Master's thesis
Master's Programme in Data Science

Modelling Indoor Air Quality Using Sensor Data and Machine Learning Methods

Dennis Muiruri

February 28, 2021

Supervisor(s): Dr. Mikko Raatikainen

Examiner(s): Professor Tommi Mikkonen
Dr. Mikko Raatikainen

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Degree programme
Faculty of Science	Master's Programme in Data Science

Tekijä — Författare — Author

Dennis Muiruri

Työn nimi — Arbetets titel — Title

Modelling Indoor Air Quality Using Sensor Data and Machine Learning Methods

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidantal — Number of pages
Master's thesis	February 28, 2021	55

Tiivistelmä — Referat — Abstract

Ubiquitous sensing is transforming our societies and how we interact with our surrounding environment; sensors provide large streams of data while machine learning techniques and artificial intelligence provide the tools needed to generate insights from the data. These developments have taken place in almost every industry sector with topics such as smart cities and smart buildings becoming key topical issues as societies seek more sustainable ways of living. Smart buildings are the main context of this thesis. These are buildings equipped with various sensors used to collect data from the surrounding environment allowing the building to adapt itself and increasing its operational efficiency.

Previously, most efforts in realizing smart buildings have focused on energy management and automation where the goal is to improve costs associated with heating, ventilation, and air conditioning. A less studied area involves smart buildings and their indoor environments especially relative to sub-spaces within a building. Increased developments in low-cost sensor technologies have created new opportunities to sense indoor environments in more granular ways that provide new possibilities to model finer attributes of spaces within a building.

This thesis focuses on modeling indoor environment data obtained from a multipurpose building that serves primarily as a school. The aim is to explore the quality of the indoor environment relative to regulatory guidelines and also exploring suitable predictive models for thermal comfort and indoor air quality. Additionally, design science methodology is applied in the creation of a proof of concept software system. This system is aimed at demonstrating the use of Web APIs to provide sensor data to clients that may use the data to render analytics among other insights to a building's stakeholders.

Overall, the main technical contributions of this thesis are twofold: (i) a potential web-application design for indoor air quality IoT data and (ii) an exposition of modeling of indoor air quality data based on a variety of sensors and multiple spaces within the same building.

Results indicate a software-based tool that supports monitoring the indoor environment of a building would be beneficial in maintaining the correct levels of various indoor parameters. Further, modeling data from different spaces within the building shows a need for heterogeneous models to predict variables in these spaces. This implies parameters used to predict thermal comfort and air quality are different in varying spaces especially where the spaces differ in size, indoor climate control settings, and other attributes such as occupancy control.

Avainsanat — Nyckelord — Keywords

IoT, Sensors, Smart Buildings, Indoor Air Quality, Predicting, Machine Learning

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1 Introduction **1**

2 Background **3**

2.1 Cyber-Physical Systems 4

2.2 Digital Twin 5

2.3 Sensors and Internet of Things 6

2.4 REST APIs and Service Oriented Architecture 8

3 Research Methodology **10**

3.1 Problem Identification and Motivation 10

3.2 Definition of the objectives 11

3.3 Design and Development 11

3.4 Demonstration 11

3.5 Evaluation and Communication 11

4 Technical Solution **12**

4.1 System Components 13

4.1.1 Data Storage 13

4.1.2 Data API Server 14

4.2 Analytics API Server 18

4.3 ML Models 18

4.4 Weather Data 19

4.5 ML Model Training 19

4.6 Front End 19

5 Data and Machine Learning Models **21**

5.1 Data Exploration 21

5.2 Modeling thermal comfort and indoor air quality 22

5.3 Predicting elements of thermal comfort and indoor air quality 27

5.3.1 Modeling CO_2 and Temperature using indoor climate variables . . . 28

5.3.2 Feature Selection by forward selection 30
5.3.3 Modeling CO_2 and Temperature using boosted regression trees . . . 33
5.3.4 Modeling CO_2 and Temperature using ARIMA models 36
5.4 Data modeling results 39

6 Discussion **42**
6.1 Data modeling 42
6.2 Application design 43
6.3 Validity 44
6.4 Future considerations 45

7 Conclusions **46**

Bibliography **47**

Appendix A **52**

1. Introduction

Smart infrastructure and buildings are increasingly emerging as one of the ways smart cities can be realized along with the transition towards more sustainable societies. This has mainly resulted from the convergence of the Internet of Things (IoT), wireless sensor networks, and cloud services among other enabling technologies. In this setting, sensors are embedded in physical entities to collect data which is further analyzed for insights on the utility of the given infrastructure. In general, IoT is anticipated to transform how buildings are managed and also lead to enhanced building management systems [29].

Innovations in smart building technologies tend to often focus on improving operational costs incurred by running heating, ventilation and air conditioning (HVAC) systems due to the significant operational costs associated with these items [14] [31]. Space heating, cooling, and ventilation are indicated to consume about 32.7% of a commercial building's electricity while lighting and office equipment accounts for 17.1% and 13.6% respectively. These large shares of electricity consumption provide an opportunity for cost optimization and hence better facility management [29].

An equally important aspect of buildings is the well-being of occupants especially as a result of indoor air quality (IAQ). It is estimated that people spend a significant amount of time(80%) indoors and require about $12m^3$ of clean air per day[†]. This motivates the need to investigate the quality of the indoor air across buildings as early steps to build the relevant building blocks required to realize intelligent or smart buildings. A condition termed sick building syndrome(SBS) has been indicated to affect occupants of a building after being indoors for long periods in a setting where the HVAC system is not optimized for good indoor air quality. SBS refers to a situation where occupants of a building experience symptoms such as headache, eye, nose, throat irritation, and fatigue after being in a building for prolonged periods [21].

Optimizing for both good indoor air quality and a building's energy cost can be somewhat two antagonistic objectives. To maintain good indoor air quality, the HVAC system may need to run at a given level that supplies the correct conditions for good air quality and thermal comfort. On the other hand, if the primary goal is to save energy costs associated with running the HVAC system, then the optimal indoor air quality might not

[†]https://www.vttresearch.com/en/ourservices/smart-buildings

be achieved. Data analysis provides an opportunity to investigate the indoor air quality and the effects of the HVAC system on the indoor environment.

There are two distinct sets of existing research efforts associated with smart buildings. One focuses on smart buildings intending to improve HVAC costs, the second set of studies are those that focus on modeling temperature and carbon dioxide as elements of thermal comfort and indoor air quality. Often this second category of studies do not indicate any other parameters of the indoor environment and neither is their exclusion justified.

This study aims to augment knowledge on studies that model temperature and carbon dioxide especially by considering other indoor environment variables measured through multiple sensors. Additionally, this study will segment the spaces in a building as distinct rooms to study for potential structural differences evident from the collected data. The reference building in this study is a multi-purpose building named Ypsilon*, located in the city of Turku, Finland. Ypsilon serves as an early education school but also provides facilities for other public services. The objectives of this study are outlined as follows:

- Perform exploratory analysis on sensor data obtained from Ypsilon building to generate insights on the indoor air quality.
- Create machine learning models from the data to predict selected parameters that constitute thermal comfort.
- Design an application that allows users to explore this data and serve as a prototype for a potential indoor air quality management platform.

Following a design science methodology [34], this study presents a software design and system that is used to visualize analytics, and predictive modeling results generated from the data are also presented.

The rest of the thesis is organized as follows: Chapter 2 contains the background literature where a review of cyber-physical systems and IoT solutions, IoT, and REST APIs are discussed as integrated topics. Chapter 3 is a presentation of the theoretical methodology applied in this thesis while Chapter 4 presents the technical solution implemented to show a proof of concept. Chapter 5 is an exploration of the data used in the development of analytics and machine learning aspects of the solution. Chapter 6 is a discussion section where the overall results and their implications are discussed. Chapter 7 contains this study's concluding remarks.

*https://www.turku.fi/toimipaikat/yli-maarian-monitoimitalo-ypsilon-yli-maarian-koulu

2. Background

This chapter serves as a high-level literature discussion covering Cyber-Physical Systems as a general discussion on replicating real physical objects to a computational representation. A discussion of Digital Twins as a recent evolution of Cyber-Physical systems and a more concrete form of Cyber-Physical Systems is also presented.

Given the increasingly pervasive sensors, more and varying types of data can be collected. Sensors are networked to form IoT and therefore data collection networks implicitly emerge. Due to the potentially wide scope of data that can be collected, big data challenges such as storage and computational resource constraints become evident. However, collecting data is still necessary as it provides the input required to generate insights through statistical methods, machine learning (ML), and/or Artificial Intelligence (AI) models. A generic review of how sensors and their related networking technologies have evolved is discussed in this chapter.

To create a web-based solution, the Service Oriented Architecture (SOA) provides a generic and logical architectural blueprint that can be used as a design reference. Such services tend to be composed of multiple sub-components and therefore an integration model is required to facilitate data exchange across components and external third party consumers. Web-based Application Programming Interfaces (APIs) provide one of the integration solutions and for both internal and external service consumers. Such web APIs tend to make use of the REpresentational State Transfer (REST) [13] protocol for data exchange. This topic is discussed further in this chapter particularly focusing on REST-based APIs in the context of an IoT solution.

Generally, the main premise of this discussion is based on the observation that the Internet has become more and more incorporated into daily life through the embedding of computational capabilities into traditionally non-computing objects. The scope of objects connected to the internet cuts across personal, professional, and societal aspects of life, creating a scenario where online presence is seamless across all these aspects of life. The concept of the Internet of Things (IoT) has emerged as a computing paradigm where physical objects being linked through the internet via varying kinds of connectivity technologies will dominate the computing experience [30].

2.1 Cyber-Physical Systems

Early in the development of embedded computing especially to physical entities, the term Cyber-Physical Systems (CPS) was used to define systems that integrated computation and physical processes. Allowing the ability to expand the capabilities of the physical realm through computation, communication, and control [4]. A typical CPS is considered to have a control unit, sensors, and actuators that are necessary to interact with the physical world, the ability to process data, and a communication interface to exchange data with other systems or the cloud [20]. Such systems are considered to have significant economic potential in various applications for example biomedical and healthcare systems, air transport systems, smart grids & energy, and smart buildings where HVAC systems could be optimized for improved efficiency of a building's resources [4] [25].

Creating solutions where part of the system is embedded in the physical environment inherently carries a significant challenge. This challenge mainly results from *time* and *concurrency* of events in real life, these important computing parameters are often abstracted away through the various computational layers that include the operating system's adaptation layer, middleware, run time environments, up to the application level. The impact of this abstraction is evidenced by the fact that the hardware can record and raise an interrupt at the nanosecond resolution while the operating system will propagate the interrupt in the order of milliseconds [25]. Matters such as standardized abstraction and architectures, validation, security, and reliability remain open areas that require research attention [4].

Meanwhile, IoT use cases such as those implemented in this study can manage through architectural designs that largely separate the sensing aspects from the data processing aspects of the solution although this may not be very optimal for real-time use cases where prompt feedback is expected to the physical world. This way the important part is to record the physical parameter in as real-time form as possible and concurrency can be handled by increasing the hardware sensors and also by software engineering. Internet-enabled products are expected to continue to feature in the next industrial revolution (Industry 4.0)*

Nonetheless, the key challenge of building better integrated CPS systems remains to achieve proper real real-time systems, and to achieve better real-time systems may require re-thinking much of computational abstractions as we know them [25]. The "effective orchestration of software and physical processes requires semantic models that reflect properties of interest in both" [25].

*Industry 4.0 refers the fourth industrial revolution that is characterized by the convergence of IoT, Big Data, Automation, and Artificial Intelligence

2.2 Digital Twin

In more recent literature, topical research initiatives around Industry 4.0 have resurfaced the concept of Digital Twin (DT). It is a concept that is thought to have originated from the aerospace field where early demand for engine and vehicle simulations was important for engine development especially in NASA [32]. Conceptually, there are several definitions of a DT found in the literature but generally, it is a physical and/or virtual machine or computer-based model that is simulating, emulating, mirroring, or "twinning" the life of a physical entity. An initial model of a digital twin was presented in a white paper by Grieve [15], an adaptation of such a model is shown in Fig 2.1 indicating the three elements of a digital twin which entail a physical object, a virtual object and a link for data and information flow between the real and virtual spaces [15] [5].

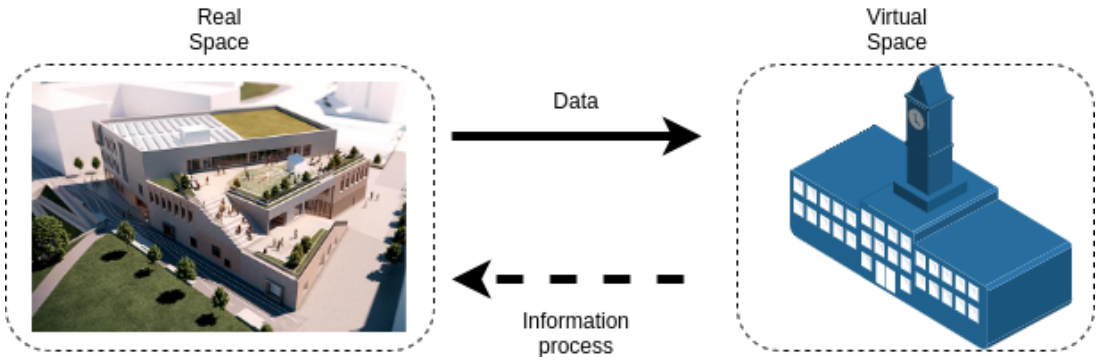

Figure 2.1: A model of a digital twin, adapted from the model introduced in Grieves white paper [15]

Tao et al [43]. consolidate studies around DTs into four groups: (i) DT Modelling, simulation verification validation, and accreditation. Modeling and simulation is the main goal of creating DTs where the physical properties of the entity are modeled based on gathered data. (ii) Data fusion involves data pre-processing, mining, and optimization of data collected from the physical entity and its environment. (iii) Interaction and collaboration, which can be modeled as interactions between physical entities, virtual entities, or physical and virtual entities. (iv) Service, studies in this category demonstrate how services such as monitoring, forecasting, and other health diagnostics of the physical entity can be provided [43].

DTs have found application in the various stages of a product lifecycle covering the product design, production, prognostics, and health management processes. The design and production processes benefit from improved reliability, flexibility, and predictability of the production process. A significant portion of DTs finds application in the prognostics and health management of their physical counterpart [43]. A DT follows the lifecycle of the physical counterpart for purposes of monitoring, control, and optimization of functions. Characteristically, a DT needs to have a seamless connection and continuous data

exchange with the physical entity and due to this connection, the DT constantly receives data from the physical twin [5].

As expected, there is more than one approach to implement a digital representation of a physical entity, this could be imagined as a spectrum of DT implementations, where one end represents DTs with limited integration to the Physical entity and the other end DTs with real-time connection to the physical entity. Can all these be considered DTs? To this, there is an emphasis on the real-time or periodic capability to exchange data between the physical entity and the DT for a model to qualify as a DT. The DT is required to be capable of handling high-dimensional data and therefore the DT requires to be tooled with relevant data analysis techniques and intelligence capabilities [16] [5]. Primarily, applications domains such as manufacturing [9], production [32], aviation, hospital management, and precision medicine [5] have been leading at the adoption of CPS. New concepts such as smart and predictive manufacturing have also been envisioned as potential developments in manufacturing implemented as DT or as parts of a CPS [26] [43].

Smart buildings have emerged as another field where DTs could find applications depending on a building's use case. Enhanced features in areas such as safety, comfort, and convenience can be provided. For example, intelligent greenhouses or data centers are reliant on critical control of temperature and other natural parameters. While the business case for smart (IoT-based) buildings is clear, some technical challenges remain an open area of exploration, in particular, middleware, computational models, fault tolerance, quality of data, and virtual run-time environment [39].

Due to the young nature of IoT development, multiple vendors have unique implementations of middleware which makes issues like scalability, privacy, and/or access control and service management an evolving challenge. Data quality is a significant problem when it comes to implementing intelligence at the Digital Twin, data collected in the physical realm needs to be reliable enough to be used in modeling and action triggers sent back to the physical entity. Lack of data integrity slows down the benefits and increases the risks of using CPSs. Such and more technical challenges remain open areas that require research effort to realize smart buildings [39].

