# EDGE COMPUTING APPROACH TO INDOOR TEMPERATURE PREDICTION USING MACHINE LEARNING

by

**Hyemin Kim**

**A Thesis**

*Submitted to the Faculty of Purdue University*

*In Partial Fulfillment of the Requirements for the degree of*

**Master of Science**

Department of Computer and Information Technology

West Lafayette, Indiana

December 2021

# THE PURDUE UNIVERSITY GRADUATE SCHOOL
# STATEMENT OF COMMITTEE APPROVAL


**Dr. Eric T. Matson, Chair**

Department of Computer and Information Technology

**Prof. Anthony H. Smith**

Department of Computer and Information Technology

**Dr. Byung-Cheol Min**

Department of Computer and Information Technology


**Approved by:**

Dr. John A. Springer

*This thesis is dedicated to God, my family, and my friends, who are always there for me.*

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| **IoT** | Internet of Things |
| **ICT** | Information and Communications Technology |
| **WSN** | Wireless Sensor Network |
| **FRL** | Federated Region-Learning |
| **LPWAN** | Low-Power Wide-Area Network |
| **AP** | Access Point |
| **PoE** | Power Over Ethernet |
| **ECD** | Edge Computing Device |

# GLOSSARY

**Artificial Neural Network (ANN)** – a computational model inspired by networks of biological neurons, wherein neurons compute output values from inputs.

**Cloud Computing** – a computing paradigm that shifts the location of computer infrastructure to the network.

**Edge Computing** – a computing paradigm that makes use of resources at the edge.

**Internet of Things** – an extension of the Internet in which physical devices, vehicles, buildings, and other physical items are enabled to collect and transfer data and provide services via connections.

**Machine Learning** – a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

**Smart Building** – a building that uses its intelligence to collect actionable data from user devices, sensors, systems, and services on the premises. Using artificial intelligence and machine learning (AI/ML) makes the building both programmable and responsive to the needs of the users and the building manager.

**Smart City** – a city where investments in human and social capital and traditional and modern communication infrastructure fuel sustainable economic growth and high quality of life, with a wise management of natural resources.

**Time Series Forecasting** – forecasts that are made based on data comprising one or more time series

# ABSTRACT

This paper aims to present a novel approach to real-time indoor temperature forecasting to meet energy consumption constraints in buildings, utilizing computing resources available at the edge of a network, close to data sources. This work was inspired by the irreversible effects of global warming accelerated by greenhouse gas emissions from burning fossil fuels. As much as human activities have heavy impacts on global energy use, it is of utmost importance to reduce the amount of energy consumed in every possible scenario where humans are involved. According to the US Environmental Protection Agency (EPA), one of the biggest greenhouse gas sources is commercial and residential buildings, which took up 13 percent of 2019 greenhouse gas emissions in the United States. In this context, it is assumed that information of the building environment such as indoor temperature and indoor humidity, and predictions based on the information can contribute to more accurate and efficient regulation of indoor heating and cooling systems. When it comes to indoor temperature, distributed IoT devices in buildings can enable more accurate temperature forecasting and eventually help to build administrators in regulating the temperature in an energy-efficient way, but without damaging the indoor environment quality. While the IoT technology shows potential as a complement to HVAC control systems, the majority of existing IoT systems integrate a remote cloud to transfer and process all data from IoT sensors. Instead, the proposed IoT system incorporates the concept of edge computing by utilizing small computer power in close proximity to sensors where the data are generated, to overcome problems of the traditional cloud-centric IoT architecture. In addition, as the microcontroller at the edge supports computing, the machine learning-based prediction of indoor temperature is performed on the microcomputer and transferred to the cloud for further processing. The machine learning algorithm used for prediction, ANN (Artificial Neural Network) is evaluated based on error metrics and compared with simple prediction models.

# CHAPTER 1. INTRODUCTION

The advances in sensor and communication technologies have enabled things to connect at any time from any place. The connectivity made possible through the breakthrough is well-known as the Internet of Things (IoT) (Gubbi et al., 2013), and has been integrated into our lives in a variety of ways such as smart thermostats, voice assistants for Smart Home (e.g., Alexa and Siri), wearable devices, etc. Supported by these IoT devices and sensors, smart buildings are becoming a reality facilitated by IoT-integrated building management systems. Smart (mobile) things equipped with sensors, internet connection, and automatic identification (such as RFID) enable things to have the intelligence to collect data with much more details about the real world. (Moreno et al., 2014).

