aantal datastructuren. Als een constructor bijvoorbeeld acht parameters heeft die logischerwijs in twee groepen van vier parameters bestaan, is het logisch om twee nieuwe objecten te introduceren.

Voorbeelden in jullie project:

- `PMV.js:PMV`

Bij **Unit Size** wordt er gekeken naar het percentage code dat bovengemiddeld lang is. Dit kan verschillende redenen hebben, maar de meest voorkomende is dat een methode te veel functionaliteit bevat. Vaak was de methode oorspronkelijk kleiner, maar is deze in de loop van tijd steeds verder uitgebreid. De aanwezigheid van commentaar die stukken code van elkaar scheiden is meestal een indicator dat de methode meerdere verantwoordelijkheden bevat. Het opsplitsen van dit soort methodes zorgt er voor dat elke methode een duidelijke en specifieke functionele scope heeft. Daarnaast wordt de functionaliteit op deze manier vanzelf gedocumenteerd via methodenamen.

Voorbeelden in jullie project:

- `NetatmoDataResource.getMeasurements(String,String,Property...)`
- `OAuthCallbackServlet.doGet(HttpServletRequest,HttpServletResponse)`
- `NetatmoFetchDataTimerBean.fetchAllTheDatas()`
- `backend.py:getTrendyLine()`
- `backend.py:submitSurvey()`

Als laatste nog de opmerking dat er geen (unit)test-code is gevonden in de code-upload. Het is sterk aan te raden om in ieder geval voor de belangrijkste delen van de functionaliteit automatische tests gedefinieerd te hebben om ervoor te zorgen dat eventuele aanpassingen niet voor ongewenst gedrag zorgen. Op lange termijn maakt de aanwezigheid van unit tests je code ook flexibeler, omdat aanpassingen kunnen worden doorgevoerd zonder de stabiliteit in gevaar te brengen.

Final feedback

In de tweede upload zien we dat het codevolume is gegroeid, terwijl de score voor onderhoudbaarheid is gestegen.

We zien dat de verbeterpunten uit de feedback op de eerste upload zijn aangepast, en op deze gebieden is dan ook een verbetering in de deelscores te zien. Jullie hebben ook een vrij grote stap weten te maken op de genoemde gebieden.

Het is goed om te zien dat er naast nieuwe productiecode ook nieuwe testcode is geschreven.

Uit deze observaties kunnen we concluderen dat de aanbevelingen van de vorige evaluatie zijn meegenomen in het ontwikkeltraject.

H

Model parameters and characteristics

Table H.1: Common model parameters

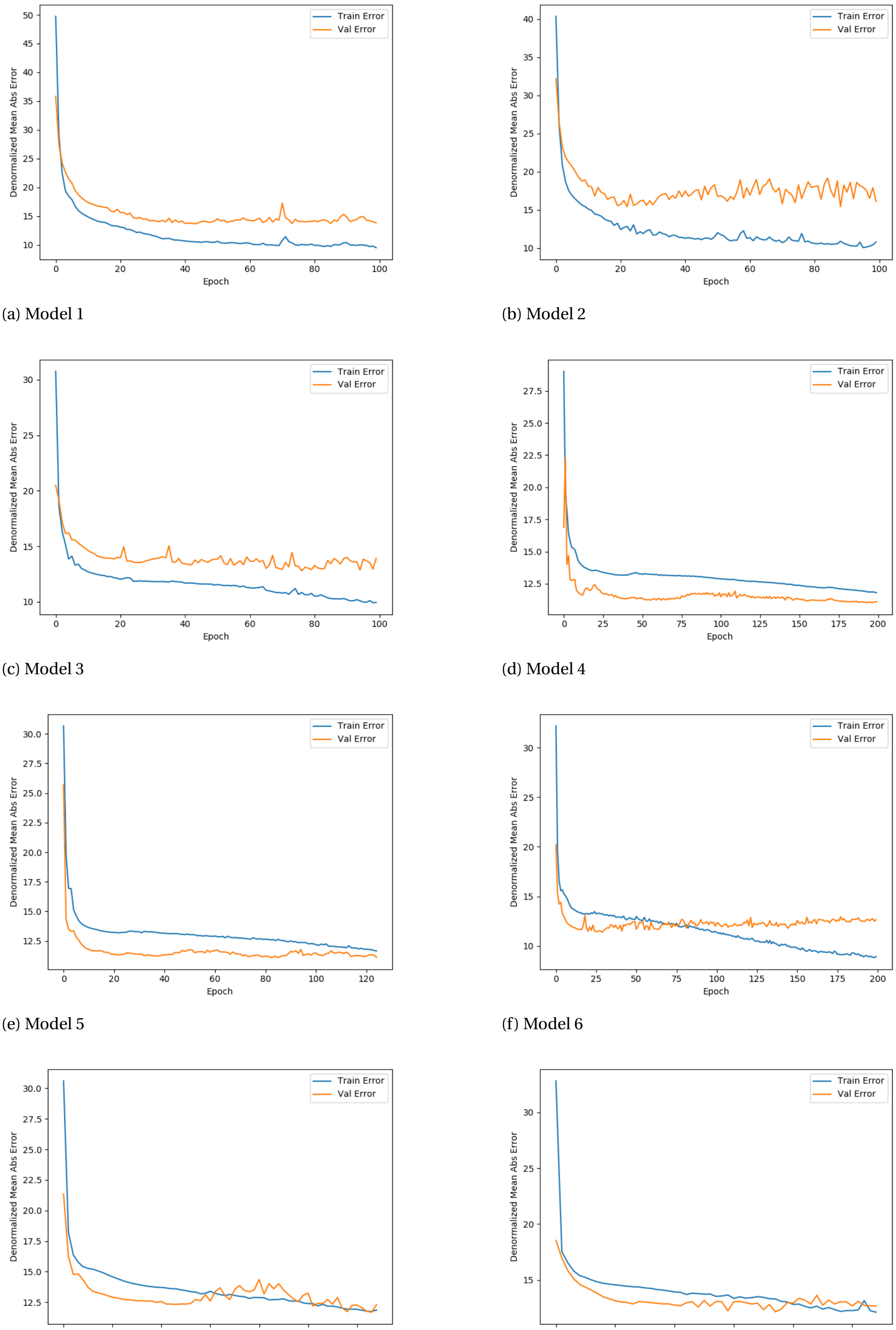
Batch size	128
Train : validation : test ratio	70 : 20 : 10

Table H.2: Parameters per model

Model	Loss function	Epochs	Train window	Prediction window	Resampling interval
1	MAE	100	96 samples (24 hours)	4 samples (1 hour)	15min
2	MSE	100	96 samples (24 hours)	4 samples (1 hour)	15min
3	MAE	100	288 samples (24 hours)	12 samples (1 hour)	5min
4	MAE	200	12 samples (1 hour)	12 samples (1 hour)	5min
5	MAE	125	24 samples (2 hours)	12 samples (1 hour)	5min
6	MAE	200	36 samples (3 hours)	12 samples (1 hour)	5min
7	MAE	65	84 samples (7 hours)	12 samples (1 hour)	5min
8	MAE	55	132 samples (11 hours)	12 samples (1 hour)	5min
9	MAE	100	288 samples (24 hours)	12 samples (1 hour)	5min
10	MAE	100	288 samples (24 hours)	12 samples (1 hour)	5min
11	MAE	100	288 samples (24 hours)	12 samples (1 hour)	5min
12	MAE	40	288 samples (24 hours)	12 samples (1 hour)	5min
13	MAE	200	672 samples (56 hours)	12 samples (1 hour)	5min
14	MAE	200	288 samples (24 hours)	12 samples (1 hour)	5min
15	MAE	200	288 samples (24 hours)	12 samples (1 hour)	5min
16	MAE	100	288 samples (24 hours)	12 samples (1 hour)	5min
17	MAE	200	288 samples (24 hours)	12 samples (1 hour)	5min
18	MAE	200	288 samples (24 hours)	12 samples (1 hour)	5min
19	MAE	400	288 samples (24 hours)	12 samples (1 hour)	5min