2.3 Sensors and Internet of Things

Advancements in micro-electro-mechanical systems (MEMS) technology has resulted in the development of cheaper, smaller in size, and more powerful sensors that are easily deployed on large scale. Generally, a typical sensor node consists of a sensing unit, simple data processing capability, and communication components that allow the collected data to be transmitted to other sensors in the network, a gateway device, or the end-user application [2].

In large-scale applications, sensors are deployed as a wireless sensor network and the data produced is often big data that requires significant storage and computing resources to analyze. A unit sensor could, for example, be used to measure a specific ambient quantity such as temperature, humidity, mechanical stress levels among other attributes. A greater potential, however, lies in the interconnection of these sensors to provide measurements of multiple attributes, this gives rise to many applications [2] and the notion of the Internet of Things.

The definition of IoT has evolved, first-generation IoT technology referred to Radio Frequency Identification (RFID) technology based solutions. RFID is a technology that works by chips transmitting their identifier to an RFID reader through wireless communication, in this era, "things" were tagged with RFID tags [45] [3]. A typical RFID IoT architecture consists of three layers: a perception layer, a network layer, and the application layer where the perception layer consists of components that collect data (readers and writers), the network layer consists of the transport elements and networking technologies. The service layer is the application layer where collected data is stored and managed for a business-specific implementation. This technology's main applications were found in supply chain management, health care and medicine, military and defense, payment transactions, warehousing, and distribution systems among others [22].

The second generation of IoT technology leverages the Internet Protocol (IP) to connect actual devices to the internet and developments in wireless sensor networks (WSNs). Sensor networks are designed to be fault-tolerant, scalable, low cost to produce, adaptable to various operating environments, and capable of operating with constrained hardware power resources [2]. Research efforts to adapt IP and HTTP technologies to constrained devices continue to take place to provide better integration across IoT technologies through standardization and open development of networking technologies [19].

The third generation of IoT computing mainly incorporates cloud computing among other developments for example growth in social networks and development of local connectivity technologies such as near field communications (NFC) [45]. Cloud computing provides complementary functionalities to IoT due to the provision of storage and computational resources that are quite limited in sensors [23]. Integrating IoT to the cloud to realize the "cloud of things" means that IoT can leverage software delivery models such as software as a service to develop new models -Sensing and Actuation as a Service (SAaaS) [3].

IoT networks can communicate at varying ranges using different connectivity technologies, networks depicted such as Nano Networks, Body Area Networks, Personal Area Networks, Local Area Networks are examples of networks under which IoT devices can operate. Technologies that enable networking at these levels include Bluetooth, Ultra-Wideband, ZigBee, Z-Wave, Wi-Fi, SigFox, LoRa, and LTE. All these technologies have

different properties that render them suitable for specific use cases [28] [27].

Connecting unique heterogeneous sensor nodes requires re-thinking of the existing internet technologies since protocols such as IPv4 were considered not scalable to accommodate such an increase of devices in the address space. IPv6 is however presented as more suitable and capable to adapt to IoT use cases [37]. Given the ad hoc nature of IoT networks, Cirani et al. demonstrate an architecture that would support service and resource discovery that would allow the further machine-to-machine interaction between IoT networks to take place [8].

2.4 REST APIs and Service Oriented Architecture

The Service-Oriented Architecture (SOA) framework provides a high-level approach on how to organize information technology infrastructure to meet given business objectives [11]. The term *service* can have multiple connotations depending on the context or perspective. Commonly, a service can be viewed from a business perspective or a technical perspective [35]. The business perspective considers the value (what) a customer is willing to pay for irrespective of technology (how) and the technology perspective concerns encapsulation of functionality abstracted from the context [35].

A service represents a minimal re-usable component that is loosely coupled from other services, this allows for easy adaption and on-demand provisioning of composite services [11] [42]. Further, there are four characteristics used to describe SOA. First, SOA interfaces are described using WSDL (Web Service Description Language) [7] composed in Extensible Markup Language (XML). Second, an XML schema known as XML Schema Definition (XSD) is used for messaging. Third, a universal description, discovery, and integration (UDDI) based registry that maintains a list of the services provided is required and lastly, a service is required to maintain a level of quality defined by the quality of service requirement [11].

Web-based APIs are considered a more lightweight alternative to web services that are based on WSDL and SOAP. This is largely due to REST's create, read, update and delete interface that makes it easy for clients to make use of the APIs [42]. In this context, REST is considered the communication protocol and JSON as the content format. REST-based Web APIs are a dominant solution for IoT solutions [42].

To further adapt the SOA architecture to an IoT solution, Xu et al [45] proposes a four-layered architecture that includes a sensing layer, a networking layer, a service layer, and an interface layer. In this design the sensing layer is representative of the devices used to sense and collect data, the networking layer addresses connectivity issues among devices and data transfer to backend services.

The service layer is where middleware services such as storage, service discovery,

service composition, service APIs, and trustworthiness establishment are managed. The interface layer provides the mechanism for users to interact with the application and its services, this could be in the form of an application frontend or an application API [45].

In Principal, the REST architectural style is derived from four main constraints: resource identification through URIs, a uniform interface, statelessness, and self-descriptive messages [13]. RESTful web services are considered a practical solution when it comes to connecting heterogeneous devices and provisioning APIs to these devices [40].

A RESTful architectural design also makes it possible to create an application API which is a foundation for building a frontend service on which end users can use to visualize measurements taken by the sensors [18]. Comparatively, REST is best suited for ad hoc integration use cases compared to a traditional web service approach such as Simple Object Access Protocol (SOAP) or Web Service Description Language (WSDL). This is because REST makes use of generic HTTP clients which means clients can be implemented in a variety of programming languages, URIs make it possible to discover web-based resources before a registration process, and the services can be stateless therefore allowing the service to scale up when necessary [33].

Guinard and Trifa [17] explore two ways how a RESTful web-oriented IoT architecture can be implemented. One involves making the device's web addressable, in this design a simple web server runs on the device itself and the device is assigned an IP address, effectively turning the sensor node into a RESTful resource. The challenge with this approach is the fact that most embedded devices are not IP-enabled. The second architectural design provides a RESTful interface through an intermediate gateway that supports IP, in this design, the gateway abstracts the underlying devices thereby abstracting their communication protocols. Such a design has the advantage of managing one web server that connects limited devices to the web in addition to providing a platform to develop more features and logic from the underlying data collected from the sensors.

As a result of the increase in pervasive sensors, cloud computing, and other IoT related technologies, opportunities for big data related applications have emerged. Concepts such as smart and connected communities can be realized as a variety of data can be collected and utilized to support decision making and provide intelligent services to citizens [12] [41].

3. Research Methodology

Data used in this thesis was obtained from a multi-purpose building named Ypsilon located in the city of Turku, Finland. Ypsilon building includes facilities for basic education, a library among other public services. Also, the Ypsilon building commissioned in 2020, is designed as a smart building aiming for better energy efficiency and improved comfort of occupants. The obtained data is collected from multiple sensors installed in the building and it is used to conduct modeling and other analysis in later chapters.

This study follows a design science research methodology, intending to create an application or system that could serve as a proof of concept for a potential IAQ analytics and management platform. The design science framework followed outlines a six-step procedure [34]:

- Problem identification and motivation
- Definition of the objectives
- Design and Development
- Demonstration
- Evaluation and communication

3.1 Problem Identification and Motivation

A comprehensive IoT solution in a smart building setting involves bringing together three components, IoT sensors, Big Data Management, and Analytics into an integrated solution [6]. The data management aspects can be further refined into sub-processes that involve data engineering, data preparation, and data analytics [46]. Currently, only sensors that generate data are installed in the Ypsilon building and the data collected is stored in a raw unprocessed manner. This poses some challenges to building stakeholders, for example, due to lack of analytics capability, building stakeholders do not have the tools to explore the collected data, secondly an expert is required to study the data and make the necessary reports every so often. Further, data extraction and analysis are

manual which means that any new analysis requires the tedious cycle to be repeated. In general, performing any necessary air quality interventions through this manual process is tedious and sub-optimal for an IoT setting.

3.2 Definition of the objectives

The objective of this work is to develop an analytics solution that allows easier exploration of the data and incorporates predictive modeling capabilities to some of the data elements. The data collected by the time of this study was not significantly large in quantity to be considered as big data, however, increasing the number of sensors easily leads to a big data setting. Such a solution would highlight the status of the IAQ and provide a platform to develop other features for example real-time data analysis.

3.3 Design and Development

The resulting artifact is a functional application constituting a front end and a back end. Requirements were inferred from project meetings held with various project members. Developing the system follows a Micro-Service Architecture and the interfaces are implemented as RESTful interfaces. This ensures the system is built on Principals that support reusability and easy maintenance of the application. Machine learning training is conducted outside of the application (offline) however the implemented models are provided as RESTful APIs.

3.4 Demonstration

The developed system is loaded with historical data collected by sensors. This data is provided via a RESTful API to other sub-components of the system such as a front-end client. The resulting system makes data effectively more visible, easier to browse, and easier for users to interact with.

3.5 Evaluation and Communication

For this project, the evaluation and communication took place hand in hand. These sessions were mainly conducted in the scope of stakeholder workshops where presentations about the data analysis results and artifact implementation were presented. Feedback both direct and inferred is considered and included in the improvement of the artifact.

4. Technical Solution

One of the main technical contributions of this project is the realization of the designed artifact as a web-based application. All the elements of this study that involve data, data analysis, and integration through APIs are combined to form the application. The application shows a potential architecture for such a service and further provides a good ground for other stakeholders to review the utility of sensing, storing data, and investing in developing smart buildings. Making use of recorded historical data can be beneficial especially towards the integration of smart HVAC that responds to anticipated occupancy and also improving maintenance practices around buildings. To this effect, a proof of concept system was developed whose architecture is presented in Figure 4.1.

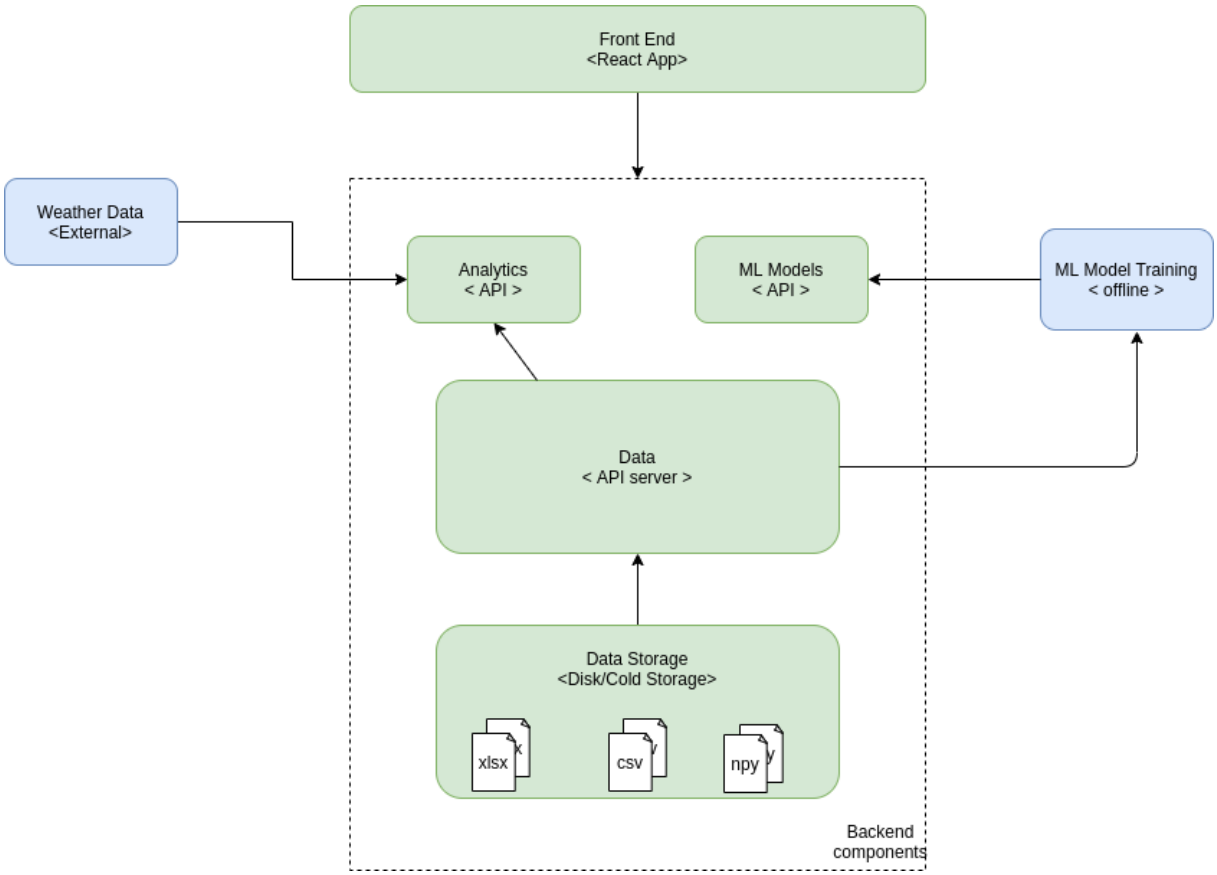

Figure 4.1: A high level architectural view of the data analysis system.

4.1 System Components

This system is designed mainly to facilitate data analysis and insightful presentation. Sensing and data collection aspects were considered to be out of the scope of this project, which means the project begins with a package of files containing various sensor data. Micro-service architecture is considered for the overall design of the system, smaller independent units with their corresponding APIs are integrated through the frontend application. All servers are built using Python's Flask application framework.

4.1.1 Data Storage

Data storage refers to the components used for processing the data and the actual files that hold data in the local disk. Historical data is initially obtained from excel files, which are pre-processed and finally stored in the local hard drive as NumPy(*.npy*) files. Numpy files are provided by Python's NumPy library as a mechanism to store single arbitrary binary arrays to disk in a persistent way [38]. The Numpy file format also stores full information about the array and due to the binary state, these files can be reconstructed across machines easily. They were observed to load faster than other types of files such as CSV or text. Since the data used in the project is time-series data, NumPy files were considered a reasonable intermediate solution. To put things into perspective, the entire data pipeline is envisioned as presented in Figure 4.2, this software component involves only data cleaning and conversion stages. Since the data collected is time series format, it is important to preserve the order of recording observations. Python's pandas library is well designed to manipulate timeseries data and was used in this case to load the stored numpy files.

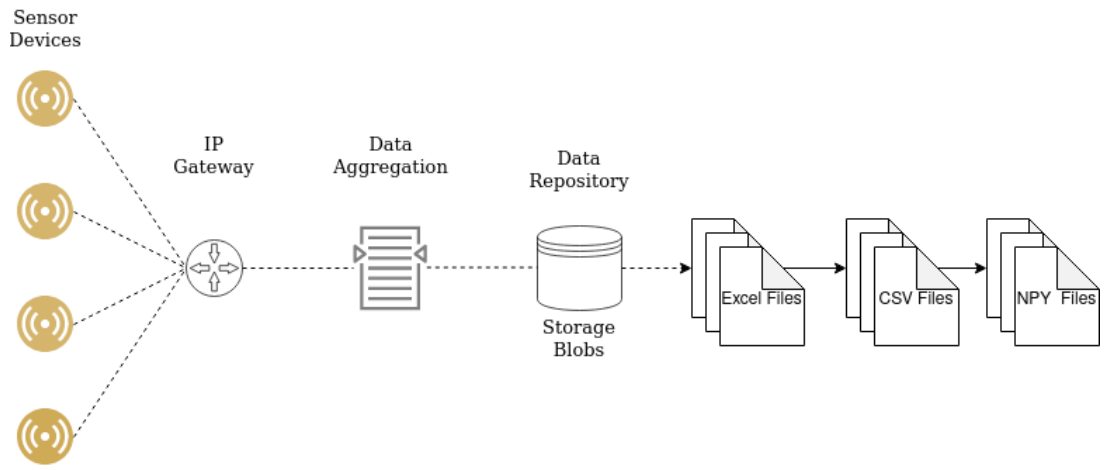

Figure 4.2: Data processing pipeline

Table 4.1: URI structures and their respective endpoints

	URI Structure
Tier 1	http://<site url>/api/v1.0/<endpoint>
Tier 2	http://<site url>/api/v1.0/room/<endpoint>
Tier 3	http://<site url>/api/v1.0/room/<room>?category=<n>&sensor=<n>&freq=<n>&resampler=<method>
	Rest Endpoint
Tier 1 Endpoint	http://localhost:5024/api/v1.0/rooms
Tier 2 Endpoint	http://localhost:5024/api/v1.0/room/c2105
Tier 3 Endpoint	http://localhost:5024/api/v1.0/c2105?category=5&sensor=1&freq=W&resampler=mean

4.1.2 Data API Server

This is a web server designed to provide a low-level API whose main purpose is to provide access to the stored raw data. Additionally, the API design can be considered to have three tiers as shown in Fig 4.4 where the top tier is the root Uniform Resource Locator (URL), the second tier contains a URL to each available rooms and the third tier contains all data Uniform Resource Identifier(URI) for each sensor available in a room.