While both commercial and industrial buildings take up a tremendous amount of global energy use, recent findings claim that the IoT shows promising potential to empower the better management of building resources and energy consumption. Having a sensor network in place, where internet-connected things can share the temperature information, can improve the spatial resolution of the information for the centralized heating system (Monteiro et al., 2018). In addition, the positive impact of edge computing on energy consumed in IoT devices has been highlighted through calculating how much even a simple decision made utilizing the edge network resources can prolong the battery life of internet-connected devices significantly (Mocnej et al., 2018).

To achieve an effective energy management strategy in buildings, an accurate indoor temperature model is essential (Afroz et al., 2017). By providing future boundary conditions and targets, indoor temperature predictions can contribute to achieving an optimum amount of consumed power in buildings. In addition, the predictions can be used in predictive control systems and training for future scenarios. Machine learning, part of Artificial Intelligence and Statistical Inference, has the capability of generating forecasts from the sensor. The study will utilize the benefit of the machine learning methodologies in order to produce indoor temperature forecasts.

## 1.1 Problem Statement

This research is focused on the building of an IoT system, adopting the edge computing paradigm and machine learning technologies for indoor temperature forecasting. Hong &

Varghese (2019) defined edge computing as a computing model that makes use of resources located at the edge of the network. Edge computing can complement a lot of existing IoT solutions with a centralized architecture – they consist of distributed monitoring nodes and a central data center to which all the data are sent. In the traditional IoT architecture, the big data from the distributed IoT devices are transferred to the remote cloud, via the Internet (Truong & Dustdar, 2015). In spite of the Cloud-centric Internet of Things (CIoT) being a common way to implement IoT systems, CIoT-based systems are struggling with new problems arising in IoT such as bandwidth, latency, uninterrupted, resource-constraint, and security (Chiang & Zhang, 2016).

While the edge computing paradigm has the potential to alleviate the constraints of existing IoT systems, its plausibility for smart building temperature prediction has not been explored by a lot of researchers. Artificial Neural Networks (ANNs) have been widely considered superior to the other machine learning techniques when it comes to indoor temperature forecasting, however, their accuracy and feasibility have not been evaluated in IoT scenarios where resources at the edge and different building parameters are utilized. This paper aims to tackle such a research gap and provide a detailed assessment of its prediction accuracy in an IoT context.

## 1.2 Significance

The information on energy consumption in buildings has become significant over the years due to the rapidly increasing global energy depletion. Concerns over draining energy, shortage of supply, and negative impact on environmental factors (ozone depletion, rising temperature, extreme weather, etc.) have been raised after the sixth assessment of climate science was released in August by the Intergovernmental Panel on Climate Change. According to the report (Lee et al., 2021), even if nations take immediate steps today to sharply reduce emissions, so much carbon dioxide, and other greenhouse gases have been put into the atmosphere that global warming will continue at least until the middle of the century. Globally, the energy consumption at the commercial and residential buildings shares a one-third amount of the energy demand which is partially utilized by HVAC systems. For addressing the concerns of global climate change and reduction of energy wastage, household energy conservation is more important than ever. It can be concluded that temperature information and prediction can be used for more precise energy management to tackle these issues.

On the other hand, the conventional IoT design, in which big sensor data are directly transmitted from the deployed IoT devices to the remote data center, is facing the difficulties of the higher cost for the data transfer and huge consumption of bandwidth, energy, and time, especially in the wake of the proliferation of the Internet of Things devices (Sun & Ansari, 2016). Machine-To-Machine (M2M) connections, which are also called IoT in Cisco Annual Internet Report (2018–2023), will show the most rapid growth in the device and connections group, increasing almost 2.4-fold to 14.7 billion connections by 2023 (Cisco, 2020). The employment of a cloud server for the management of countless Internet-connected objects and the data created by them may introduce network congestion and latency issues, which will eventually hinder timely data analytics and lower energy efficiency from being performed in the cloud.

## 1.3 Purpose Statement

In this paper, an edge computing-based IoT system for indoor temperature forecasting will be presented. The purpose of this study is to examine the prediction accuracy of the presented system can achieve.