Table H.3: Evaluation results per model

Model	Loss	Normalized MAE	MAE	Normalized MSE
1	0.0373	0.0373	4.5674	0.0038
2	0.0081	0.0615	7.5309	0.0081
3	0.0370	0.0370	4.5100	0.0043
4	0.0410	0.0410	5.2367	0.0044
5	0.0385	0.0385	4.9087	0.0045
6	0.0374	0.0374	4.7765	0.0047
7	0.0400	0.0400	5.1015	0.0043
8	0.0420	0.0420	5.3588	0.0046
9	0.0389	0.0389	4.7454	0.0046
10	0.0372	0.0372	4.5302	0.0053
11	0.0393	0.0393	4.7900	0.0058
12	0.0371	0.0371	4.5255	0.0043
13	0.0583	0.0583	7.4118	0.0126
14	0.0383	0.0383	4.8895	0.0050
15	0.0356	0.0356	4.5419	0.0049
16	0.0524	0.0524	6.6940	0.0079
17	0.0357	0.0357	4.5573	0.0046
18	0.0389	0.0389	4.9721	0.0062
19	0.0345	0.0345	4.4057	0.0049



(a) Model 1

(b) Model 2

(c) Model 3

(d) Model 4

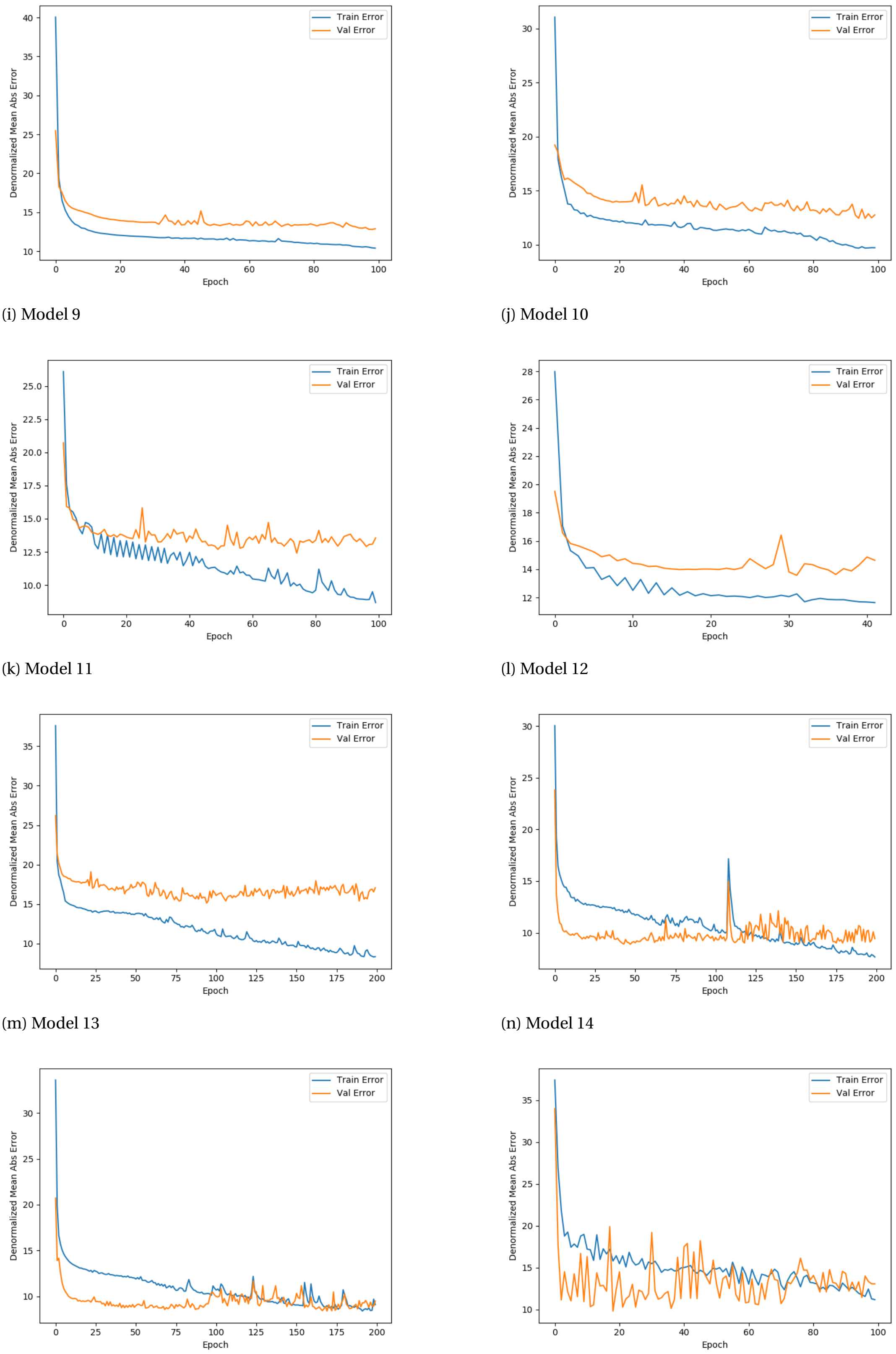
(e) Model 5

(f) Model 6

(g) Model 7

(h) Model 8

Figure H.1: Training and validation losses per epoch



(i) Model 9

(j) Model 10

(k) Model 11

(l) Model 12

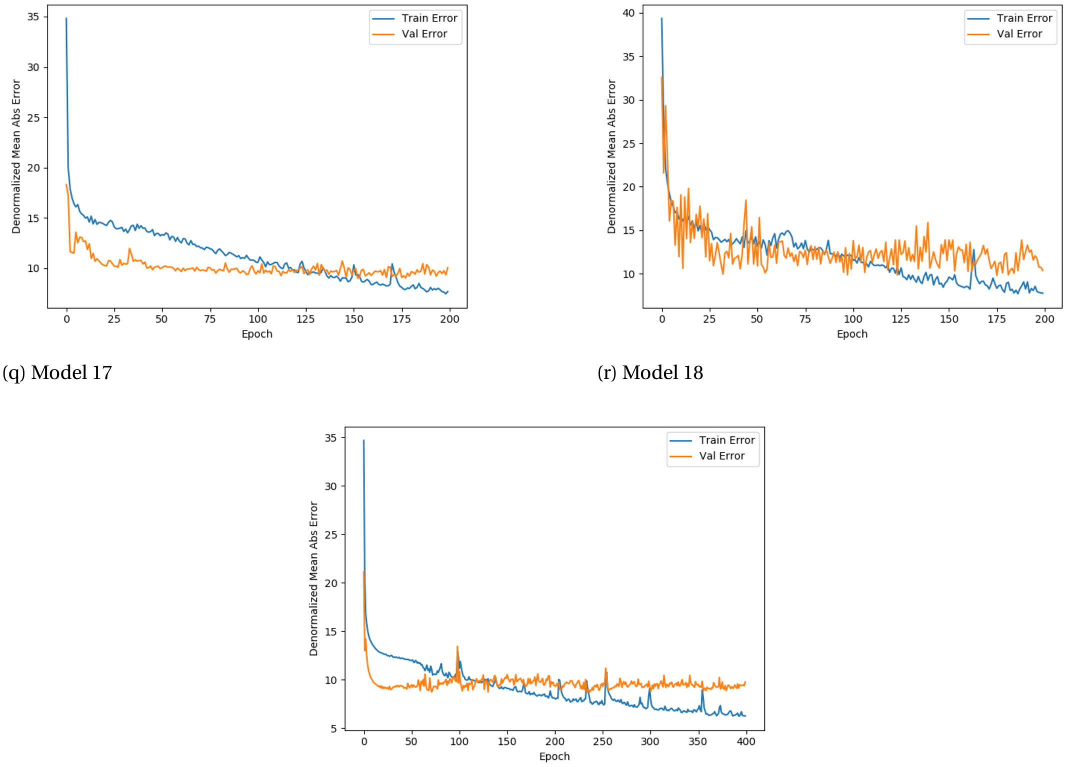
(m) Model 13

(n) Model 14

(o) Model 15

(p) Model 16

Figure H.1: Training and validation losses per epoch (cont.)