Such a tiered structure allows potential scaling by the addition of more sensor installations in additional rooms within the building. The structure of URIs at the three tiers and their sample endpoints are presented in Table 4.1. In this design, the root endpoint (Tier 1) returns an object containing the URIs listed in Table 4.2, each of the second tier endpoints, in turn, returns an object with sensor data URIs. In other words, each of these endpoints returns a list of other URIs such as those listed in the Table A.1 in the appendix section. This design ensures the system facilitates the discovery of the system's resources.

Tier three endpoints take query parameters that allow for querying different data types, the parameters are category, sensor, freq, and resampler. Categories and sensors are queried based on their IDs, a category has multiple sensors that are mapped as shown in Table 4.3. For example, category 0 has three sensors (0, 1, 3) where a query on category 0 and sensor ID 0 will return temperature (Lampotila) data. The freq and resampler parameters function together to support different data frequencies and different resampling methods and the supported parameter values are shown in Table 4.3 below.

An example of a successful GET API call to the RESTful data API endpoint is

Table 4.2: A listing of the second tier URLs provided by the API server

Room	Highlevel Endpoints
c2105	http://localhost:5024/api/v1.0/room/c2105
c3032	http://localhost:5024/api/v1.0/room/c3032
c3060	http://localhost:5024/api/v1.0/room/c3060

Table 4.3: Category and sensors parameters mapping

Category	Category Name	Sensors in Category
0	Tuloilma	{'0': 'Lampotila', '1': 'Ilmankosteus', '2': 'Paristojannite'}
1	Sisailman laatu	{'0': 'TVOC', '1': 'Hiilidioksidi', '2': 'Ilmankosteus', '3': 'Ilmanpaine-ero', '4': 'Lampotila', '5': 'Maarakonsentraatio PM0.5', '6': 'Maarakonsentraatio PM1.0', '7': 'Tyypillinen hiukkaskoko', '8': 'Maarakonsentraatio PM2.5', '9': 'Paine', '10': 'Massakonsentraatio PM4.0', '11': 'Massakonsentraatio PM2.5', '12': 'Maarakonsentraatio PM10.0', '13': 'Massakonsentraatio PM1.0', '14': 'Massakonsentraatio PM10.0', '15': 'Maarakonsentraatio PM4.0'}
2	Poistoilma	{'0': 'Ilmankosteus', '1': 'Lampotila', '2': 'Paristojannite', '3': 'Signaalin voimakkuus, vastaanotto', '4': 'Signaalin voimakkuus, lahetys'}
3	Tuloilman tilavuusvirta (l/s)	{'0': 'Ilmavirran tilavuus litroina', '1': 'Ilmanpaine-ero', '2': 'Signaalin voimakkuus, vastaanotto'}
4	Poistoilman tilavuusvirta (l/s)	{'0': 'Ilmavirran tilavuus litroina', '1': 'Ilmanpaine-ero', '2': 'Signaalin voimakkuus, vastaanotto'}
5	Henkilomaara	{'0': 'Lukumaara sisaan', '1': 'Kokonaismaara', '2': 'Lukumaara ulos', '3': 'Paristojannite' }
freq		{'H': 'Hourly', '6H': '6 hours', '12H':'12 hours', 'D': 'Daily', 'W': 'weekly' 'M':'Monthly', 'BH': 'Business hours', 'Business Days':'B' }
resampler		{'mean': 'aggregate by average', 'sum': 'aggregate by sum' }

shown in the listing below where the resulting data contains category, room, sensor, timestamp, unit, and value information. The API's client can then manipulate this data, in this architecture such a client is the analytics server that fetches this data and computes statistical parameters of a given data set.


```
[
   {
      "category":"Poistoilma", "room":"c2105",
      "sensor":"Ilmankosteus", "timestamp":"2019-09-30T00:00:00.000Z",
      "unit":"%", "value":45.5115482638
   },
   ...
   {
      "category":"Poistoilma", "room":"c2105",
      "sensor":"Ilmankosteus", "timestamp":"2020-03-31T00:00:00.000Z",
      "unit":"%", "value":31.3205847491
   }
]
```


Performance of the API is dependent on the frequency of fetched data since frequency informs the size of the payload -data request with hourly resolution(freq=H) from an endpoint leads to a larger payload compared to a monthly resolution that will result in a

smaller payload. Server response time can be considered to have two components, waiting time and receiving time. Waiting time can infer performance as a function of the internal implementation of the API. In this implementation, the actual data is stored in minutes resolution and all other frequencies are generated on runtime after an API call is received. This design could have an impact on the APIs performance. An alternative design would be to store data in form of the supported frequencies, however, this could result in higher storage requirements but provide a better experience to calling clients.

Figure 4.3 shows these differences when one endpoint was tested over a tunneled connection to simulate an arbitrary network API call. The subgraph titled transfer size vs freq parameter shows results of calling an API endpoint with varying values of the freq parameter. The observed transfer sizes for hourly, daily, weekly, and monthly are 915.9kB, 3.9kB, 5,98kB, and 1,36kB respectively.

The second graph shows the API server's latency (waiting time) given a supported freq parameter value. As expected, the server takes longer to respond with Hourly data, since these are more observations to compute. The values obtained on this test are 2237ms, 520ms, 485ms, and 509ms when the freq parameter is set to hourly, daily, weekly, and monthly respectively. Equally observed but not reported were latency to receive the payload (receiving times) -when data is large(hourly), the client will take longer in receiving the data and therefore increase the API's overall latency. In practice, the trade-offs are about storing various resolutions of the data and their incurring storage costs, or compute various resolutions on run-time and incur performance and computational costs.

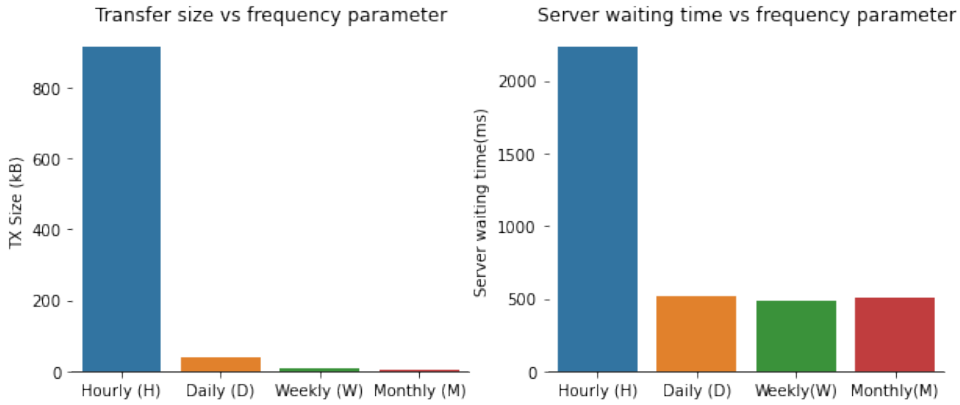

Figure 4.3: Performance of the API Server relative to data size and server waiting times based on freq parameter.

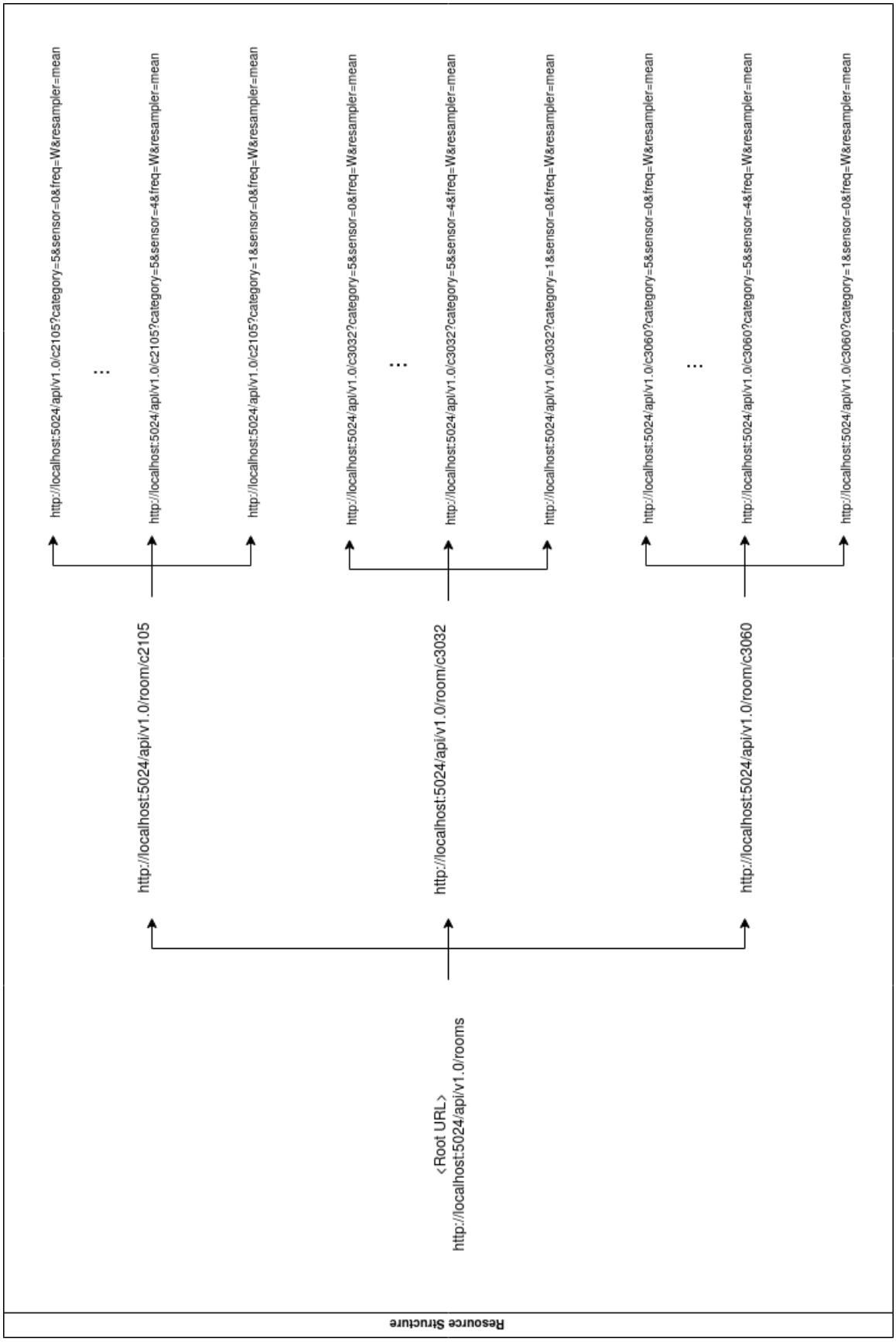

Figure 4.4: Three tier resource structure encapsulated by the API server.

4.2 Analytics API Server

The Analytics API Web server is a server that facilitates the generation of statistical metrics of the underlying data. Statistical algorithms used to generate relevant metrics are run from this server and results are stored in memory within the server. An example of an API call to this server and the results are shown below.

```
curl http://localhost:5034/statsapi/v1.0/c3060?category=0
```


```
[
  {
    "index": "Lampotila",
    "max": 20.0,
    "mean": 20.0,
    "min": 19.0,
    "std": 0.0
  },
  {
    "index": "Ilmankosteus",
    "max": 53.0,
    "mean": 33.0,
    "min": 18.0,
    "std": 6.0
  }
]
```


4.3 ML Models

The ML Models component is a dedicated server that provides APIs to forecasted data produced from machine learning models that are extensively discussed in the next chapter. The server can also store machine learning models after they have been trained. Further, issues such as the versioning of the models can also be handled in this component. Model training and tuning are considered to be outside the scope of this component but generally, the output of such modeling pipelines can be stored in this component and provided as an API accessible to other components.

4.4 Weather Data

The weather data component symbolizes an external third-party API that is used to get outdoor weather data for example humidity. Such data is applied in the visualization aspects of the analytics or used in supporting internal modeling of tracking

4.5 ML Model Training

This component represents the tools and pipeline used to train machine learning models, it is considered to be out of the scope of the application but it makes use of the Data API server to fetch machine learning input data. Model training and tuning is resource-intensive and therefore considered to be an independent process. Also, model development was conducted in Python's scientific computing environment using all the relevant libraries for machine learning such as Sklearn, statsmodels among others. Models were mainly trained locally on Jupyter notebooks, a more comprehensive discussion on the training and tuning strategies applied is discussed in later chapters.

4.6 Front End

This is mainly a visualization application in particular build on React technology. React was considered particularly useful in this context as graphs could be developed as independent components and composed in the final interface. This application is the main tool users can use to explore stored data to derive insights about their indoor environment.

Figure 4.5: A screenshot of the application's front end

5. Data and Machine Learning Models

5.1 Data Exploration

Measurement data was obtained from sensor modules installed in three distinct rooms in the Ypsilon building, the rooms are labeled $c2105$, $c3032$, $c3060$ with floor area $100m^2$, $38m^2$ and $65m^2$ respectively. Each room contained multiple sensors in a setup as depicted in Fig 5.1. Data from the sensors in a room is aggregated and stored locally as file blobs. Such file blobs were collected from the sensor devices across the rooms.

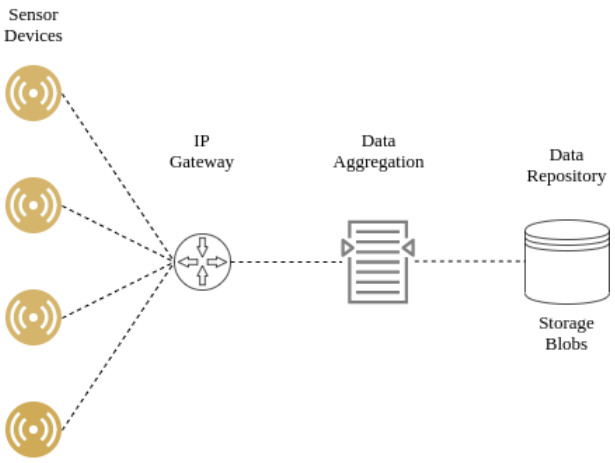

Figure 5.1: Sensor Data Collection Architecture

A sample of an aggregated data blob is shown in Table 5.1 from one room. Sensors record measurements at varying intervals, for example, the temperature sensor (lampotila) measurements are taken at the minute resolution while pressure sensor (Ilmanpaine-ero) measurements are entered every two minutes. A comprehensive list of the sensors in a room is indicated in the Appendix section, Table A.2. Only a subset of the sensors and data that related to thermal comfort are used in the data modeling exercise, the entire data set is however used to provision a data API as shown in the implementation section 4.