## 1.4 Research Questions

The research questions that will be addressed in this study include:

(1) Can an edge device produce indoor temperature forecasts?
(2) Can the ANN prediction model achieve improved accuracy compared to the other models?

## 1.5 Assumptions

The assumptions of this study include:

- It is assumed that there is no power outage and wireless connection is secured during the experiment.

## 1.6 Delimitations

The delimitations of this study include:

- The evaluation of the algorithm used in this study will be based on the data collected in a residential building in Spain, which is retrieved from the UCI machine learning repository.
- The most relevant three features are used for the actual prediction.
- Missing data points were filled in through linear interpolation.

## 1.7 Limitations

The limitations of this study include:

- This study uses a public dataset available on the UCI machine learning repository.
- There are missing data points in the dataset.

## 1.8 Summary

In this chapter, an introduction of the study was provided, including the problem statement, significance, research questions, assumptions, delimitations, and limitations.

# CHAPTER 2. REVIEW OF LITERATURE

This chapter presents a review of the literature relevant to this study. The subsections of the chapter are organized as follows: first, the definition, advantages, and characteristics of edge computing are defined. They are followed by related works on edge computing for environmental monitoring, smart city, and smart buildings. Lastly, studies on IoT-based indoor temperature prediction were presented.

## 2.1 Edge Computing

Access to remote computing resources provided by cloud data centers has become the standard model for most Internet-based applications (Hong & Varghese, 2019). The prevalent infrastructure for cloud applications depends on a single data center to process and store the data generated from end-user devices such as smartphones and wearable devices, or sensors in smart cities. The cloud services using data centers of a single provider have a uniform architecture and provide obvious advantages enabling the cloud to act as a computational and data processing platform as well as data storage (Varghese & Buyya, 2018; Truong & Dustdar, 2015; Wang et al., 2019). However, this computing model for cloud applications is not practical for the future, considering the exploding number of things connected via the Internet (Hong & Varghese, 2019). With billions of IoT devices, the increase in communication latencies will have an adverse impact on the general quality of service provision for end-user applications (Hong et al., 2018).

The efforts to push computing and storage resources to the Internet's edge, close to the sensors and mobile devices led to the design of an alternative architecture to traditional cloud computing. This alternative computing model was known to mitigate the mentioned problem by bringing computing power closer to end-users and offloading data processing (even if only partial) performed in the cloud to the edge. To realize the model, recent studies aim to decentralize some of the computing resources available in large data centers by distributing them towards the edge of the network as described in Figure 1 (Hong & Varghese, 2019). An edge computing model makes use of resources available at the edge, unlike existing cloud computing models. (Satyanarayanan, 2017; Shi et al, 2016). Moreover, the term "fog computing" has been used interchangeably with edge computing. Here, fog computing is defined as a model that makes use of both edge resources and the cloud (Bonomi et al., 2012; Buyya & Srirama, 2019; Varghese & Buyya, 2018).