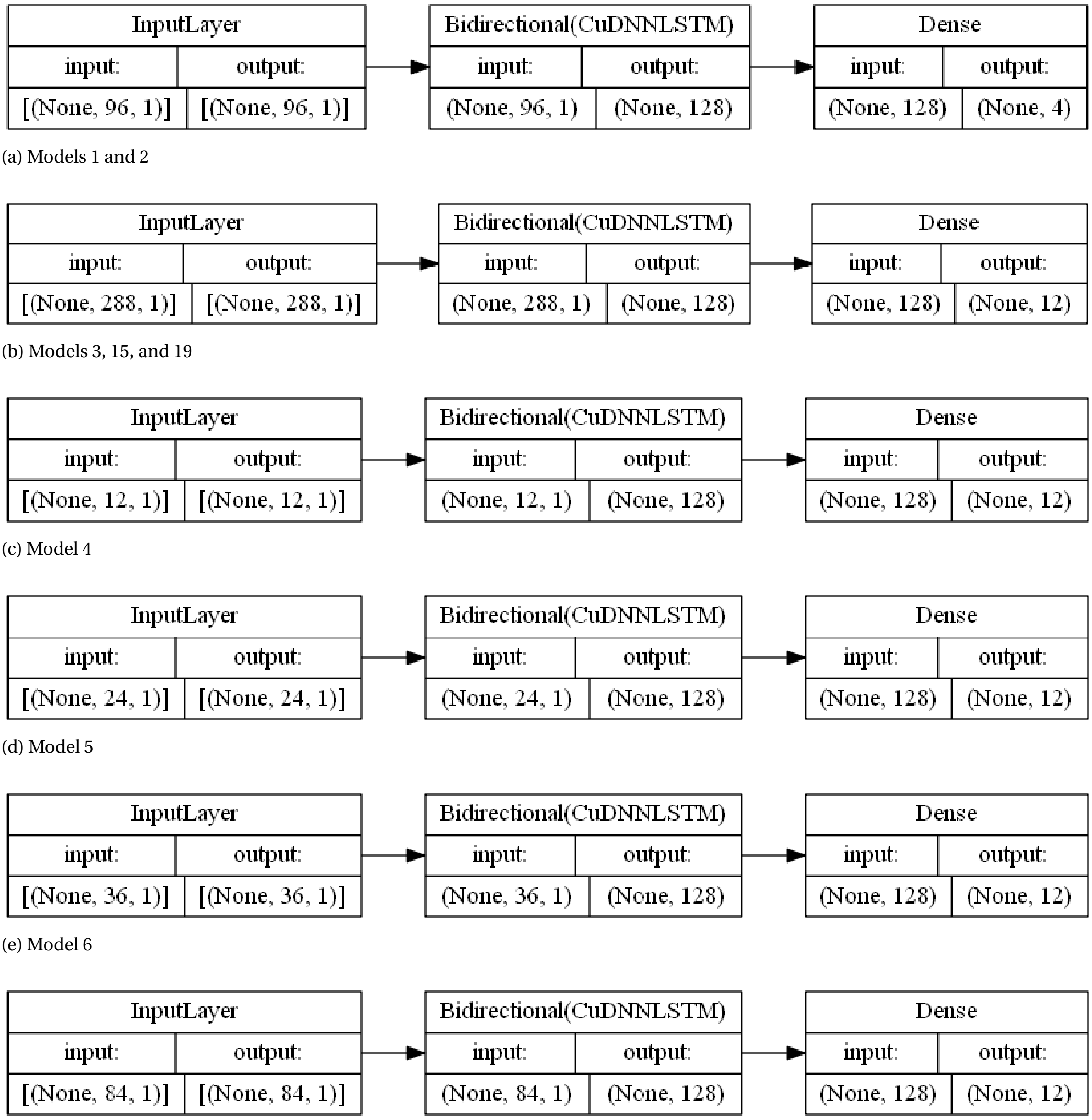


(q) Model 17

(r) Model 18

(s) Model 19

Figure H.1: Training and validation losses per epoch (cont.)