Table 5.1: Snapshot of original aggregated blob

Aika	Anturi	Sensorityyppi	Yksikko	Arvo
30.11.2019 23:59:00	Poistoilman tilavuusvirta (l/s)	Ilmanpaine-ero	Pa	-7.2
30.11.2019 23:59:00	Poistoilman tilavuusvirta (l/s)	Ilmavirran tilavuus litroina	slm	156.7
30.11.2019 23:57:00	Tuloilman tilavuusvirta (l/s)	Ilmanpaine-ero	Pa	3.3
30.11.2019 23:57:00	Tuloilman tilavuusvirta (l/s)	Ilmavirran tilavuus litroina	slm	73.7
30.11.2019 23:56:00	Poistoilma	Ilmankosteus	%	24.1
30.11.2019 23:56:00	Poistoilma	Lampotila	°C	19.3
30.11.2019 23:55:00	Henkilomaara	Kokonaismaara	pcs	0
30.11.2019 23:55:00	Henkilomaara	Lukumaara sisaan	pcs	0
30.11.2019 23:55:00	Henkilomaara	Lukumaara ulos	pcs	0
30.11.2019 23:55:00	Sisailman laatu	Hiilidioksidi	ppm	371
30.11.2019 23:55:00	Sisailman laatu	Ilmankosteus	%	21.7
30.11.2019 23:55:00	Sisailman laatu	Ilmanpaine-ero	Pa	-0.4
30.11.2019 23:55:00	Sisailman laatu	Lampotila	°C	18.7
30.11.2019 23:55:00	Sisailman laatu	Maarakonsentraatio PM0.5	pcs/cm^3	2.7
30.11.2019 23:55:00	Sisailman laatu	Maarakonsentraatio PM1.0	pcs/cm^3	3

5.2 Modeling thermal comfort and indoor air quality

A common noble objective of modern buildings is to provide a good indoor environment to occupants. Although HVAC systems can be used to control indoor climate, achieving a good indoor environment requires embedding the relevant Principals across the various stages of a building's life cycle: design, construction, and use of a building. One objective of exploring data collected from the Ypsilon building was to investigate the state of indoor climate and factors influencing indoor climate variables in the building.

In Finland, the Ministry of Social Affairs & Health together with the Ministry of Environment set the relevant quality requirements and regulatory metrics for indoor environments. Such requirements have been adapted to The classification of indoor environment 2018 guideline which is a guideline generated by industry stakeholders [1]. The guideline provides a baseline for various indoor climate measurements referenced in this thesis. In particular, the guideline outlines indoor environment classifications, target values for thermal conditions, indoor air quality, acoustic environment, airtightness of the building envelope, and ventilation [1].

According to the guideline, there are three categories of occupant satisfaction levels denoted as S1, S2, and S3 [1]. S1 category refers to an individual indoor environment where thermal conditions are comfortable with no detectable odors, draught, or overheating. In this category, the user may individually control thermal conditions. S2 category requires good indoor air quality and thermal environment, no draught but overheating is possible especially during the summer season. S3 category is known as a satisfactory

indoor environment, in this category, indoor air quality and thermal conditions meet the minimum requirements stipulated in the building code [1]. A building's indoor air climate can therefore be evaluated against these classification categories.

Parameters used to investigate thermal comfort are generally categorized into personal and ambient parameters. Personal parameters are represented by the individual characteristics of the occupant such as age, metabolic rate, clothing insulation among others, such parameters are out of the scope of this thesis. Instead, the focus is on ambient parameters that include temperature, air velocity, relative humidity [10] [44]. The concentration of carbon dioxide is used as a proxy for indoor air quality [24]. To this effect, *temperature*, CO_2, *number of people* in a room, and *ventilation rates* were selected as relevant variables for analysis.

Summary statistics for the selected variables are shown in Table 5.2. Each subtable represents summary statistics relevant to one parameter across three different rooms. The presented summary statistics are generated based on daily frequency although alternative frequencies such as half-hourly, hourly, weekly can also be computed for different modeling purposes. Before generating summary statistics, the data was adjusted for outliers which are visible from the box plot shown in Fig 5.2. An alternative way to visualize characteristics of the data is shown through the time-series plots in Fig 5.3, an example of an outlier entry is indicated in the ventilation rates subplot with a circle marker in addition to missing data points shown in the temperature subplot with an arrow.

Since the Ypsilon building primarily serves as a school facility, non-business days have been omitted from the data to focus on modeling indoor air climate during periods of occupancy. Temperature is reported in degrees Celsius (°C), Carbon dioxide in parts per million (ppm), ventilation rates in liters per second (l/s), and the average total number of people in a room represent the average number of people in a room per day.

Table 5.2: Summary statistics for indoor room temperature, carbon dioxide, ventilation rates and Average total number of people in a room.

	Indoor Temperature		
	c2105	**c3032**	**c3060**
count	102	104	105
mean	19.1	20.9	21.0
std	0.2	0.2	0.2
min	18.7	20.4	20.5
25%	19.1	20.7	20.9
50%	19.2	20.9	21.1
75%	19.3	21.1	21.2
max	19.6	21.5	21.5

	Carbon dioxide		
	c2105	**c3032**	**c3060**
count	104	97	101
mean	407.5	449.8	447.0
std	22.2	11.8	28.3
min	363.4	417.3	386.3
25%	393.1	445.2	435.7
50%	408.0	451.7	452.1
75%	421.2	456.6	465.2
max	461.7	475.7	498.5

	Ventilation rates		
	c2105	**c3032**	**c3060**
count	68	65	65
mean	74.7	65.3	54.8
std	1.4	0.3	0.5
min	72.5	64.3	53.4
25%	73.7	65.0	54.5
50%	74.5	65.3	54.9
75%	75.9	65.5	55.3
max	77.5	66.0	55.9

	Avg of Total No. of People/day		
	c2105	**c3032**	**c3060**
count	98	92	98
mean	7.8	21.9	2.6
std	9.3	13.7	3.2
min	0.0	0.0	0.0
25%	2.9	14.3	1.3
50%	5.1	21.7	2.1
75%	8.3	29.6	2.7
max	47.6	55.5	23.2

The average temperatures across rooms c2105, c3032, and c3060 are 19.1°C, 20.9°C, 21.0°C respectively, each of the rooms' standard deviation of 0.2 indicates minimal variance. This is consistent with a managed indoor temperature through air conditioning. According to the previously introduced guideline, S1 and S2 categories of indoor climate recommend a minimum temperature that lies above 20°C, a maximum that does not exceed 23°C and an operative temperature of 21.5°C [1]. Based on these guidelines, room c2105 would require intervention to increase the minimum temperature so that it would be above 20°C.

Daily carbon dioxide levels are on average 407ppm, 450ppm, and 446 ppm in rooms c2105, c3032, and c3060 respectively. According to the guideline, buildings aiming for categories S1 and S2 should maintain carbon dioxide levels that are not in more than 350

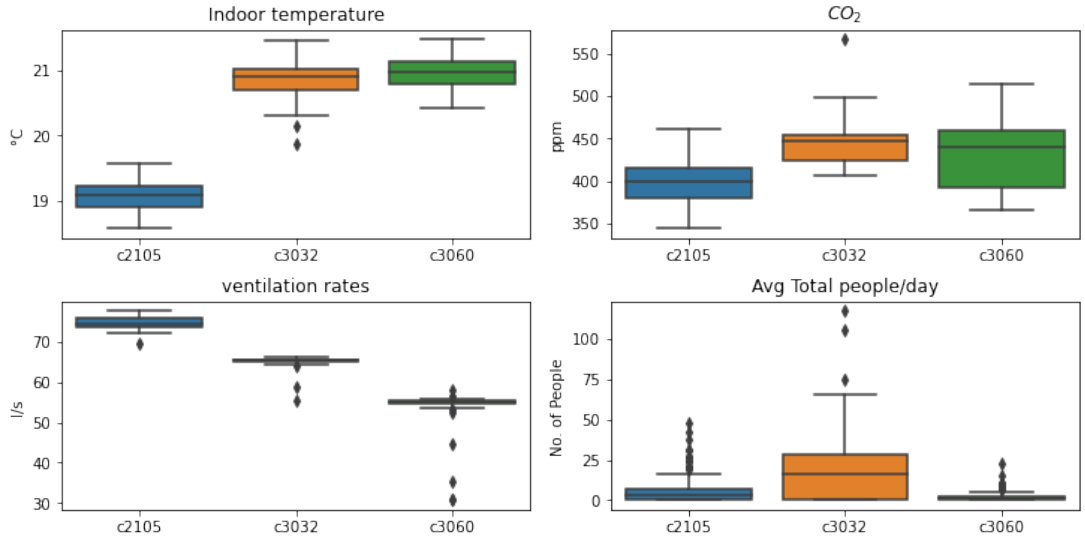

Figure 5.2: Box plots for room temperature, CO_2, Ventilation rates and average of total people in a room per day.

and 550 ppm compared to the Carbon dioxide concentration found in the outdoor air [1]. Carbon dioxide's standard deviation also shows actively managed indoor air quality such that carbon dioxide does not accumulate excessively.

Ventilation rate refers to the inflow of outdoor air measured in liters per second. On average, air flows in at 75, 65 and 65 litres/second in rooms c2105, c3032 and c3060 respectively. Recommendations from the guideline on ventilation indicate that a classroom of S1 category should be ventilated at 5.5 dm^3/s per m^2 and in S2 category 4.0 dm^3/s [1]. The daily average rate of ventilation shown in Table 5.2 indicates ventilation rates 0.75 dm^3/s per m^2, 1.72 dm^3/s per m^2 and 0.84 dm^3/s per m^2 for classes c2105, c3032 and c3060 respectively. These ventilation rates are lower compared to the levels recommended in the guideline.

The guideline indicates that a building should have an average minimum ventilation rate of $0.15 \ldots 0.2$ dm^3/s per m^2 when the building is not occupied. Running ventilation during periods when a building is not occupied works to remove impurities that may come from the building, further running the ventilation system at the normal level for two hours before occupancy is recommended [1]. To summarise all the comparisons between observations and guidelines proposals, the values have been consolidated in Table 5.3.

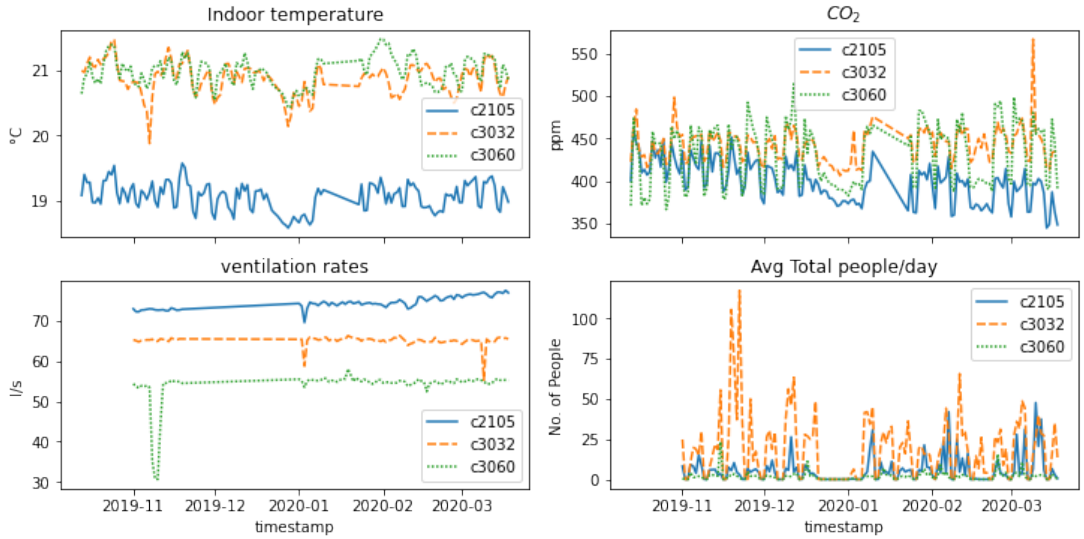

Figure 5.3: Timeseries of room temperature, CO_2, Ventilation rates and average of total people in a room per day.

Table 5.3: Summary of comparison between average values and values recommended from the Finnish classification of indoor climate 2018.

	Indoor Temperature				Carbon dioxide		
	c2105	**c3032**	**c3060**		**c2105**	**c3032**	**c3060**
Classroom size (m2)	100	38	65	**Classroom size (m2)**	100	38	65
Classroom mean (°C)	19.1	20.9	21.0	**Classroom mean (ppm)**	407.5	449.8	447.0
S1 & S1 min (°C)	20	20	20	**S1 (ppm)**	< 350	< 350	< 350
S1 & S2 max (°C)	23	23	23	**S2 (ppm)**	< 550	< 550	< 550

	Ventilation rates				Avg of Total No. of //day		
	c2105	**c3032**	**c3060**		**c2105**	**c3032**	**c3060**
Classroom size (m2)	100	38	65	**Classroom size (m2)**	100	38	65
Classroom mean (dm3/s per m2)	0.75	1.72	0.84	**Classroom mean (people/m2)**	0.16	1.46	0.07
S1 (dm3/s per m2)	5.5	5.5	5.5	**S1 (people per m2)**	2	2	2
S2 (dm3/s per m2)	4	4	4				

To compare the effect of occupancy and accumulation of CO_2, scatter plots presented in Fig 5.4 were plotted. In classrooms c2105 and c3060, there is an observable trend where CO_2 increases with the number of people in the room. Which unlike the effect observed in room c3032 where increasing the number of occupants does not seem to increase CO_2 levels. This observation is consistent with the summary statistics presented in Table 5.3 that indicate room c3032 has the highest ventilation rate per m^2 of floor space.

Similarly, the effect of occupancy on the temperature is shown in Fig 5.5. A general increase in temperature in rooms c2105 and c3060 can be noted as the number of occupants is increased. On the contrary, the room c3032 scatter plot shows that temperature tends to stay consistently between 20 and 21°C irrespective of the level of occupancy. This

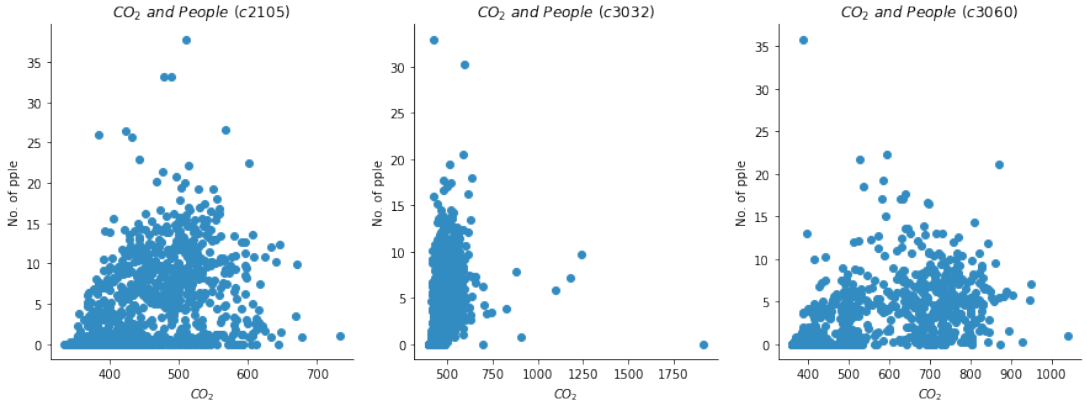

Figure 5.4: Comparing occupancy and accumulation of CO_2 across rooms.

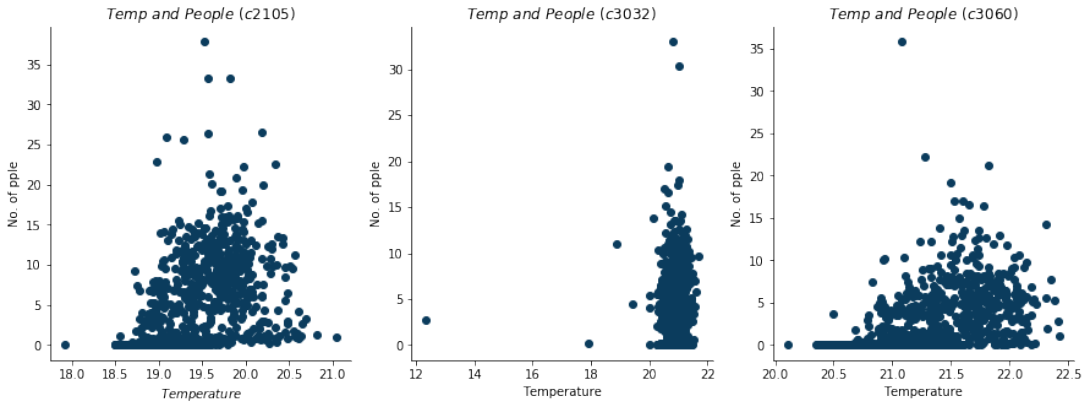

Figure 5.5: Comparing occupancy and temperature across rooms.

is also consistent with a high rate of ventilation compared to the other rooms. While increasing the number of occupants in a room would naturally lead to an increase in CO_2 and temperature in a given room, the presence of ventilation ensures this increase is only moderate and the linear relationship does not hold beyond a certain point.