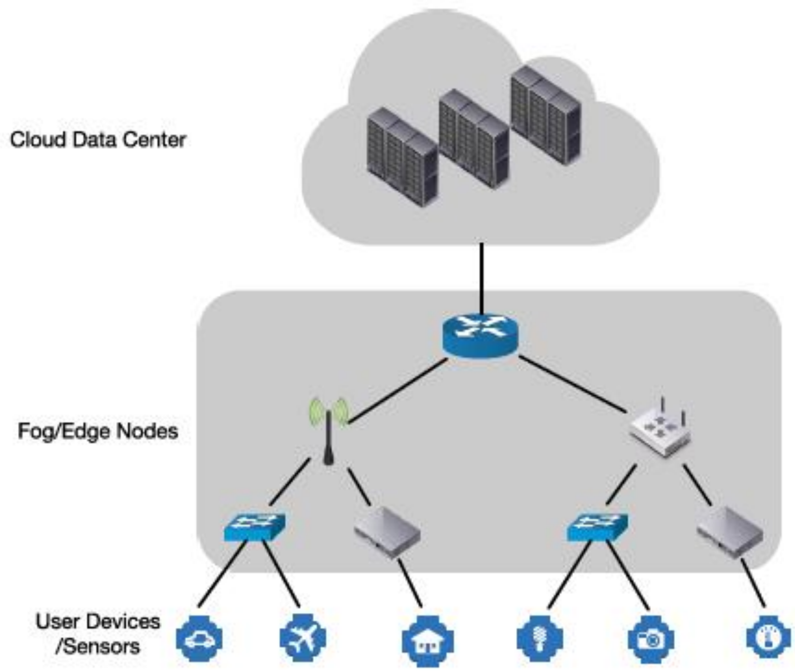


Figure 2.1. An architecture based on edge computing. It consists of the cloud, fog/edge nodes comprising the network at the edge, and user devices/sensors (Hong & Varghese, 2019)

In IoT, edge computing can provide a complement to the cloud by filling the gap between the cloud and IoT devices toward offering an uninterrupted service (Chiang & Zhang, 2016; Varghese & Buyya, 2018). Integrating edge computing into IoT architectures enables applications to be scaled across different computing tiers. Workloads can be offloaded from cloud data centers to edge nodes, or from user devices to edge nodes allowing computation to be performed in close proximity to the data source, rather than geographically remote data centers (Varghese & Buyya, 2018). Tapping into computing power on the edge that wasn't traditionally utilized in IoT can contribute to mitigating possible network congestion through data aggregation for pruning or filtering (Rajagopalan & Varshney, 2006), preventing unnecessary data from being transmitted beyond the edge of the network. Accordingly, the IoT architectures employing edge computing will work in coordination with the cloud services instead of rendering them obsolete.

### 2.1.1 Edge Computing and Environmental Monitoring

In the recent few years, edge and fog computing paradigms have attracted wide research interests; especially with regard to environmental monitoring and warning systems. In addition to the cloud services, data processing for large volumes of environmental monitoring data generated by sensors can be further facilitated by the offloading mechanism and decentralized architecture in which edge devices are located close to the IoT sensors. As much as a lot of studies on environmental monitoring leveraging the IoT Technology are being conducted, the term “environment” used in environmental monitoring edge computing-based IoT systems can imply various meanings in different contexts. For example, a fog-enabled Industrial Internet of Things (IIoT) system and IoT data scheduling for hierarchical fog computing were devised incorporating computing power at the edge in the industrial environment (Aazam et al., 2018; Chekired et al., 2018). On the other hand, Sittón-Candanedo et al. (2019) used the term and conducted their research in the context of smart buildings and Mendiboure et al. (2019) for vehicular environments. The existing studies on fog/edge computing-based IoT systems for Smart City will be explored in the next subsection.

There are recent studies that aimed to realize the vision of IoT for continuous environmental monitoring in a real-time manner. Durresi et al. (2018) proposed a communication design in which the same smart device moves across multiple wireless sensor networks (WSNs) and interacts securely. To utilize the advantages of both smartphones and sensor nodes, the gateway nodes act as base stations to sensor nodes and aggregate the information gathered by the sensor network, as well as intermediates through which smartphones obtain information from sensor networks. All the gateways are connected to an Authentication Server (AS), which is in charge of every task regarding security; as smartphones are supposed to intercommunicate across the networks, a Central Server (CS) is included in the design and used to provide the secure communication links between sensor networks and AS’s. In this study, CS functionalities are installed in micro-clouds of Multi-access edge cloud computing (MEC). The advances of edge cloud computing have led to the invention of Multi-access edge cloud computing (MEC), which enables the delivery of applications with high bandwidth requirements (e.g., video streaming, online gaming, augmented reality, etc.) to the mobile users who rely on wireless networks such as WiFi and 5G (Shahzadi et al., 2017; Taleb et al., 2017).

In 2020, Gao et al. presented a distributed inference framework named Federated Region-Learning (FRL) for continuous environmental monitoring in urban areas, pointing out the infrequent sensor data due to scarce sites and records which consequently impedes detailed environmental sensing. Unlike the existing centralized training models visualized in Figure 2.2, FRL is based on federated learning where the training data are assigned on mobile devices or designated locations through the aggregation of recent changes. FRL further considers the local attributes such as sites' location and the arrangement of training data. Through the presented edge computing-based approach, a local model in a region is designed for each micro cloud of monitoring sites to improve the accuracy. The entire architecture of FRL is visualized in Figure 2.3. Moreover, the strategies for synchronous and asynchronous global model aggregation were proposed to tackle different bandwidth needs. In the synchronous FRL scenario, the central server waits for all micro clouds to transfer their local models and then performs a global model aggregation, which is more fit for regions with less bandwidth. In contrast, the central server conducts a global model aggregation at the time of receiving a local model of the micro cloud that completes the training first, which is suitable for micro clouds with more relaxed bandwidth requirements and uneven sample distribution. By means of the comparison between the centralized training mode and FRL Gao et al. prove the improved computational efficiency and accuracy.