(a) Models 1 and 2

(b) Models 3, 15, and 19

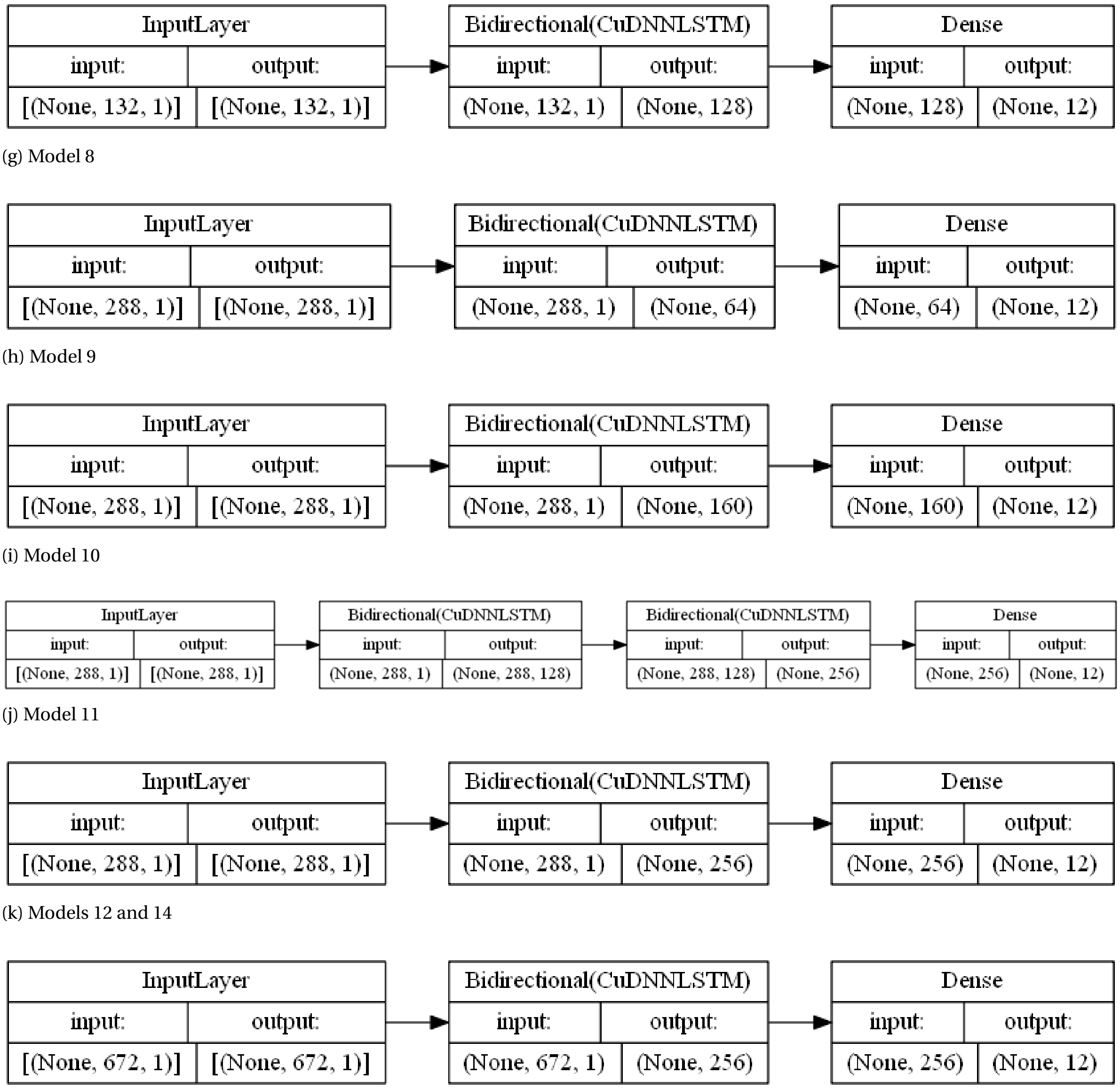
(c) Model 4

(d) Model 5

(e) Model 6

(f) Model 7

Figure H.2: Diagram of each model



(g) Model 8

(h) Model 9

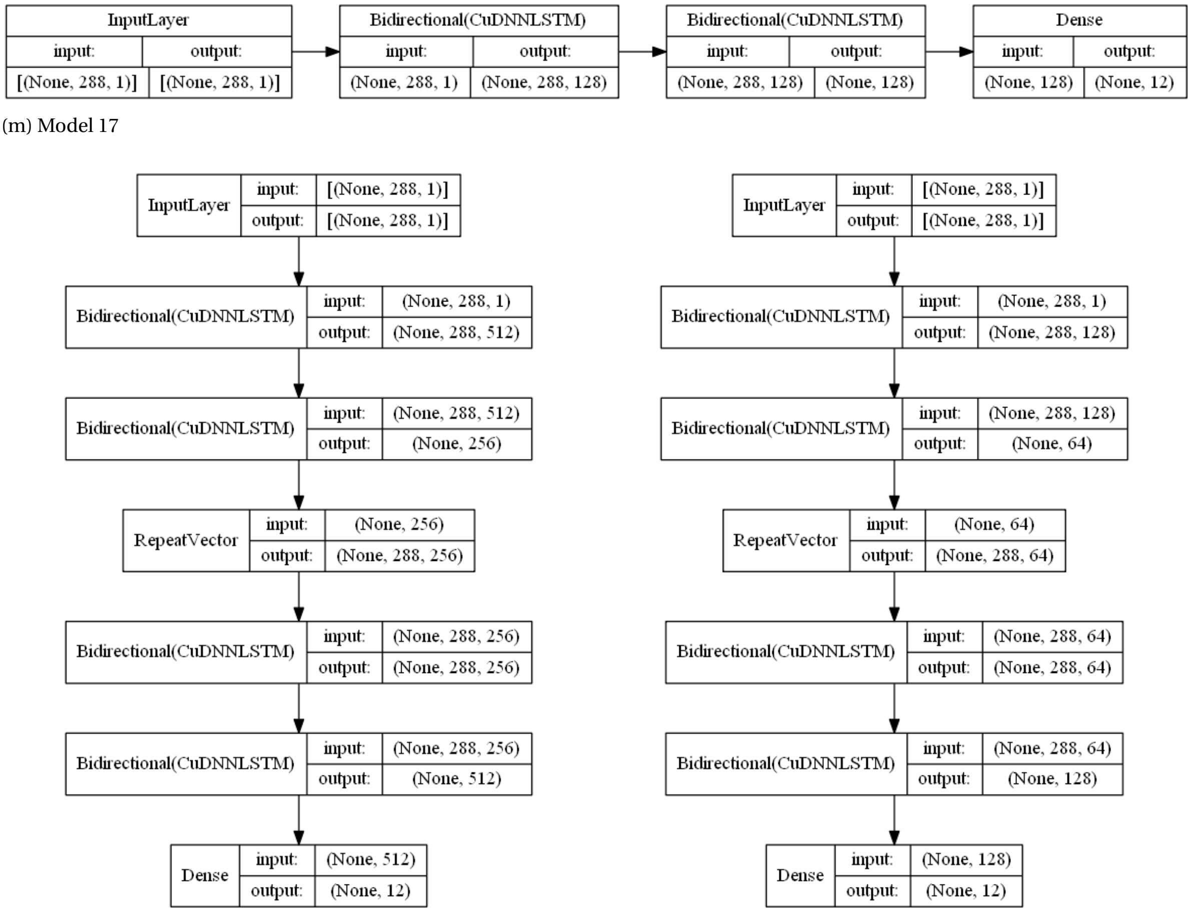
(i) Model 10

(j) Model 11

(k) Models 12 and 14

(l) Model 13

Figure H.2: Diagram of each model (cont.)



(m) Model 17

(n) Model 16

(o) Model 18

Figure H.2: Diagram of each model (cont.)