5.3 Predicting elements of thermal comfort and indoor air quality

To get a holistic view of the indoor environment, all the data collected from the indoor environment (Category Sisailman laatu shown in A.2 in the Appendix section) is checked for their relevance in predicting the main variables of interest. A correlation map as shown in Fig 5.6 indicates that particle related data* tends to be highly and positively corre-

*Maarakonsentraatio PM0.5, Maarakonsentraatio PM1.0, Maarakonsentraatio PM2.5, Massakonsentraatio PM4.0, Massakonsentraatio PM2.5, Maarakonsentraatio PM10.0, Massakonsentraatio PM1.0, Massakonsentraatio PM10.0, Maarakonsentraatio PM4.0

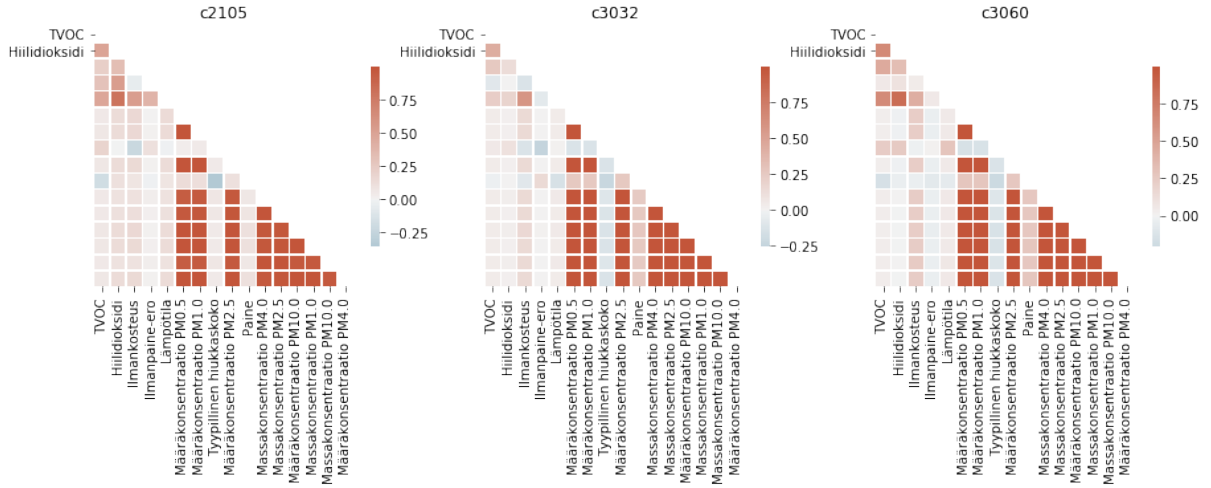

Figure 5.6: Correlation between all indoor sensor data

lated (dark red) unlike the case among non-particle measurements (TVOC, Hiilidioksidi, Ilmankosteus, Ilmanpaine-ero, Lampotila, Paine). This means it is potentially possible to combine particle-related measurements into one variable if need. The low correlations among non-particle data also prove that the variables are naturally independent of each other.

5.3.1 Modeling CO_2 and Temperature using indoor climate variables

Similar conclusions can be obtained by conducting a *Principal Component Analysis* (PCA) on the same indoor dataset. Such PCA results are presented in Fig 5.7. The first row of the graph is a plot of the variance explained by considering only Principal components for variables that represent data related to particles. Across all rooms, the plots show that the first Principal component is enough to explain 98.7% of the variance across the entire group, which is consistent with the observations that they are highly correlated.

The second row of the graphs shows the variance explained when considering only non-particle variables. In this case, it requires all the six Principal components to explain the variability in the data which supports the notion that the variables are largely independent of each other, a significant part of the variance(information) would be lost in an attempt to perform dimensionality reduction across those variables. Practically, the variance explained by the fifth Principal components are 97.3%3, 93.5%, 97.7% across data c2105, c3032, and c3060 respectively.

The third row of the graph shows the variance explained by the Principal components when all indoor variables are included. In this setting, the sixth Principal components can explain 96.2%, 94.%4, 96.8% in across c2105, c3032, and c3060 respectively as seen in

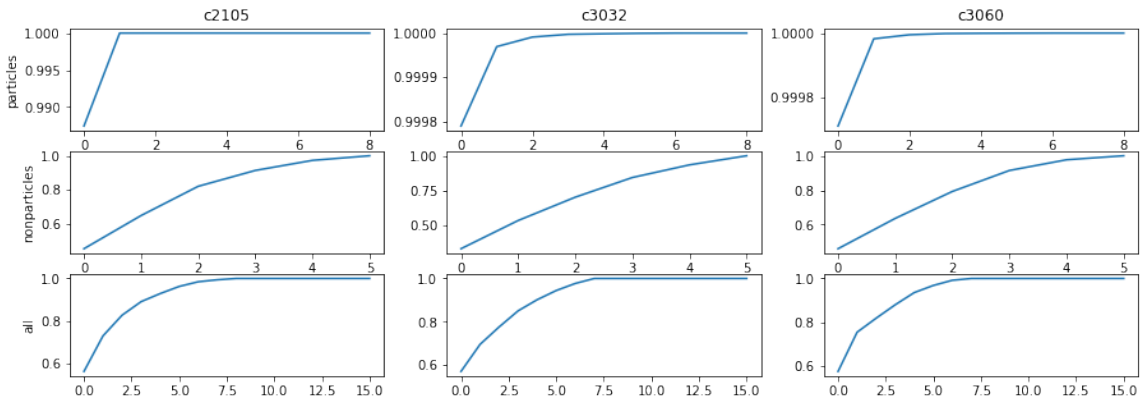

Figure 5.7: Variance explained by the Principal components across particle related variables, non-particle variables and all the indoor climate variables

the graphs. Such a result implies that it is possible to reduce the dimension of the indoor climate variables from 16 to 5 and still capture the variance of the significant source.

To predict CO_2 and temperature levels, several approaches are used. The first approach is to use the Principal components obtained from all indoor climate variables (Sisailman laatu) to conduct a Principal component regression (PCR) on CO_2 and Temperature. Models of the form shown in Eq. 5.1 and Eq. 5.2 are estimated for CO_2 and temperature respectively with data from each room independently. $\theta_0, \theta_1, \ldots, \theta_M$ indicate coefficients of the model, and $Z_1, \ldots, Z_M$ represents M the Principal components being used to estimate the model.

$$CO_{2i} = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, \qquad i = 1, \ldots, n \tag{5.1}$$

$$Temp_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, \qquad i = 1, \ldots, n \tag{5.2}$$

A summary of results from this experiment is presented in Fig 5.8 where the top row of the graph matrix represents results from CO_2 models and the lower row represents results from temperature models in each of the corresponding rooms indicated. In each of the graphs, the y-axis represents the cross-validation root mean square error (RMSE) plotted against the number of Principal components used to train a model. Obtained RMSE values represent the mean of repeated cross-validation RMSE following 10-fold cross-validations repeated 10 times, this is done to improve stability of results.

For the CO_2 model, the lowest cross-validation error occurs when 6, 4, and 9 Principal components are used in the regressions for rooms c2105, c3021, and c3060 respectively, this corresponds to 31.10, 48.94 and 62.00 error values and R^2 metrics in the order of 0.71,

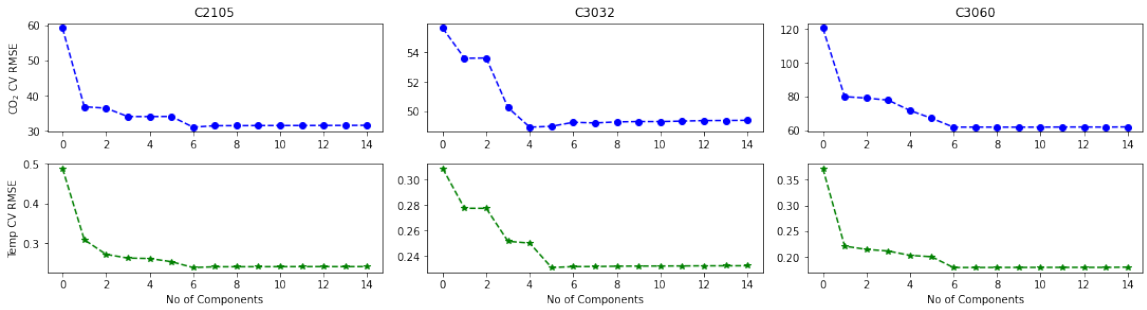

Figure 5.8: Principal Component Regression on CO_2 and Temperature

0.21 and 0.72. Intuitively, based on the results, it is possible to benefit from dimensionality reduction across the models although the benefits are varied, a summary of these results is also presented in tabular form in Table 5.4. For example, the best CO_2 model on room c3060 requires 9 Principal components while a similar model on room c2105 can provide better results with a lower number of components. It is also notable that the model fitted in room c3060 provides a fit closely similar to c2105 but the error metrics are distinctively different -c2105 performs better from an based on the error metrics. Fitting temperature models produced results presented in the second row of Fig 5.8 which are also summarized in Table 5.4. Models with 6, 5, and 6 Principal components produced the best results also indicating the potential benefit of reducing the feature space from 15 dimensions. The best temperature model is obtained from c3060 based on the lowest errors and highest R^2 metric. c3032 appears to perform relatively poorly in both CO_2 and temperature models. Comparatively, fitting the Principal components on temperature data produced slightly better results than fitting the same on data CO_2, probably CO_2 has better predictive power over CO_2 compared to the alternative regression.

Table 5.4: Summary of Principal Component Regressions on CO_2 and Temperature

	CO_2				Temperature		
	Components	RMSE	R^2		Components	RMSE	R^2
c2105	6	31.10	0.71	c2105	6	0.24	0.75
c3032	4	48.94	0.21	c3032	5	0.23	0.47
c3060	6	62.02	0.72	c3060	6	0.18	0.76

5.3.2 Feature Selection by forward selection

The second approach applied is to determine any relationship between CO_2 and Temperature to other indoor parameters of indoor climate (Sisailman laatu), this is done

based on forward selection. This process is conducted by applying a forward selection approach adding features to the models until the final model contains all the features. Each of the models is fitted in Repeated(10x) cross-validation (10-fold). The obtained results are presented in Fig 5.9, the top row of the graph matrix are plots of the errors (RMSE) against the number of features for the CO_2 models while the lower row presents results from temperature models. Accompanying metrics are summarised in Table 5.5 which shows results from both selected CO_2 and Temperature models.

All CO_2 models have a drastic improvement in the reported error after a given feature is added to the model. At some point, the model does not appear to significantly improve following additional variables. In this regard, CO_2 models with 3, 6, and 2 features record the best results with low error values 21.7, 25.7, and 11.09 for rooms c2105, c3032, and c3060 respectively. Notably, all rooms approach their optimal CO_2 models in different ways as seen by their different curve structures. Corresponding R^2 achieved from these CO_2 models are also reported along with their RMSE values in Table 5.5. The best set of CO_2 models across the rooms are summarised in Equation 5.3 where the features are mapped as X1, X2, X3, X4, X5, X6 to represent variables TVOC, Ilmankosteus, Ilmanpaine-ero, Lampotila, Maarakonsentraatio PM0.5 and Maarakonsentraatio PM1.0 respectively.

$$
\begin{aligned}
CO_{2c2105i} &= \beta_0 + \beta_1 X1_i + \beta_2 X2_i + \beta_3 X3_i + \epsilon_i \qquad i = 1, \ldots, n \\
CO_{2c3032i} &= \beta_0 + \beta_1 X1_i + \beta_2 X2_i + \beta_3 X3_i \ + \beta_4 X4_i + \beta_5 X5_i + \beta_6 X6 + \epsilon_i \\
CO_{2c3060i} &= \beta_0 + \beta_1 X1_i + \beta_2 X2_i + \epsilon_i
\end{aligned}
\tag{5.3}
$$

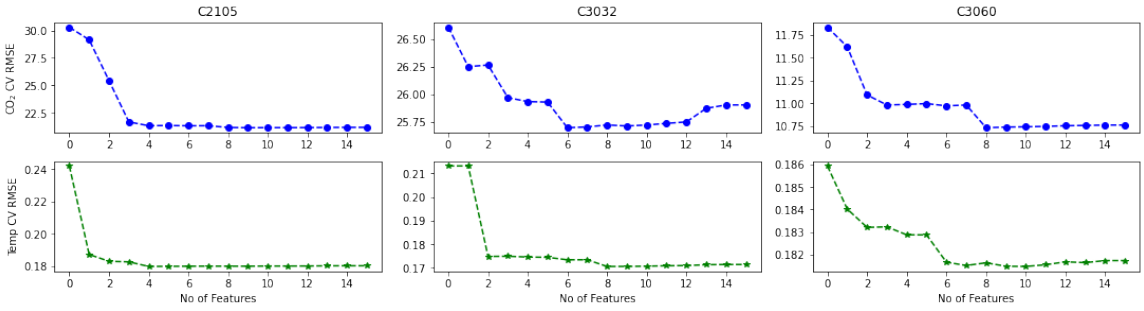

Figure 5.9: Fitting CO_2 and Temperature models with forward selection

Results from temperature models also show a distinct impact from selecting models with a subset of the features. Figure 5.9 bottom row graphs show error curves from repeated cross-validations regressions on temperature and the added features are also selected from the set of indoor climate variables (Sisailman laatu). Temperature models made up of 1, 2, and 2 features are observed to produce the best results for rooms c2105,

c3032, and c3060 respectively. These models are presented in Equation 5.4

$$\begin{aligned} Temp_{c2105i} &= \beta_0 + \beta_1 X1_i + \epsilon_i \qquad i = 1, \ldots, n \\ Temp_{c3032i} &= \beta_0 + \beta_1 X1_i + \beta_2 X2_i + \epsilon_i \\ Temp_{c3060i} &= \beta_0 + \beta_1 X1_i + \beta_2 X2_i + \epsilon_i \end{aligned} \tag{5.4}$$

where the features are mapped as X1 and X2 represent variables TVOC, Ilmankosteus respectively. Corresponding RMSE and R^2 values are presented in Table 5.5, the models produce generally low errors and their fit for the data is presented in the R^2 column. As indicated by R^2 values 0.58, 0.50 and 0.11, models fitted for c2105, c3032 appear to fit the data better compared to the c3060 model.

To conclude this forward selection approach, Table 5.6 presents a summary of the coefficients and their corresponding p-values obtained when fitting the optimal models discussed above. R^2 values reported in this table are slightly different from those presented in Table 5.5 due to cross-validation approach. A key observation from this table is that some coefficients are highly significant (considering 1% significance level) for example the c2105 CO_2 model, while other coefficients are not significant such as those that appear in c3032's CO_2 model where the constant term(0.26), Ilmanpaine-ero(0.11), Maarakonsentraatio PM0.5(0.91) and Maarakonsentraatio PM1.0(0.92) are all insignificant at the 10% significance level. Coefficients for the temperature models are almost all significant at the 1% level save for one(0.01) that is still significant at the 5% level. By looking at the skewness and kurtosis statistics, it can be concluded that the models do make a considerable effort at fitting the data given that the expected skewness and kurtosis for a normal distribution should be 0 and 3 respectively. However, the adjusted R^2 shows that the model would benefit from additional features to improve its predictive ability.