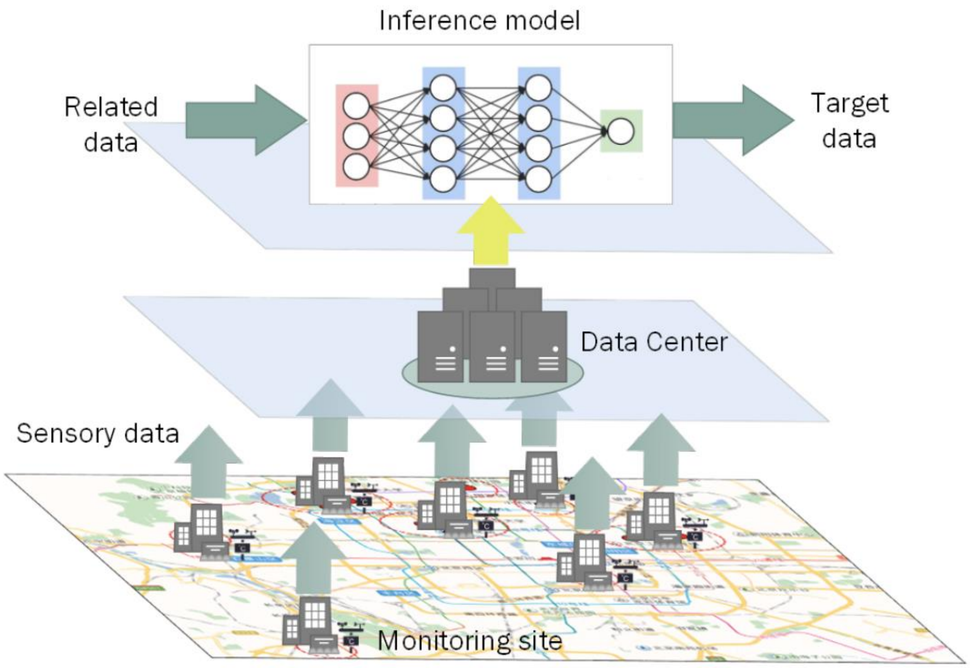


Figure 2.2. Centralized training mode for environmental monitoring (Gao et al., 2020)

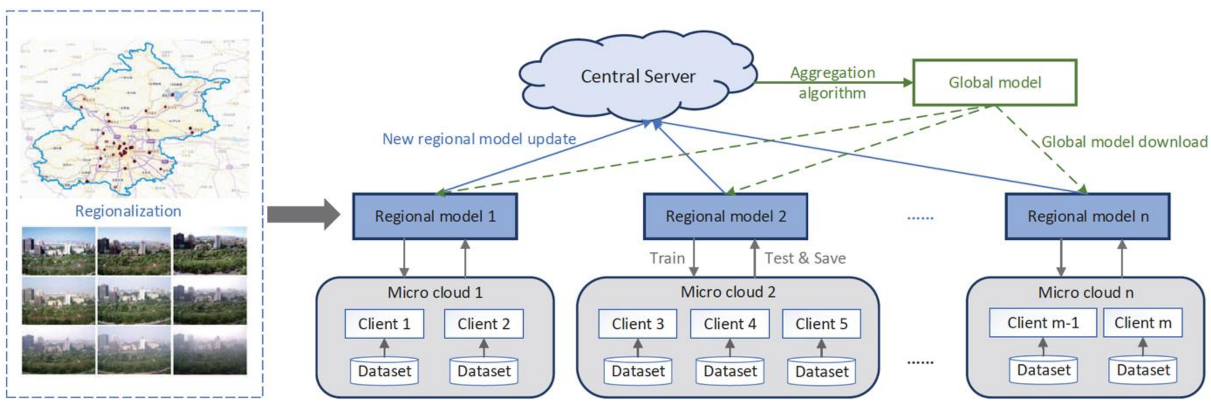


Figure 2.3. Architecture of Federated Region-Learning (Gao et al., 2020)