I

Final sensor

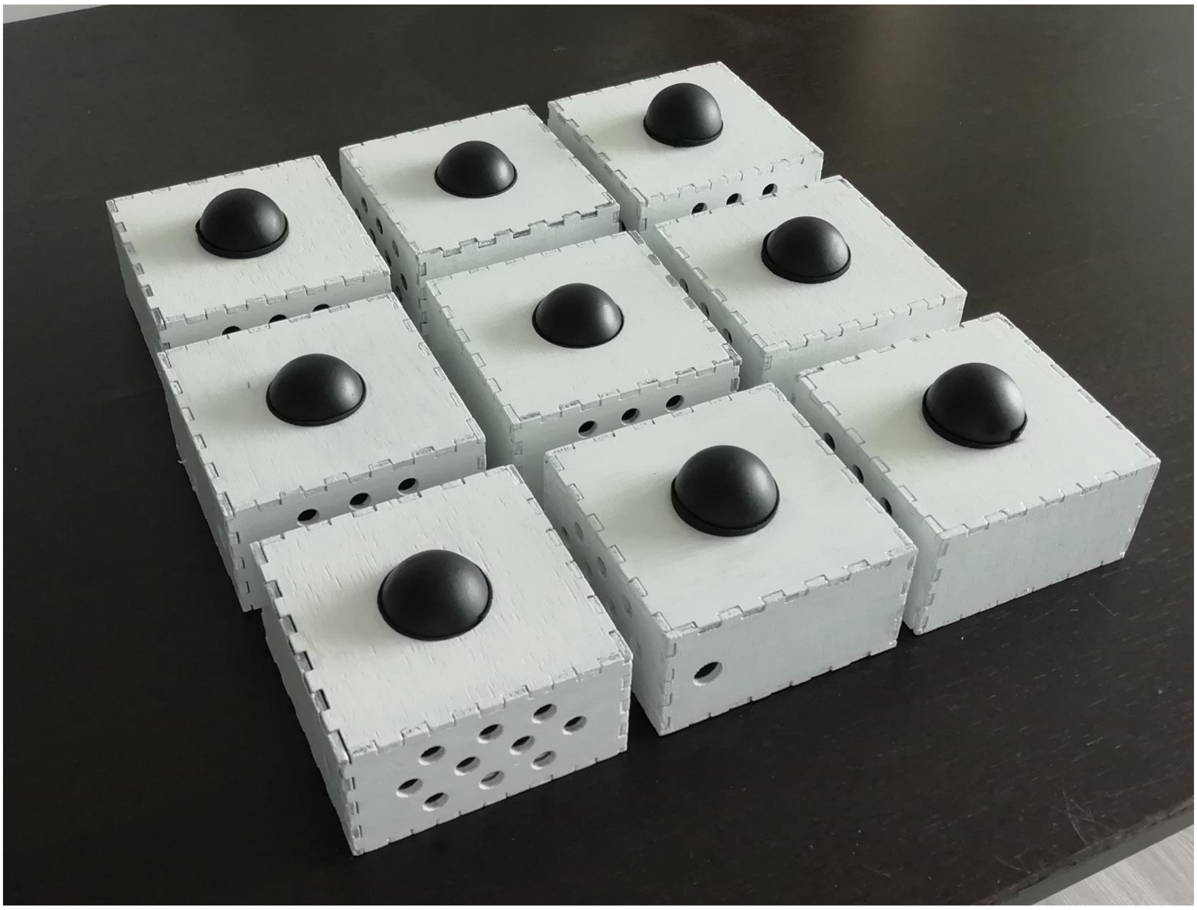


Figure I.1: The sensors that were built for this project

Glossary

Bluetooth A wireless communication technology frequently used in Internet of Things applications. 11, 12
Building Symptom Index A pseudo standardized scoring based on reported Sick Building symptoms among employees [5]. 1, 2

GeoJSON A JSON based format used for storing geographical features. 21

I2C A serial communication bus used in electronic circuits. 15
InfluxDB A popular time series database system. 13, 14, 29, 30

JSON Web Token A method of authentication through self-contained authentication tokens. 21

Likert scale A psychometric scale commonly used in questionnaires. 17
LoRaWAN A wireless communication technology frequently used in Internet of Things applications. 12

map rendering library a library involved in displaying geographical charts. 21
mesh Devices in a mesh topology connect to each other in a non-hierarchical manner, enabling any device in the network to act as a repeater. 12
message broker A message broker is an intermediary module that routes messages from senders to receivers. Message broker is an enterprise integration patterns described in *Enterprise Integration Patterns : Designing, Building, and Deploying Messaging Solutions*. 4, 13

neural network A computing system used in machine learning. 25

Particulate Matter an indoor air pollutant. 2–4, 36, 43, 47, 49
PostgreSQL A popular relational database system. 13
Predicted Mean Vote method for estimating the perceived thermal comfort of a group of people. 17, 41
publish-subscribe Publish–subscribe is a messaging pattern where senders of messages to not address specific receivers, but instead publish messages into topics without prior knowledge of which receivers are listening to this topic. Publish-subscribe is an enterprise integration patterns described in *Enterprise Integration Patterns : Designing, Building, and Deploying Messaging Solutions*. 13

REST Representational State Transfer (REST) is a software architecture style consisting of guidelines and best practices for creating scalable web services. REST was introduced by Fielding. 17

Sick building syndrome Sick building syndrome is a medical condition identified by a set of symptoms generally observed among office workers, is estimated to be present in 30% of all office buildings and can cause serious health damage over time [2]. 1, 41, 43, 57

templating library a library involved in binding data to the user interface at the client side. 21
time series A set of data points with temporal order. 4

UART A serial communication bus used in electronic circuits. 15

Z-Wave A wireless communication technology frequently used in home automation applications. 11, 12
ZigBee A wireless communication technology frequently used in home automation applications. 11, 12

Acronyms

GPU Graphics processing unit. 27, 30

HVAC Heating, ventilation and air conditioning. 1, 39, 43, 57

IAQ Indoor Air Quality - the air quality within buildings and structures. 31, 41, 43, 47, 49, 57

PAQ Perceived Air Quality - an umbrella of reported descriptors like temperature, presence of odor/smell, and experience of stuffy, dry or wet (humid) air [34]. 41, 42
PMV Predicted Mean Vote. 17, 31, 37, 39, 41, 42, 45, *Glossary:* Predicted Mean Vote
PPD Predicted Percentage Dissatisfied, derived from the Predicted Mean Vote. 15, 41, 42

SBS Sick Building Syndrome. 1, 41–43, 57, *Glossary:* Sick building syndrome

TVOC Total Volatile Organic Compounds - a uniform procedure to measure total volatile organic compounds (VOCs). 4, 43

VOC Volatile Organic Compounds - an indoor air pollutant. 3, 36, 41–43, 47–49

WHO World Health Organization. 43

Bibliography

[1] Neil Klepeis et al. “The National Human Activity Pattern Survey (NHAPS): A Resource for Assessing Exposure to Environmental Pollutants”. In: (Jan. 2001).

[2] United States Environmental Protection Agency. “Indoor Air Facts No. 4 (revised) Sick Building Syndrome”. In: *Research and Development (MD-56)* Air and Radiation (6609J) (1991). URL: `https://www.epa.gov/sites/production/files/2014-08/documents/sick_building_factsheet.pdf`.

[3] CBRE. “Global Workplace Solutions Research Review”. In: (2017).

[4] MetLife. “The Power of Employee Benefits in a Fast Changing Workplace”. In: (2017).

[5] Gary J. Raw et al. “Questionnaire design for sick building syndrome: An empirical comparison of options”. In: *Environment International* 22.1 (1996). The Psychology of Reactions to Environmental Agents, pp. 61–72. ISSN: 0160-4120. DOI: `https://doi.org/10.1016/0160-4120(95)00104-2`. URL: `http://www.sciencedirect.com/science/article/pii/0160412095001042`.

[6] A. Moreno-Rangel et al. “Field evaluation of a low-cost indoor air quality monitor to quantify exposure to pollutants in residential environments”. In: *Journal of Sensors and Sensor Systems* 7.1 (2018), pp. 373–388. DOI: `10.5194/jsss-7-373-2018`. URL: `https://www.j-sens-sens-syst.net/7/373/2018/`.

[7] C. R. Martin et al. “Evaluation and environmental correction of ambient CO_2 measurements from a low-cost NDIR sensor”. In: *Atmospheric Measurement Techniques* 10.7 (2017), pp. 2383–2395. DOI: `10.5194/amt-10-2383-2017`. URL: `https://www.atmos-meas-tech.net/10/2383/2017/`.

[8] S. Bhattacharya, S. Sridevi, and R. Pitchiah. “Indoor air quality monitoring using wireless sensor network”. In: *2012 Sixth International Conference on Sensing Technology (ICST)*. Dec. 2012, pp. 422–427. DOI: `10.1109/ICSensT.2012.6461713`.

[9] Jong-Jin Kim, Sung Kwon Jung, and Jeong Kim. “Wireless Monitoring of Indoor Air Quality by a Sensor Network”. In: *Indoor and Built Environment* 19 (Feb. 2010), pp. 145–150. DOI: `10.1177/1420326X09358034`.

[10] Sherin Abraham and Xinrong Li. “A Cost-effective Wireless Sensor Network System for Indoor Air Quality Monitoring Applications”. In: *Procedia Computer Science* 34 (2014). The 9th International Conference on Future Networks and Communications (FNC’14)/The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC’14)/Affiliated Workshops, pp. 165–171. ISSN: 1877-0509. DOI: `https://doi.org/10.1016/j.procs.2014.07.090`. URL: `http://www.sciencedirect.com/science/article/pii/S1877050914009454`.

[11] S. Mansour et al. “Wireless Sensor Network-based air quality monitoring system”. In: *2014 International Conference on Computing, Networking and Communications (ICNC)*. Feb. 2014, pp. 545–550. DOI: `10.1109/ICCNC.2014.6785394`.