Table 5.5: Summary of RMSE and R^2 results from fitting CO_2 and Temperature using

CO_2				Temperature			
	Components	RMSE	R^2		Components	RMSE	R^2
c2105	3	21.66	0.55	c2105	1	0.19	0.58
c3032	6	25.70	0.23	c3032	2	0.17	0.50
c3060	2	11.09	0.16	c3060	2	0.18	0.11

Table 5.6: Regression results from selected CO_2 and Temperature models

	CO_2 regression models					
	c2105		c3032		c3060	
	coef	pvalue	coef	pvalue	coef	pvalue
const	-804.5	0.00	71.9	0.26	412.1	0.00
TVOC	0.0	0.00	0.2	0.00	0.1	0.00
Ilmankosteus	0.6	0.00	-1.3	0.00	-0.5	0.00
Ilmanpaine-ero	19.7	0.00	5.1	0.11	10.5	0.00
Lampotila	63.1	0.00	16.8	0.00		
Maarakonsentraatio PM0.5			-4.8	0.91		
Maarakonsentraatio PM1.0			3.8	0.92		
$R^2 - Adjusted$	0.57		0.23		0.16	
Skewness	0.41		1.18		1.18	
Kurtosis	4.36		4.49		6.02	

	Temperature regression models					
	c2105		c3032		c3060	
	coef	pvalue	coef	pvalue	coef	pvalue
const	16.8	0.00	19.6	0.00	19.7	0.00
TVOC	0.0017	0.00	0.0002	0.01	0.0014	0.00
Ilmankosteus	0.6426	0.00	0.0300	0.00	0.0044	0.00
$R^2 - Adjusted$	0.52		0.51		0.14	
Skewness	1.24		0.07		0.22	
Kurtosis	5.77		2.99		2.89	

5.3.3 Modeling CO_2 and Temperature using boosted regression trees

Using regression trees provides control against mostly parametric models previously applied, given the non-linear structure of the data, this approach may provide better prediction accuracy. In this study boosted regression trees are used. Before fitting regression trees, a parameter tuning process is set up to identify a set of parameters that would enhance the regression tree's performance. To this effect, a parameter grid shown in Ta-

Table 5.7: Parameter Space

Parameter Grid				
criterion	mse	friedman_mse	-	-
max_depth	3	5	10	-
max_features	auto	sqrt	log2	None

Table 5.8: Parameter search results.

	c2105		c3032		c3060	
	CO_2		CO_2		CO_2	
	Grid Search	Random Search	Grid Search	Random Search	Grid Search	Random Search
criterion	mse	mse	mse	friedman_mse	mse	friedman_mse
max_depth	5	5	3	3	3	3
max_features	auto	auto	auto	None	auto	None
	Temperature		Temperature		Temperature	
	Grid Search	Random Search	Grid Search	Random Search	Grid Search	Random Search
criterion	mse	friedman_mse	friedman_mse	friedman_mse	mse	mse
max_depth	3	3	5	5	3	3
max_features	auto	auto	auto	None	sqrt	log2

ble 5.7 is applied to determine the best set of parameter combinations. The parameter space is searched using both *grid search* and *random search* approaches.

An estimator tree is set to different parameters from the parameter space and by using both grid search and random search cross-validation, the parameter combination set that yields the best results (lowest Mean Squared Error (MSE)) is considered to produce the best decision tree. Each of the parameter searching algorithms produces the best tree and these trees are compared for their performance. Results from the tuning process are shown in Table 5.8.

Fitting regression trees based on the tuning process shows that the best of grid search and the best of random search chosen parameters results in models whose performance (RMSE) is indistinguishable. Results for this process are shown in Fig 5.10. The plots indicate the value of RMSE obtained(y-axis) and modeling fitting iterations(x-axis), all m1 models are color-coded blue and they are based on the best possible parameters obtained via grid search while m2 based models are colored orange and they represent the models estimated based on parameters selected by random search. Due to the similarity of the obtained curves, the blue curves are obstructed by the orange curves in almost all the plots. Also, due to a fixed *seed*, an algorithm with the same parameters will return similar numerical results hence the exactness of the two curves.

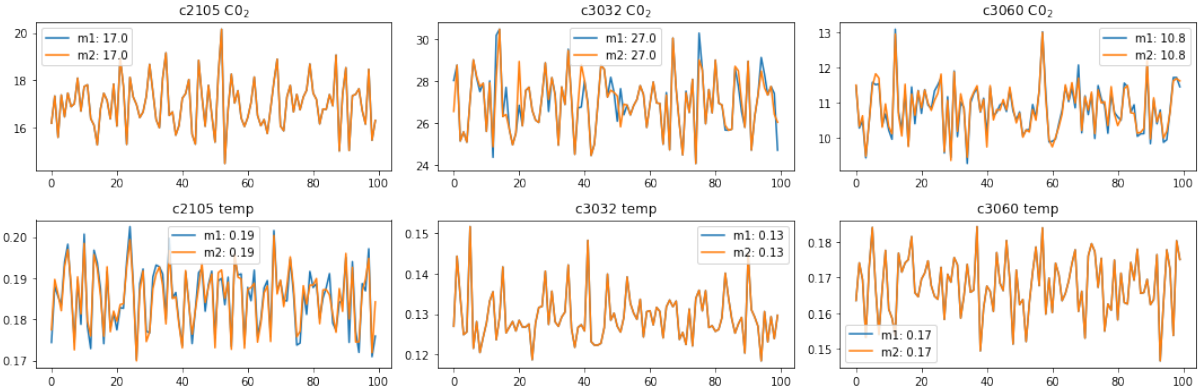

Figure 5.10: regression tree tuning results, m1 model is based on grid search, m2 is model based on random search.

Table 5.9: Average RMSE values from fitted regression trees based on parameter search approach.

	c2105		c3032		c3060	
	CO_2		CO_2		CO_2	
	Grid Search	Random Search	Grid Search	Random Search	Grid Search	Random Search
RMSE	17.0	17.0	27.0	27.0	10.8	10.8
	Temperature		Temperature		Temperature	[b]
	Grid Search	Random Search	Grid Search	Random Search	Grid Search	Random Search
RMSE	0.19	0.19	0.13	0.13	0.17	0.17

Parameter settings for each model (m1 and m2) are presented in Table 5.8. For example, model m1 for CO_2 refers to a model whose tree is designed to have a maximum depth(max_depth) of 5, the applied tree splitting criteria (criterion) is mse and the maximum number of features applied to the tree(max_features) is set to auto meaning that all features are considered. Other corresponding models follow the same pattern.

These average RMSE results are shown in Table 5.9 the results are similar between the parameter searching approaches. An example of a fitted regression tree used to predict CO_2 is presented in Figure 5.11, the presented tree is none boosted but shows important features used to predict CO_2, the tree has a maximum depth of 3, the function used to measure the quality of a split (criterion) is set to MSE and the number of features considered when selecting a split(max_features) is set to auto, which means all features are considered in each split of the tree. According to this model presented by the tree, it shows that only temperature(Lampotila) and TVOC are relevant features for predicting CO_2. Each node presents a decision that eventually determines the predicted CO_2 value.

A similar tree for predicting temp is presented in Fig 5.12, the tree is set to similar parameter values for demonstration purposes, max_dept=3, criterion=mse and

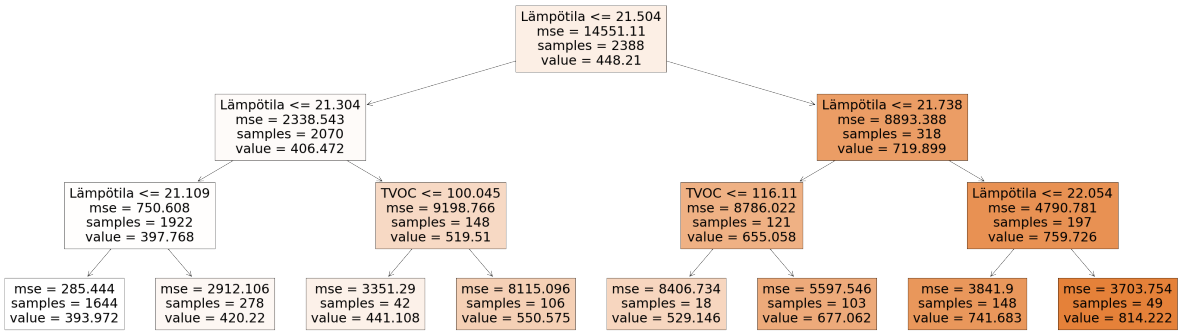

Figure 5.11: An example of a CO_2 regression tree

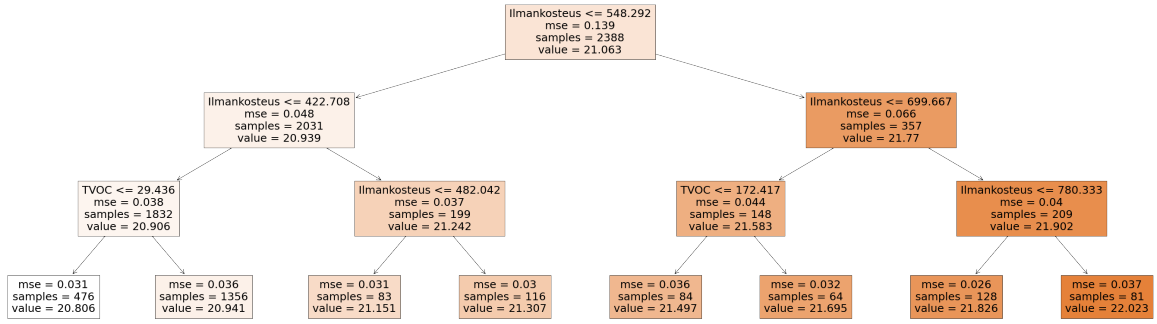

Figure 5.12: An example of a CO_2 regression tree

max_feature=auto. In this model, humidity(Ilmankoetus) and TVOC are the features used to predict temperature. Generally, using a regression tree-based model has the added advantage that it allows testing non-parametric models use case and the assumption of linearity that is assumed in linear models is not necessary in this case.

Boosted regression trees with m2 settings shown in Table **??** were tested and their results are presented in Table 5.10. In these tests, the test set is held at 33% of the whole dataset in each of the model's runs. As expected, training errors are lower than the test errors, in some cases, the margin between the training error and test error is large (c3032) in both the CO_2 case and the temperature case. This points to structural issues with that particular dataset.

5.3.4 Modeling CO_2 and Temperature using ARIMA models

Finally, the last modeling strategy follows a time series approach, where models that are specific to time series data are applied. To motivate this approach a plot of time-shifted series is plotted against non-time-shifted series. The plot shown in Fig 5.13 visually confirms the presence of autocorrelation among the data, particularly from rooms c2105 and c3060. Subplots in the top row of the figure indicate that CO_2 observations one day forward $(t + 1)$ have a strong linear dependency to present (t) observations across. The

Table 5.10: RMSE values obtained from boosted regression trees

	CO_2	
	Train RMSE	**Test RMSE**
c2105	23	28
c3032	44	68
c3060	41	47

	Temperature	
	Train RMSE	**Test RMSE**
c2105	0.16	0.21
c3032	0.14	0.33
c3060	0.13	0.15

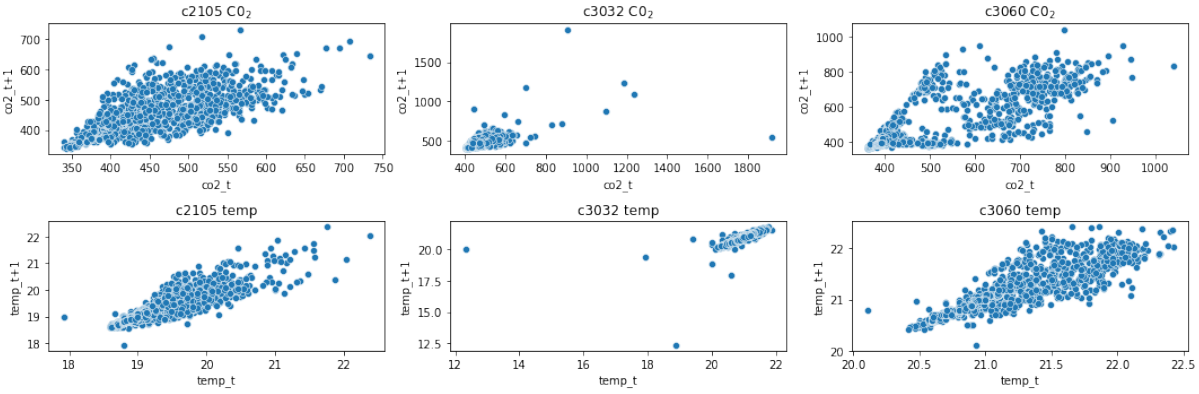

Figure 5.13: An example of a CO_2 regression tree

same conclusion can be observed on the temperature data shown in the second row of the figure. Data based on room c3032 shows no strong autocorrelation both in CO_2 and temperature data. These observations generally provide the required evidence to further explore time-series models.

Auto Correlation Function (ACF) and Partial Auto Correlation Function (PACF) plots were generated to determine the structure of the underlying structural time series characteristics of the data. ACF and PACF Plots related to CO_2 ACF and PACF plots are shown in Fig 5.14. These plots indicate the presence of a seasonal component and non-seasonal component in each of the CO_2 data sets, in particular, the ACF and PACF plots show a significant lag at lag 24. Notably, all the cO_2 datasets produce plots with a similar structure.

Similar ACF and PACF plots for temperature are also presented in Fig 5.15, these plots show significant lags at lag 24 for the data sets representing c2105 and c3060 while c3032 shows the presence of seasonality at lag 24 although the plots are distinctively dif-

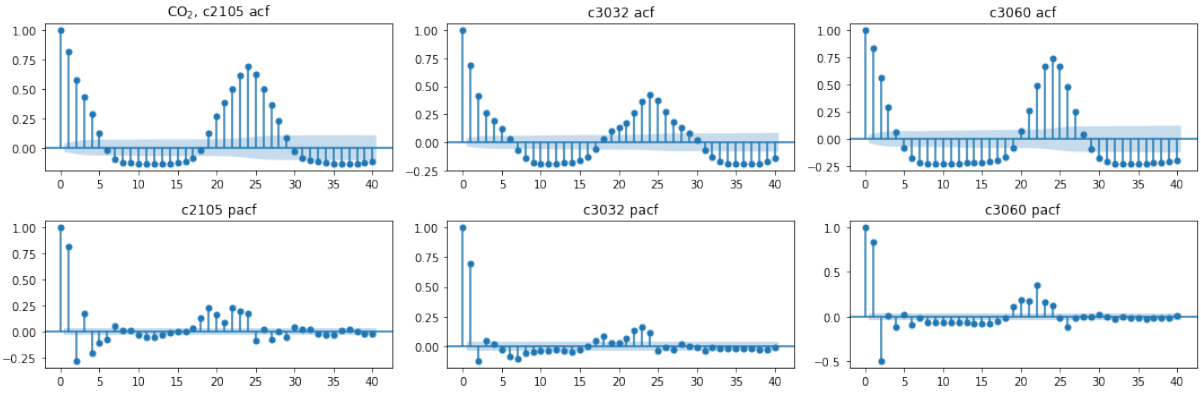

Figure 5.14: ACF and PACF plots for CO_2

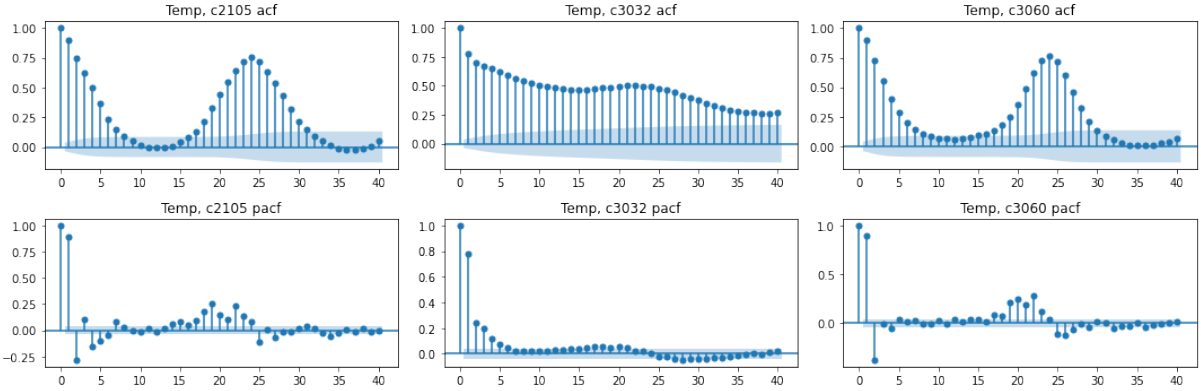

Figure 5.15: ACF and PACF plots for Temperature

ferent from the rest. In general, these plots confirm the presence of a seasonal component and a nonseasonal component across the time-series.