[12] Tareq Alhmiedat. “A Survey on Environmental Monitoring Systems using Wireless Sensor Networks”. In: *Journal of Networks* 10 (Jan. 2016). DOI: `10.4304/jnw.10.11.606-615`.

[13] Ibrar Yaqoob et al. “Enabling Communication Technologies for Smart Cities”. In: *IEEE Communications Magazine* 55 (Jan. 2017). DOI: `10.1109/MCOM.2017.1600232CM`.

[14] Sakina Elhadi et al. “Comparative Study of IoT Protocols”. In: *SSRN Electronic Journal* (Jan. 2018). DOI: `10.2139/ssrn.3186315`.

[15] *Comparison of MQTT implementations - Wikipedia*. `https://en.wikipedia.org/wiki/Comparison_of_MQTT_implementations`. (Accessed on 06/03/2019).

[16] *Particle Reference Documentation | Cloud API*. `https://docs.particle.io/reference/device-cloud/api/`. (Accessed on 06/03/2019).

[17] *hirotakaster/MQTT: MQTT for Photon, Spark Core*. `https://github.com/hirotakaster/MQTT`. (Accessed on 06/03/2019).

[18] *PostgreSQL: Documentation: 10: 5.10. Table Partitioning.* https://www.postgresql.org/docs/10/ddl-partitioning.html. (Accessed on 06/03/2019).

[19] *TimescaleDB vs. PostgreSQL for time-series: 20x higher inserts, 2000x faster deletes, 1.2x-14,000x faster queries.* https://blog.timescale.com/timescaledb-vs-6a696248104e/. (Accessed on 06/03/2019).

[20] *DB-Engines Ranking - popularity ranking of time Series DBMS.* https://db-engines.com/en/ranking/time+series+dbms. (Accessed on 06/03/2019).

[21] Andreas Bader, Oliver Kopp, and Michael Falkenthal. "Survey and Comparison of Open Source Time Series Databases". In: *Datenbanksysteme für Business, Technologie und Web (BTW 2017) - Workshopband.* Ed. by Bernhard Mitschang et al. Bonn: Gesellschaft für Informatik e.V., 2017, pp. 249–268.

[22] *Introducing M3 – Towards Data Science.* https://towardsdatascience.com/introducing-m3-8790c503ce24. (Accessed on 06/03/2019).

[23] Henry Friday Nweke et al. "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges". In: *Expert Systems with Applications* 105 (2018), pp. 233–261. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2018.03.056. URL: http://www.sciencedirect.com/science/article/pii/S0957417418302136.

[24] Eric B. Baum and David Haussler. "What Size Net Gives Valid Generalization?" In: *Neural Comput.* 1.1 (Mar. 1989), pp. 151–160. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.1.151. URL: http://dx.doi.org/10.1162/neco.1989.1.1.151.

[25] Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural Networks* 61 (2015), pp. 85–117. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2014.09.003. URL: http://www.sciencedirect.com/science/article/pii/S0893608014002135.

[26] Z. Zhao et al. "LSTM network: a deep learning approach for short-term traffic forecast". In: *IET Intelligent Transport Systems* 11.2 (2017), pp. 68–75. ISSN: 1751-956X. DOI: 10.1049/iet-its.2016.0208.

[27] Daniel Hsu. "Time Series Forecasting Based on Augmented Long Short-Term Memory". In: *CoRR* abs/1707.00666 (2017). eprint: 1707.00666. URL: http://arxiv.org/abs/1707.00666.

[28] John Cristian Borges Gamboa. "Deep Learning for Time-Series Analysis". In: *CoRR* abs/1701.01887 (2017). arXiv: 1701.01887. URL: http://arxiv.org/abs/1701.01887.

[29] S. Selvin et al. "Stock price prediction using LSTM, RNN and CNN-sliding window model". In: *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI).* Sept. 2017, pp. 1643–1647. DOI: 10.1109/ICACCI.2017.8126078.

[30] R. J. Frank, N. Davey, and S. P. Hunt. "Time Series Prediction and Neural Networks". In: *J. Intell. Robotics Syst.* 31.1-3 (May 2001), pp. 91–103. ISSN: 0921-0296. DOI: 10.1023/A:1012074215150. URL: https://doi.org/10.1023/A:1012074215150.

[31] S. L. Ho, M. Xie, and T. N. Goh. "A Comparative Study of Neural Network and Box-Jenkins ARIMA Modeling in Time Series Prediction". In: *Comput. Ind. Eng.* 42.2-4 (June 2002), pp. 371–375. ISSN: 0360-8352. DOI: 10.1016/S0360-8352(02)00036-0. URL: http://dx.doi.org/10.1016/S0360-8352(02)00036-0.

[32] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. URL: http://www.deeplearningbook.org.

[33] Francois Chollet. *Deep Learning with Python.* 1st. Greenwich, CT, USA: Manning Publications Co., 2017. ISBN: 1617294438.

[34] Peder Wolkoff. "Indoor air humidity, air quality, and health – An overview". In: *International Journal of Hygiene and Environmental Health* 221.3 (2018), pp. 376–390. ISSN: 1438-4639. DOI: https://doi.org/10.1016/j.ijheh.2018.01.015. URL: http://www.sciencedirect.com/science/article/pii/S1438463917306946.

[35] David Wyon. "The effects of indoor air quality on performance and productivity". In: *Indoor air* 14 Suppl 7 (Feb. 2004), pp. 92–101. DOI: 10.1111/j.1600-0668.2004.00278.x.

[36] Jan Sundell and Thomas Lindvall. "Indoor Air Humidity and the Sensation of Dryness as Risk Indicators of SBS". In: *Indoor Air* 3 (Dec. 1993), pp. 382–390. DOI: 10.1111/j.1600-0668.1993.00024.x.

[37] Li Lan et al. "Effects of thermal discomfort in an office on perceived air quality, SBS symptoms, physiological responses, and human performance". In: *Indoor air* 21 (Feb. 2011), pp. 376–90. DOI: 10.1111/j.1600-0668.2011.00714.x.

[38] O.A. Seppänen, W.J. Fisk, and Mark Mendell. "Association of Ventilation Rates and CO2 Concentrations with Health andOther Responses in Commercial and Institutional Buildings". In: *Indoor air* 9 (Jan. 2000), pp. 226–52. DOI: 10.1111/j.1600-0668.1999.00003.x.

[39] Refrigerating American Society of Heating and Georgia) Air Conditioning Engineers (Atlanta. *ANSI/ASHRAE Standard 55-2013: Thermal Environmental Conditions for Human Occupancy*. ASHRAE standard. ASHRAE, 2013. URL: https://books.google.nl/books?id=AGxFjwEACAAJ.

[40] Ashrae. *2017 ASHRAE Handbook: Fundamentals*. ASHRAE Handbook Fundamentals Inch-Pound System. ASHRAE, 2017. ISBN: 9781939200570. URL: https://books.google.nl/books?id=6VmztwEACAAJ.

[41] P.O. Fanger and B. Berg-Munch. "The influence of air temperature on the perception of body odor". In: *Environment International* 8 (1982), pp. 333–335. DOI: https://doi-org.tudelft.idm.oclc.org/10.1016/0160-4120(82)90046-0.

[42] C.W. Bayer et al. "THE RELATIONSHIP BETWEEN HUMIDITY AND INDOOR AIR QUALITY IN SCHOOLS". In: *Proceedings: Indoor Air 2002* 1 (June 2002).

[43] Dan Norbäck, Klas Nordström, and Zhuohui Zhao. "Carbon dioxide (CO2) demand-controlled ventilation in university computer classrooms and possible effects on headache, fatigue and perceived indoor environment: An intervention study". In: *International archives of occupational and environmental health* 86 (Mar. 2012). DOI: 10.1007/s00420-012-0756-6.

[44] Dai-Hua Tsai, Jia-Shiang Lin, and Chang-Chuan Chan. "Office Workers' Sick Building Syndrome and Indoor Carbon Dioxide Concentrations". In: *Journal of occupational and environmental hygiene* 9 (May 2012), pp. 345–51. DOI: 10.1080/15459624.2012.675291.

[45] Christine Erdmann, Kate C. Steiner, and Michael Apte. "Indoor carbon dioxide concentrations and sick building syndrome symptoms in the BASE study revisited: Analyses of the 100 building dataset". In: *Proceedings of Indoor Air 2002* 3 (Jan. 2002).

[46] Xiaojing Zhang, Pawel Wargocki, and Zhiwei Lian. "Physiological Responses during Exposure to Carbon Dioxide and Bioeffluents at Levels Typically Occurring Indoors". In: *Indoor air* 27 (Feb. 2016). DOI: 10.1111/ina.12286.

[47] A.N. Myhrvold, E Olsen, and O Lauridsen. "Indoor environment in schools - Pupils health and performance in regard to CO2 concentrations". In: *Proceedings of Indoor Air '96: The 7th International Conference on Indoor Air Quality and Climate* 4 (Jan. 1996), pp. 369–374.

[48] Joseph Allen et al. "Associations of Cognitive Function Scores with Carbon Dioxide, Ventilation, and Volatile Organic Compound Exposures in Office Workers: A Controlled Exposure Study of Green and Conventional Office Environments". In: *Environmental Health Perspectives* 124 (Oct. 2015). DOI: 10.1289/ehp.1510037.

[49] Shaobin Wang, H.M. Ang, and Moses O. Tade. "Volatile organic compounds in indoor environment and photocatalytic oxidation: State of the art". In: *Environment International* 33.5 (2007), pp. 694–705. ISSN: 0160-4120. DOI: https://doi.org/10.1016/j.envint.2007.02.011. URL: http://www.sciencedirect.com/science/article/pii/S0160412007000281.

[50] Hakan Sevik et al. "The Influence of House Plants on Indoor CO2". In: *Polish Journal of Environmental Studies* 26 (June 2017). DOI: 10.15244/pjoes/68875.

[51] Peder Wolkoff and Gunnar D Nielsen. "Organic compounds in indoor air—their relevance for perceived indoor air quality?" In: *Atmospheric Environment* 35.26 (2001), pp. 4407–4417. ISSN: 1352-2310. DOI: https://doi.org/10.1016/S1352-2310(01)00244-8. URL: http://www.sciencedirect.com/science/article/pii/S1352231001002448.

[52] Yinping Zhang et al. "Influence of temperature on formaldehyde emission parameters of dry building materials". In: *Atmospheric Environment* 41.15 (2007). Indoor Air 2005 - 10th International Conference on Indoor Air Quality and Climate (Part I), pp. 3203–3216. ISSN: 1352-2310. DOI: https://doi.org/10.1016/j.atmosenv.2006.10.081. URL: http://www.sciencedirect.com/science/article/pii/S1352231006011459.

[53] Simone Simões Amaral et al. "An Overview of Particulate Matter Measurement Instruments". In: *Atmosphere* 6 (Sept. 2015), pp. 1327–1345. DOI: 10.3390/atmos6091327.

[54] Anondo Mukherjee et al. "Assessing the Utility of Low-Cost Particulate Matter Sensors over a 12-Week Period in the Cuyama Valley of California". In: *Sensors* 17.8 (2017). ISSN: 1424-8220. DOI: 10.3390/s17081805. URL: http://www.mdpi.com/1424-8220/17/8/1805.

[55] Jayakarthigeyan Prabakar, Vysakh Mohan, and Karthik Ravisankar. "Evaluation of low cost particulate matter sensor for indoor air quality measurement". In: *Evaluation* 4.2 (2015).

[56] I.S. Walker and D.J. Wilson. "Evaluating Models for Superposition of Wind and Stack Effect in Air Infiltration". In: *Building and Environment* 28.2 (1993), pp. 201–210. DOI: https://doi.org/10.1016/0360-1323(93)90053-6. URL: https://doi.org/10.1016/0360-1323(93)90053-6.

[57] Jongwon Kwon et al. "A study on NDIR-based CO2 sensor to apply remote air quality monitoring system". In: *ICCAS-SICE 2009 - ICROS-SICE International Joint Conference 2009, Proceedings* (Jan. 2009).

[58] Tomomi Yasuda, Seiichiro Yonemura, and Akira Tani. "Comparison of the Characteristics of Small Commercial NDIR CO2 Sensor Models and Development of a Portable CO2 Measurement Device". In: *Sensors* 12.3 (2012), pp. 3641–3655. ISSN: 1424-8220. DOI: 10.3390/s120303641. URL: http://www.mdpi.com/1424-8220/12/3/3641.

[59] Xiao Liu et al. "A Survey on Gas Sensing Technology". In: *Sensors* 12.7 (2012), pp. 9635–9665. ISSN: 1424-8220. DOI: 10.3390/s120709635. URL: http://www.mdpi.com/1424-8220/12/7/9635.

[60] R. Zhou et al. "Reliable CO2 sensors with silicon-based polymers on quartz microbalance transducers". In: *Sensors and Actuators B: Chemical* 19.1 (1994), pp. 415–420. ISSN: 0925-4005. DOI: https://doi.org/10.1016/0925-4005(93)01018-Y. URL: http://www.sciencedirect.com/science/article/pii/092540059301018Y.

[61] SeungHwan Yi et al. "Novel NDIR CO/sub 2/ sensor for indoor air quality monitoring". In: *The 13th International Conference on Solid-State Sensors, Actuators and Microsystems, 2005. Digest of Technical Papers. TRANSDUCERS '05.* 2 (2005), 1211–1214 Vol. 2.

[62] Alphasense. *IRC-A1 CARBON DIOXIDE INFRARED SENSOR.* 2019. URL: http://www.alphasense.com/WEB1213/wp-content/uploads/2018/04/IRC-A1.pdf (visited on 04/23/2019).

[63] SenseAir. *K30.* 2019. URL: https://senseair.com/products/flexibility-counts/senseair-k30/ (visited on 04/23/2019).

[64] SenseAir. *K33 ELG.* 2019. URL: https://senseair.com/products/flexibility-counts/k33-elg/ (visited on 04/23/2019).

[65] SenseAir. *SenseAir S8 LP.* 2019. URL: https://senseair.com/products/size-counts/senseair-s8-lp/ (visited on 04/23/2019).

[66] SenseAir. *LP8.* 2019. URL: https://senseair.com/products/power-counts/lp8/ (visited on 04/23/2019).

[67] ELT Sensor. *CO2 Sensor Module T-110.* 2019. URL: http://eltsensor.co.kr/2012/eng/product/co2-sensor-module-T110.html (visited on 04/23/2019).

[68] Winsen. *MH-Z14A User Manual.* 2019. URL: https://www.winsen-sensor.com/d/files/infrared-gas-sensor/mh-z14a_co2-manual-v1_01.pdf (visited on 04/23/2019).

[69] Winsen. *MH-Z19B User Manual.* 2019. URL: https://www.winsen-sensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-ver1_0.pdf (visited on 04/23/2019).

[70] Sensirion. *SCD30 - Sensor Module for HVAC and Indoor Air Quality Applications.* 2019. URL: https://www.sensirion.com/en/environmental-sensors/carbon-dioxide-sensors-co2/ (visited on 04/23/2019).