To distinguish between the performance of various configurations of a suitable time series model, a variety of seasonality configurations are simulated and compared for their performance based on the Akaike Information Criteria (AIC) approach. Simulation results are presented in Table 5.11 where the best parameter set for each dataset is presented. The presented parameters suggest the best parameters used to fit the seasonal Auto-Regressive Integrated Moving Average (ARIMA) model for each of the datasets from a possible 64 settings of varying ARIMA order and Seasonal Order parameters. The best CO_2 model for c2105 is for example is fitted with a model with an ARIMA order set to (1, 0, 1) and a seasonal order set to (1, 1, 1, 24).

Corresponding results obtained from testing these models are shown in Table 5.11 which reports their relevant RMSE values. The training error from the c3032 CO_2 model is remarkably different from the test error and the test performs better than the training set. Similarly, test errors in Temperature models are lower than the test set errors, this is contrary to the expected behavior. A cross-validation approach was taken to study the development of RMSE values for each of the models listed in Table 5.11, the results of this

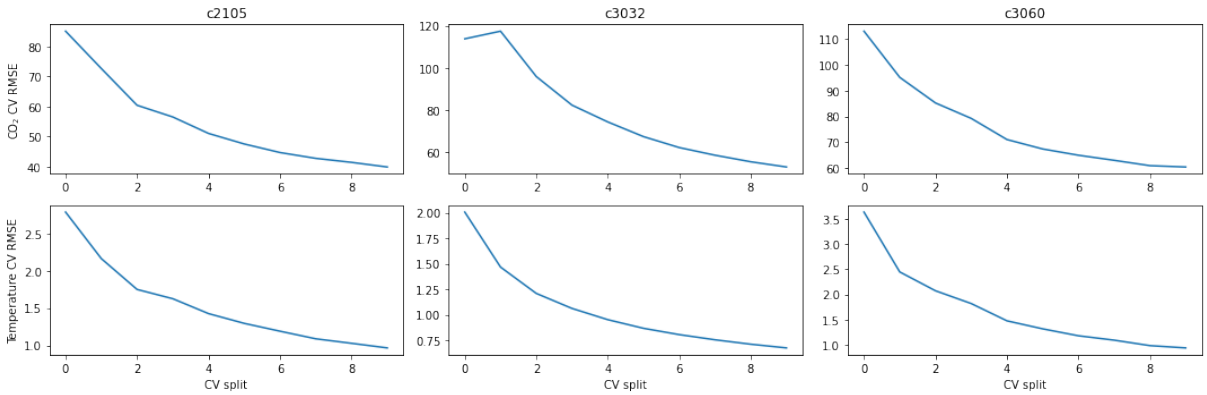

Figure 5.16: Timeseries cross validation of ARIMA models.

Table 5.11: Simulation results and RMSE values obtained from these ARIMA models

	CO_2			
	ARIMA Order	**Seasonal Order**	**Train RMSE**	**Test RMSE**
c2105	(1, 0, 1)	(1, 1, 1, 24)	44	48
c3032	(1, 0, 1)	(1, 1, 1, 24)	61	47
c3060	(1, 0, 1)	(1, 1, 1, 24)	64	71
	Temperature			
	ARIMA Order	**Seasonal Order**	**Train RMSE**	**Test RMSE**
c2105	(1, 0, 1)	(1, 1, 1, 24)	1.1	0.4
c3032	(1, 1, 1)	(1, 0, 1, 24)	0.6	0.3
c3060	(1, 0, 1)	(1, 1, 1, 24)	1.1	0.3

cross-validation exercise are shown in Figure 5.16 which shows that the error decreases in subsequent cross-validation iterations. This is because more training data is included in the training model for each of the iterations such that iteration 0 contains the smallest training set while iteration 9 contains the largest training set. For this reason, the RMSE value decreases to the right. Due to the observable trend (negative), the average of the RMSE is not calculated, instead only the lowest RMSE value is picked as the test RMSE.

5.4 Data modeling results

In general, modeling indoor air quality data has involved testing various approaches to determine good models since the obtained sensor data was not structurally similar even though the same air quality parameters were being measured across rooms. The modeling process began by exploring high-level characteristics of the data followed by an in-depth

modeling process to identify methods and models to predict CO_2 and temperature as indicators of indoor air quality and thermal comfort in a building.

The modeling part began with applying PCA, which was considered a good tool to model all features/variables collected in the indoor environment. PCA has an added benefit that the parameter space of all indoor air quality variables can be reduced and represented with fewer variables and still carry a significant variance.

The second approach applied feature selection where features were added to a model one feature at a time as their impact on the model's error values are observed. This provided the benefit of being able to observe a single variable's impact towards improving, deteriorating, or having no change to desired results.

Thirdly, decision trees were applied to the modeling problem, these were considered a non-parametric control tool and also a good candidate for modeling the data given dataset since some of the relationships between these variables was non-linear

The final model applied was a dedicated time series model (ARIMA) that takes into account the time-series nature of the data. This provided the tools required to model aspects such as seasonality and auto-correlation inherent in the data.

To summarise all data modeling findings and especially the details of the model tuning process, the best model obtained from each algorithm is picked for comparison and their resulting performance is summarily compared in Table 5.12 and a corresponding Figure 5.17. Indicated results are obtained by 10-fold with 10 repetitions cross-validation. The plots compare RMSE values obtained by each model(y-axis) against an applied prediction approach/algorithm (x-axis) in a specific dataset represented by a classroom (c2105, c3032, and c3060).

The regression tree approach (boosted regression trees with 300 estimators) produced the best results across CO_2 and temperature models. The only observed exception appears in CO_2 c3032 model. An additional conclusion is that the boosted regression trees are best suited to fit a forecasting model for this data.

For room c2105, the best CO_2 prediction model is based on a boosted tree algorithm (RMSE 24.68), similarly, a boosted tree algorithm was observed to produce the best re-

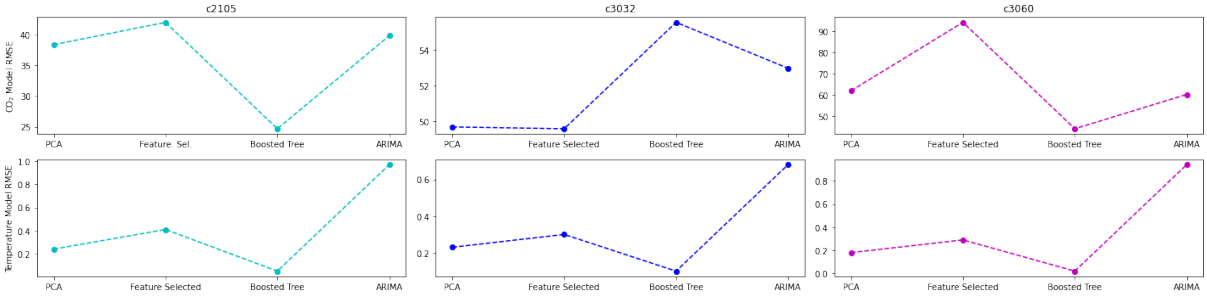

Figure 5.17: A comparison of prediction algorithms performance across the datasets.

Table 5.12: Summary of RMSE obtained by the best model in each fitting process.

c2105				
	PCA	Feature Selection	Boosted Tree	ARIMA Model
CO_2 RMSE	38.38	42.02	24.68	39.83
Temperature RMSE	0.24	0.41	0.05	0.97
c3032				
	PCA	Feature Selection	Boosted Tree	ARIMA Model
CO_2 RMSE	49.70	49.60	55.54	52.98
Temperature RMSE	0.23	0.30	0.10	0.68
c3060				
	PCA	Feature Selection	Boosted Tree	ARIMA Model
CO_2 RMSE	62.07	94.27	44.19	60.32
Temperature RMSE	0.18	0.29	0.02	0.94

sults for temperature data (0.05). Room c3032 shows that the PCA and Feature selection approach produced the best performing CO_2 models while a boosted tree algorithm produced the best results for the room's temperature data. The best performing algorithms in Room c3060 CO_2 are boosted trees, the same applies to its temperature models. Based on these results, room c3032 appears to have unique characteristics compared to the other two rooms that have the same algorithm performing well for both CO_2 and temperature data.

These results imply that room-specific characteristics need to be taken into account in a setting where control of thermal comfort and indoor air quality aims to be automated. Secondly, based on the structure of the data, most variables recorded in the indoor environment do not have a linear relationship therefore algorithms with non-linear assumptions will tend to perform better.

6. Discussion

This study was undertaken in the context of smart buildings where buildings are equipped with different kinds of sensors. In this case, the reference building for this study was equipped with indoor air quality sensors in three rooms. The notion of Smart buildings has increasingly seen a rise in focus towards managing the indoor environment with the sole purpose of improving indoor air quality and thermal comfort of the occupants.

Previous studies that have demonstrated modeling of temperature and CO_2 parameters focus on only modeling these two parameters independently. However, the indoor environment consists of much more parameters that could be measured with modern sensors and that is the premise of this study. Indoor climate variables such as humidity, pressure, total volatile organic compounds (TVOC), and small particles of various sizes have been measured and the data used for investigating the overall quality of the indoor environment in this study.

6.1 Data modeling

The data exploratory part of the study indicated that the studied rooms were on average within the prescribed band of minimum and maximum operating temperatures, except for one room (c2105) whose minimum temperature was below the prescribed minimum value. CO_2 levels were observed to be within the recommended accumulation amounts and so were the occupancy levels in the given rooms. On the other hand, ventilation rates were observed to be set below the recommended levels as per the local building guidelines. Additionally, a variety of strategies were applied to the model prediction of temperature and CO_2, in this case, boosted regression trees were observed to produce the best prediction results.

These results demonstrate that variables measured within the indoor environment tend not to be correlated which means they are not strong predictors of each other. However, these variables are useful in modeling the collective quality of the indoor environment. A technique such as *Principal Component Analysis* demonstrated that all indoor environment variables (15 of them in this study) could be combined into 4 or 5 Principal components that carry much of the information value about the indoor environ-

ment. If such an approach is adopted, the resulting components can then be published for other modeling purposes without disclosing the actual elements measured in the indoor environment.

The predictive modeling exercise demonstrated that a variety of approaches can be applied to predict temperature or CO_2. These approaches can be categorized into two broad groups, those that make use of other variables measured in the indoor environment and those that are based on the variable itself(*Auto Regressive models*), where the model makes use of a variable's past values to predict it's future value(s). Viewed from this perspective, the best scoring models were obtained from the group that made use of other indoor environment variables(boosted regression trees). This provides the incentive to collect multiple indoor environment variables for good temperature and CO_2 predictive models.

It could be further argued that among those models that make use of multiple indoor variables, those algorithms that do not depend on the assumption of linearity (boosted regression trees) produced the best results compared to those that assume linear dependency among the variables (linear regressions, autoregression models). On the other hand, not all collected data variables are important when it comes to predicting temperature and CO_2, this is evidenced when selecting features for inclusion into a model and also when tuning regression trees.

These types of analysis may find utility in a digital twin setting where a virtual counterpart can include software components that produce analytics and intelligence support for maintenance activities of a building. Even more useful would be to provide an avenue to observe the effectiveness of HVAC systems and their settings relative to air quality requirements and monitoring critical variables such as humidity that can harm a building.

6.2 Application design

There is a close relationship almost symbiotic, between software and data, business objectives can be encapsulated in software applications that make use of data and derived models. Part of this study involved creating an application with the main goal of presenting analytics, visualizations, and other data modeling results. A key part of the effort was in designing a suitable API that would provide a scalable solution to explore sensor data.

Generally, the two common architectures applicable in creating web-based APIs are Simple Object Access Protocol (SOAP) and REpresentational State Transfer (REST). REST's simple HTTP-based interface, support for JavaScript Object Notation(JSON) and speed, make it an attractive architectural solution for creating lightweight web-based

APIs. Most open or public APIs tend to be designed as RESTful APIs since they are easy for developers to understand and therefore easier for API consumers to integrate into other applications. In this study, a REST architecture was used where multiple rooms and multiple sensors per room were mapped as URI resources which provided sensor data as endpoints. Furthermore, making use of query strings allowed for a simple API design and yet support a variety of air quality sensors and data configuration options through query parameters.

As smart buildings approach digital twin models, one of the considerations among many may involve the installation of multiple sensors in all available spaces/rooms within the building. Implications of such a development have been tested in this study by modeling using data collected from three distinct rooms. Secondly, modeling of data to provide intelligence and insights back to the physical counterpart will be required as input for adjusting the indoor environment. The third consideration concerns the issue of integration between components in the virtual space and the physical space, depending on a building's use case, other API architecture beyond REST could be considered especially for any security and transaction reliability constraints.

6.3 Validity

This study contains an integration of two fields: software engineering and data science. Consequently, threats to this study's validity can emerge from these two angles. This section on validity focuses on internal and conclusion validity [36] concerning the data science aspects of the study and how these threats were mitigated.

Software engineering validity concerns may include the choice of open source technologies used. Such a concern is challenging to mitigate since no specific benchmark of a complete solution is used to compare the design and results of the design choices undertaken in this study. Secondly, current open-source software tools are anticipated to have similar reliability as proprietary software.

Threats related to data analysis can begin from the reliability of measurements taken by sensors. To control for potential episodes of unreliable readings, outlier events were removed from the data during the data cleaning process and before data analysis processes. In the data analysis process, threats to validity were greatly controlled through cross-validation, in practice, this means algorithms were mostly run on 100 repetitions, and results were recorded as the average of these runs. There are however improvements that could be undertaken to improve the internal validity of this study. This study has been conducted based on data collected from one building, therefore part of the results may be inclined to depict conditions specific to the building where the data was obtained and its indoor climate control settings. To generate a more generic result, more data

should be gathered particularly from other buildings of a similar use case to broaden the sample space and support the development of more generic models. Adding buildings with different use cases and sizes would also enrich such results.

One shortcoming of the data used for this study is the fact that the observations were recorded between September 2019 to March 2020, which means the data largely misses observations from the Summer and a significant part of the Fall season. Lack of full-year observations could affect the overall results given that changes that occur in the outdoor climate may affect indoor occupants' behavior and the variables recorded in the indoor environment. Further, full-year datasets may provide a basis to conduct seasonality adjusted analysis especially if multiple years of data can be obtained.

To improve the validity of prediction results and effectively control for the validity of the conclusions, diverse algorithms were applied to solve the same prediction problem. This approach served the purpose of identifying the best algorithm and therefore increasing confidence over the choice of model.

Alternative tools and algorithms such as neural networks can be used to conduct similar data analysis and the design of software applications can follow different design approaches. However, it is expected the choice of algorithms and technologies for application design should not greatly alter the overall results as the family of applicable algorithms have been already been presented by currently selected algorithms.

6.4 Future considerations

The data set obtained for this study demonstrated that a rich data set can be useful in modeling various aspects of the indoor environment. It is potentially possible to construct a composite indoor air quality index given similar or more variables gathered for this study. However, additional expert knowledge on the actual composition of such an index would be required to make it scientifically sound.

Given that the data used in this study can be highly granular and pervasive, such data might constitute personal data which may require additional legal frameworks to process and store. Exploration of strategies that could be applied to abstract or anonymize such data would assist in making stakeholders more willing to contribute such datasets as open data for research and other innovation purposes.

7. Conclusions

This study investigates the application of multiple sensor data on the modeling of indoor air quality. This includes data analysis and the creation of web-based APIs and an application. A design science methodology was applied as the research methodology.

The data was first reviewed within the context of local construction guidelines and recommendations. Also, data modeling was extended to machine learning approaches aimed at predicting indoor air quality whose proxy is CO_2 and thermal comfort using indoor temperature data. Predicting CO_2 and the temperature was most successful when using Ada Boosted regression trees compared to other algorithms applied.