[71] Amphenol Sensors. *Telaire T6700 Series.* 2019. URL: https://www.amphenol-sensors.com/en/telaire/co2/525-co2-sensor-modules/3215-t6700 (visited on 04/24/2019).

[72] Laurent Spinelle et al. "Review of Portable and Low-Cost Sensors for the Ambient Air Monitoring of Benzene and Other Volatile Organic Compounds". In: *Sensors* 17.7 (2017). ISSN: 1424-8220. DOI: 10.3390/s17071520. URL: http://www.mdpi.com/1424-8220/17/7/1520.

[73] A. Mirzaei, S.G. Leonardi, and G. Neri. "Detection of hazardous volatile organic compounds (VOCs) by metal oxide nanostructures-based gas sensors: A review". In: *Ceramics International* 42.14 (2016), pp. 15119–15141. ISSN: 0272-8842. DOI: `https://doi.org/10.1016/j.ceramint.2016.06.145`. URL: `http://www.sciencedirect.com/science/article/pii/S0272884216309889`.

[74] *8096-AP AIR MENTOR PRO Teardown Internal Photos CoAsia Microelectronics*. `https://fccid.io/2AEGI-8096-AP/Internal-Photos/Internal-Photos-3110733`. (Accessed on 06/12/2019).

[75] Ashley M. Collier-Oxandale et al. "Understanding the ability of low-cost MOx sensors to quantify ambient VOCs". In: *Atmospheric Measurement Techniques* 12 (Mar. 2019), pp. 1441–1460. DOI: `10.5194/amt-12-1441-2019`.

[76] Figaro. *TGS 2602 Product Information*. 2019. URL: `http://www.figarosensor.com/product/docs/TGS2602-B00%5C%20%5C%280615%5C%29.pdf` (visited on 04/23/2019).

[77] AMS. *iAQ-Core Indoor Air Quality Sensor Module*. 2019. URL: `https://ams.com/documents/20143/36005/iAQ-core_DS000334_1-00.pdf/123c67cc-92d4-9a5c-d7ca-24f1d439c6a4` (visited on 04/23/2019).

[78] Sensirion. *Multi-Pixel Gas Sensors SGP*. 2019. URL: `https://www.sensirion.com/en/environmental-sensors/gas-sensors/multi-pixel-gas-sensors/` (visited on 04/23/2019).

[79] Sensirion. *Multi-Pixel Gas Sensors SGP*. 2019. URL: `https://www.sensirion.com/en/environmental-sensors/gas-sensors/` (visited on 04/27/2019).

[80] Sinan Sousan et al. "Inter-comparison of low-cost sensors for measuring the mass concentration of occupational aerosols". In: *Aerosol Science and Technology* 50.5 (2016). PMID: 28867868, pp. 462–473. DOI: `10.1080/02786826.2016.1162901`. eprint: `https://doi.org/10.1080/02786826.2016.1162901`. URL: `https://doi.org/10.1080/02786826.2016.1162901`.

[81] Yang Wang et al. "Laboratory Evaluation and Calibration of Three Low-Cost Particle Sensors for Particulate Matter Measurement". In: *Aerosol Science and Technology* 49.11 (2015), pp. 1063–1077. DOI: `10.1080/02786826.2015.1100710`. URL: `https://doi.org/10.1080/02786826.2015.1100710`.

[82] Kirsten A. Koehler and Thomas M. Peters. "New Methods for Personal Exposure Monitoring for Airborne Particles". In: *Current Environmental Health Reports* 2.4 (Dec. 2015), pp. 399–411. ISSN: 2196-5412. DOI: `10.1007/s40572-015-0070-z`. URL: `https://doi.org/10.1007/s40572-015-0070-z`.

[83] Sinan Sousan et al. "Evaluation of consumer monitors to measure particulate matter". In: *Journal of Aerosol Science* 107 (2017), pp. 123–133. ISSN: 0021-8502. DOI: `https://doi.org/10.1016/j.jaerosci.2017.02.013`. URL: `http://www.sciencedirect.com/science/article/pii/S002185021630338X`.

[84] Sotirios P. *Review: uHoo AQI Monitor*. 2017. URL: `https://seetheair.wordpress.com/2017/09/04/review-uhoo-aqi-monitor/` (visited on 04/28/2019).

[85] Defeng Qian, Zhihong Pang, and Zheng O'Neill. "Evaluation of Low-cost Particulate Matter (PM) Sensors: A Preliminary Investigation". In: June 2018.

[86] Sharp. *Air Sensors GP2Y1010AU0F*. 2019. URL: `https://www.sharpsde.com/products/optoelectronic-components/model/GP2Y1010AU0F/` (visited on 04/24/2019).

[87] Sharp. *Sharp Sensor Products*. 2019. URL: `https://media.digikey.com/pdf/Data%5C%20Sheets/Sharp%5C%20PDFs/Sharp%5C%20Sensor%5C%20Product%5C%20Intro-%5C%20Dust%5C%20and%5C%20T&H.pdf` (visited on 04/24/2019).

[88] Seeedstudio. *Grove - Dust SensorPPD42NS*. 2019. URL: `https://www.seeedstudio.com/Grove-Dust-Sensor-PPD42NS.html` (visited on 04/24/2019).

[89] Alphasense. *OPC-N3 Particle Monitor*. 2019. URL: `http://www.alphasense.com/WEB1213/wp-content/uploads/2019/03/OPC-N3.pdf` (visited on 04/24/2019).

[90] Alphasense. *OPC-R1 Particle Monitor*. 2019. URL: `http://www.alphasense.com/WEB1213/wp-content/uploads/2019/03/OPC-R1.pdf` (visited on 04/24/2019).

[91] Honeywell. *HPM Series Particle Sensor*. 2019. URL: `https://sensing.honeywell.com/honeywell-sensing-hpm-series-particle-sensors-datasheet-32322550-e-en.pdf` (visited on 04/24/2019).

[92] Sensirion. *Particulate Matter Sensor SPS30*. 2019. URL: https://www.sensirion.com/en/environmental-sensors/particulate-matter-sensors-pm25/ (visited on 04/27/2019).

[93] Sensirion. *Datasheet SHT3x-DIS*. 2019. URL: https://www.sensirion.com/en/environmental-sensors/humidity-sensors/digital-humidity-sensors-for-various-applications/ (visited on 04/23/2019).

[94] Sensirion. *Humidity Sensors with CMOSens*. 2019. URL: https://www.sensirion.com/en/about-us/technology/cmosens-technology-for-humidity/ (visited on 04/24/2019).

[95] Bosch. *Datasheet BME280*. 2019. URL: https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME280-DS002.pdf (visited on 04/23/2019).

[96] Bosch. *Datasheet BMP180*. 2019. URL: https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BMP180-DS000.pdf (visited on 04/23/2019).

[97] Maxim Integrated. *Datasheet DS18B20*. 2019. URL: https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf (visited on 04/23/2019).

[98] Adafruit. *DHT11, DHT22 and AM2302 Sensors*. 2019. URL: https://learn.adafruit.com/dht (visited on 04/23/2019).

[99] *KNMI - Daggegevens van het weer in Nederland - Download*. http://projects.knmi.nl/klimatologie/daggegevens/selectie.cgi. (Accessed on 06/03/2019).

[100] Gregor Hohpe and Bobby Woolf. *Enterprise Integration Patterns : Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Professional, Oct. 2003. ISBN: 0321200683. URL: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20%5C&amp;path=ASIN/0321200683.

[101] Roy Thomas Fielding. "Architectural Styles and the Design of Network-based Software Architectures". AAI9980887. PhD thesis. 2000. ISBN: 0-599-87118-0.