The study shows that realizing customized indoor environments in buildings may require heterogeneous models. Such a result can pose design challenges to HVAC systems as most of them tend to be centrally controlled and universally installed across buildings.

A designed artifact in the form of a software system was developed to evaluate constraints and opportunities in a web-based sensor data API. The web API was based on a micro-service architecture and the REST protocol for data exchange. The REST architectural style was considered to provide ease of integration and a scalable design that was necessary to support multiple sensor data.

Limitations such as the size of the dataset and a single source of data may limit the generalization of the results presented in this study. However, the approach presented where more indoor variables are considered in the study of thermal comfort and indoor air quality provides new insights on the behavior of indoor environments especially in different spaces within the same building.

This study also shows that algorithms that do not rely on the assumption of linearity among the indoor variables tend to perform better. In the same breath, models that make use of these additional variables in some form tend to perform better than algorithms and techniques that do not make use of other exogenous variables.

Further studies in the realm of digital twins in the context of smart buildings remain scanty and therefore an open research opportunity. Equally rare is open data from smart buildings whose availability would foster research and innovation of digital services.

Bibliography

[1] M. Ahola, J. Säteri, and L. Sariola. Revised finnish classification of indoor climate 2018. In *E3S Web of Conferences*, volume 111. EDP Sciences, 2019.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[3] L. Atzori, A. Iera, and G. Morabito. Understanding the internet of things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, 56:122–140, 2017.

[4] R. Baheti and H. Gill. Cyber-physical systems. *The impact of control technology*, 12(1):161–166, 2011.

[5] B. R. Barricelli, E. Casiraghi, and D. Fogli. A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE Access*, 7:167653–167671, 2019.

[6] M. R. Bashir and A. Q. Gill. Towards an iot big data analytics framework: Smart buildings systems. In *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 1325–1332, 2016.

[7] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, et al. Web services description language (wsdl) 1.1, 2001.

[8] S. Cirani, L. Davoli, G. Ferrari, R. Léone, P. Medagliani, M. Picone, and L. Veltri. A scalable and self-configuring architecture for service discovery in the internet of things. *IEEE internet of things journal*, 1(5):508–521, 2014.

[9] J. Davis, T. Edgar, J. Porter, J. Bernaden, and M. Sarli. Smart manufacturing, manufacturing intelligence and demand-dynamic performance. *Computers & Chemical Engineering*, 47:145–156, 2012.

[10] D. Enescu. A review of thermal comfort models and indicators for indoor environments. *Renewable and Sustainable Energy Reviews*, 79:1353–1379, 2017.

[11] J. Erickson and K. Siau. Service oriented architecture: A research review from the software and applications perspective. *Innovations in Information Systems Modeling: Methods and Best Practices*, pages 190–203, 2009.

[12] I. A. Essa. Ubiquitous sensing for smart and aware environments. *IEEE Personal Communications*, 7(5):47–49, 2000.

[13] R. T. Fielding and R. N. Taylor. *Architectural styles and the design of network-based software architectures*, volume 7. University of California, Irvine Irvine, 2000.

[14] A. González-Vidal, V. Moreno-Cano, F. Terroso-Sáenz, and A. F. Skarmeta. Towards energy efficiency smart buildings models based on intelligent data analytics. *Procedia Computer Science*, 83:994–999, 2016.

[15] M. Grieves. Digital twin: Manufacturing excellence through virtual factory replication. 03 2015.

[16] M. Grieves and J. Vickers. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In *Transdisciplinary perspectives on complex systems*, pages 85–113. Springer, 2017.

[17] D. Guinard and V. Trifa. Towards the web of things: Web mashups for embedded devices. In *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain*, volume 15, page 8, 2009.

[18] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio. Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. *IEEE Transactions on Services Computing*, 3(3):223–235, 2010.

[19] I. Ishaq, D. Carels, G. K. Teklemariam, J. Hoebeke, F. V. d. Abeele, E. D. Poorter, I. Moerman, and P. Demeester. Ietf standardization in the field of the internet of things (iot): a survey. *Journal of Sensor and Actuator Networks*, 2(2):235–287, 2013.

[20] N. Jazdi. Cyber physical systems in the context of industry 4.0. In *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, pages 1–4, 2014.

[21] P. R. Jennings, D. Fahringer, and T. Collins. Sick building syndrome indoor air quality and your patients' health. *JAAPA-Journal of the American Academy of Physicians Assistants*, 13(8):34–34, 2000.

[22] X. Jia, Q. Feng, T. Fan, and Q. Lei. Rfid technology and its applications in internet of things (iot). In *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, pages 1282–1285, 2012.

[23] Jiehan Zhou, T. Leppanen, E. Harjula, M. Ylianttila, T. Ojala, Chen Yu, Hai Jin, and L. T. Yang. Cloudthings: A common architecture for integrating the internet of things with cloud computing. In *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 651–657, 2013.

[24] A. Kaushik, M. Arif, P. Tumula, and O. J. Ebohon. Effect of thermal comfort on occupant productivity in office buildings: Response surface analysis. *Building and Environment*, 180:107021, 2020.

[25] E. A. Lee. Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369, 2008.

[26] J. Lee, H.-A. Kao, S. Yang, et al. Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia Cirp*, 16(1):3–8, 2014.

[27] J. Lee, Y. Su, and C. Shen. A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi. In *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*, pages 46–51, 2007.

[28] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer. A comparative study of lpwan technologies for large-scale iot deployment. *ICT express*, 5(1):1–7, 2019.

[29] D. Minoli, K. Sohraby, and B. Occhiogrosso. Iot considerations, requirements, and architectures for smart buildingsâenergy optimization and next-generation building management systems. *IEEE Internet of Things Journal*, 4(1):269–283, 2017.

[30] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking. A review on internet of things (iot), internet of everything (ioe) and internet of nano things (iont). In *2015 Internet Technologies and Applications (ITA)*, pages 219–224, 2015.

[31] N. Mohamed, J. Al-Jaroodi, and I. Jawhar. Service-oriented big data analytics for improving buildings energy management in smart cities. In *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*, pages 1243–1248, 2018.

[32] E. Negri, L. Fumagalli, and M. Macchi. A review of the roles of digital twin in cps-based production systems. *Procedia Manufacturing*, 11:939–948, 2017.

[33] C. Pautasso, O. Zimmermann, and F. Leymann. Restful web services vs." big"'web services: making the right architectural decision. In *Proceedings of the 17th international conference on World Wide Web*, pages 805–814, 2008.

[34] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77, 2007.

[35] R. Perrey and M. Lycett. Service-oriented architecture. In *2003 Symposium on Applications and the Internet Workshops, 2003. Proceedings.*, pages 116–119, 2003.

[36] K. Petersen and C. Gencel. Worldviews, research methods, and their relationship to validity in empirical software engineering research. In *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement*, pages 81–89, 2013.

[37] T. Savolainen, J. Soininen, and B. Silverajan. Ipv6 addressing strategies for iot. *IEEE Sensors Journal*, 13(10):3511–3519, 2013.

[38] SciPy. Numpy documentation version 1.19 manual.

[39] C.-S. Shih, J.-J. Chou, N. Reijers, and T.-W. Kuo. Designing cps/iot applications for smart buildings and cities. *IET Cyber-Physical Systems: Theory & Applications*, 1(1):3–12, 2016.

[40] B. N. Silva, M. Khan, K. Lee, Y. Yoon, D. Muhammad, J. Han, and K. Han. Restful web of things for ubiquitous smart home energy management. In *2020 International Conference on Computing, Networking and Communications (ICNC)*, pages 176–180, 2020.

[41] Y. Sun, H. Song, A. J. Jara, and R. Bie. Internet of things and big data analytics for smart and connected communities. *IEEE Access*, 4:766–773, 2016.

[42] W. Tan, Y. Fan, A. Ghoneim, M. A. Hossain, and S. Dustdar. From the service-oriented architecture to the web api economy. *IEEE Internet Computing*, 20(4):64–68, 2016.

[43] F. Tao, H. Zhang, A. Liu, and A. Y. Nee. Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*, 15(4):2405–2415, 2018.

[44] Z. Wang, R. de Dear, M. Luo, B. Lin, Y. He, A. Ghahramani, and Y. Zhu. Individual difference in thermal comfort: A literature review. *Building and Environment*, 138:181–193, 2018.

[45] L. D. Xu, W. He, and S. Li. Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4):2233–2243, 2014.

[46] I. Yaqoob, K. Salah, M. Imran, P. P. Jayaraman, C. Perera, et al. The role of big data analytics in industrial internet of things. *arXiv preprint arXiv:1904.05556*, 2019.

Appendix A.

Table A.1: Sensors and URIs from one location, other locations have similar URIs

Higher Level URI (Room URI)	http://...room/c2105
	Sensors URIs
Henkilomaara	
Kokonaismaara	http://...c2105?category=5&sensor=1&freq=W&resampler=mean
Lukumaara sisaan	http://...c2105?category=5&sensor=0&freq=W&resampler=mean
Lukumaara ulos	http://...c2105?category=5&sensor=2&freq=W&resampler=mean
Paristojannite	http://...c2105?category=5&sensor=3&freq=W&resampler=mean
Poistoilma	
Ilmankosteus	http://...c2105?category=2&sensor=0&freq=W&resampler=mean
Lampotila	http://...c2105?category=2&sensor=1&freq=W&resampler=mean
Paristojannite	http://...c2105?category=2&sensor=2&freq=W&resampler=mean
Signaalin voimakkuus, lahetys	http://...c2105?category=2&sensor=4&freq=W&resampler=mean
Signaalin voimakkuus, vastaanotto	http://...c2105?category=2&sensor=3&freq=W&resampler=mean
Poistoilman tilavuusvirta (l/s)	
Ilmanpaine-ero	http://...c2105?category=4&sensor=1&freq=W&resampler=mean
Ilmavirran tilavuus litroina	http://...c2105?category=4&sensor=0&freq=W&resampler=mean
Signaalin voimakkuus, vastaanotto	http://...c2105?category=4&sensor=2&freq=W&resampler=mean
Sisailman laatu	
Hiilidioksidi	http://...c2105?category=1&sensor=1&freq=W&resampler=mean
Ilmankosteus	http://...c2105?category=1&sensor=2&freq=W&resampler=mean
Ilmanpaine-ero	http://...c2105?category=1&sensor=3&freq=W&resampler=mean
Lampotila	http://...c2105?category=1&sensor=4&freq=W&resampler=mean
Massakonsentraatio PM1.0	http://...c2105?category=1&sensor=13&freq=W&resampler=mean
Massakonsentraatio PM10.0	http://...c2105?category=1&sensor=14&freq=W&resampler=mean
Massakonsentraatio PM2.5	http://...c2105?category=1&sensor=11&freq=W&resampler=mean
Massakonsentraatio PM4.0	http://...c2105?category=1&sensor=10&freq=W&resampler=mean
Maarakonsentraatio PM0.5	http://...c2105?category=1&sensor=5&freq=W&resampler=mean
Maarakonsentraatio PM1.0	http://...c2105?category=1&sensor=6&freq=W&resampler=mean
Maarakonsentraatio PM10.0	http://...c2105?category=1&sensor=12&freq=W&resampler=mean
Maarakonsentraatio PM2.5	http://...c2105?category=1&sensor=8&freq=W&resampler=mean
Maarakonsentraatio PM4.0	http://...c2105?category=1&sensor=15&freq=W&resampler=mean
Paine	http://...c2105?category=1&sensor=9&freq=W&resampler=mean
TVOC	http://...c2105?category=1&sensor=0&freq=W&resampler=mean
Tyypillinen hiukkaskoko	http://...c2105?category=1&sensor=7&freq=W&resampler=mean
Tuloilma	
Ilmankosteus	http://...c2105?category=0&sensor=1&freq=W&resampler=mean
Lampotila	http://...c2105?category=0&sensor=0&freq=W&resampler=mean
Paristojannite	http://...c2105?category=0&sensor=2&freq=W&resampler=mean
Tuloilman tilavuusvirta (l/s)	
Ilmanpaine-ero	http://...c2105?category=3&sensor=1&freq=W&resampler=mean
Ilmavirran tilavuus litroina	http://...c2105?category=3&sensor=0&freq=W&resampler=mean
Signaalin voimakkuus, vastaanotto	http://...c2105?category=3&sensor=2&freq=W&resampler=mean

Table A.2: Sensors and corresponding units of measurements

Sensor	**Measurement Units**
Henkilomaara	
Kokonaismaara	pcs
Lukumaara sisaan	pcs
Lukumaara ulos	pcs
Paristojannite	V
Poistoilma	
Ilmankosteus	%
Lampotila	°C
Paristojannite	V
Signaalin voimakkuus, lahetys	dBm
Signaalin voimakkuus, vastaanotto	dBm
Poistoilman tilavuusvirta (l/s)	
Ilmanpaine-ero	Pa
Ilmavirran tilavuus litroina	slm
Signaalin voimakkuus, vastaanotto	dBm
Sisailman laatu	
Hiilidioksidi	ppm
Ilmankosteus	%
Ilmanpaine-ero	Pa
Lampotila	°C
Massakonsentraatio PM1.0	$\mu g/m^3$
Massakonsentraatio PM10.0	$\mu g/m^3$
Massakonsentraatio PM2.5	$\mu g/m^3$
Massakonsentraatio PM4.0	$\mu g/m^3$
Maarakonsentraatio PM0.5	pcs/cm^3
Maarakonsentraatio PM1.0	pcs/cm^3
Maarakonsentraatio PM10.0	pcs/cm^3
Maarakonsentraatio PM2.5	pcs/cm^3
Maarakonsentraatio PM4.0	pcs/cm^3
Paine	hPa
TVOC	ppb
Tyypillinen hiukkaskoko	μm
Tuloilma	
Ilmankosteus	%
Lampotila	°C
Paristojannite	V
Tuloilman tilavuusvirta (l/s)	
Ilmanpaine-ero	Pa
Ilmavirran tilavuus litroina	slm
Signaalin voimakkuus, vastaanotto	dBm

Table A.3: A translation from Finnish to English of the sensor categories and their respective sensors

Finnish Term	English Translation
Henkilomaara	Number of people
Kokonaismaara	Total amount
Lukumaara sisaan	Number in
Lukumaara ulos	Number out
Paristojannite	Battery voltage
Poistoilma	Outgoing air
Ilmankosteus	Humidity
Lampotila	Temperature
Paristojannite	Battery voltage
Signaalin voimakkuus, lahetys	Signal strength, transmission
Signaalin voimakkuus, vastaanotto	Signal strength, reception
Poistoilman tilavuusvirta (l/s)	Outgoing air volume flow (l / s)
Ilmanpaine-ero	Barometric pressure difference
Ilmavirran tilavuus litroina	Air flow volume in liters
Signaalin voimakkuus, vastaanotto	Signal strength, reception
Sisailman laatu	Indoor air quality
Hiilidioksidi	Carbon dioxide
Ilmankosteus	Humidity
Ilmanpaine-ero	Barometric pressure difference
Lampotila	Temperature
Massakonsentraatio PM1.0	Mass concentration PM1.0
Massakonsentraatio PM10.0	Mass concentration PM10.0
Massakonsentraatio PM2.5	Mass concentration PM2.5
Massakonsentraatio PM4.0	Mass concentration PM4.0
Maarakonsentraatio PM0.5	Quantitative concentration PM0.5
Maarakonsentraatio PM1.0	Quantity concentration PM1.0
Maarakonsentraatio PM10.0	Quantity concentration PM10.0
Maarakonsentraatio PM2.5	Quantity concentration PM2.5
Maarakonsentraatio PM4.0	Quantity concentration PM4.0
Paine	Pressure
TVOC	TVOC
Tyypillinen hiukkaskoko	Typical particle size
Tuloilma	Supply air
Ilmankosteus	Humidity
Lampotila	Temperature
Paristojannite	Battery voltage
Tuloilman tilavuusvirta (l/s)	Supply air volume flow (l / s)
Ilmanpaine-ero	Barometric pressure difference
Ilmavirran tilavuus litroina	Air flow volume in liters
Signaalin voimakkuus, vastaanotto	Signal strength, reception