Along with the studies on the architecture and algorithm employing the distributed approach of edge computing for general environmental monitoring, some studies address specific environmental incidents or continuous monitoring for certain environmental parameters. Avgeris et al. (2019) pointed out the direct impacts that climate change has on weather conditions in Europe and how they intensify the frequency of unexpected forest fires and the importance of detection of a forest fire for in-time firefighting. In addition to their great scalability and low-cost operational expenditures, IoT monitoring nodes deployed in remote locations have enabled the detection of wildfires, and their ability to perform basic data processing and exchange information make IoT networks suitable for continuous monitoring in large forest areas. Through the utilization of computing power at the edge of networks and social media data for the realization of the participatory data gathering paradigm, they proposed a novel Cyber-Physical Social System (CPSS) for fast fire detection that offloads computation-intensive tasks to the edge servers.

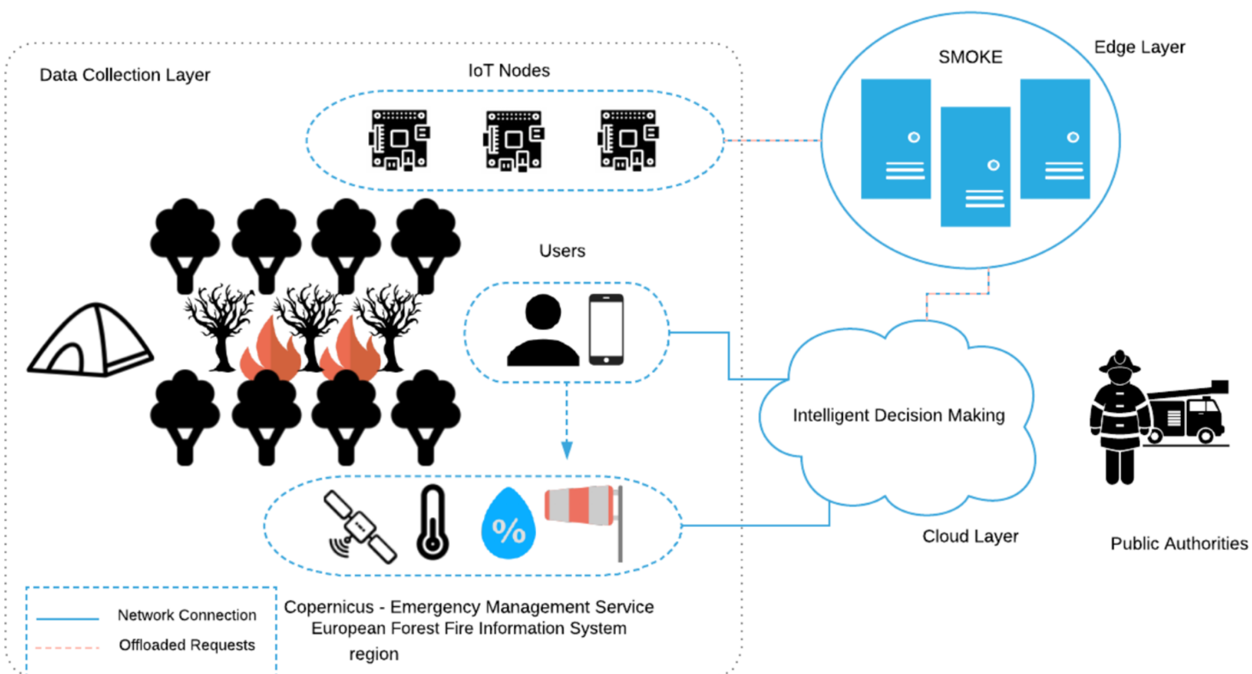


Figure 2.4. Architecture of the proposed CPSS. (Avgeris et al., 2019)

In the first data collection layer, camera modules are embedded in the IoT nodes, and they enable computer vision-based fire detection. The SMOKE framework in the middle layer is a dynamic resource scaling mechanism for IoT applications and assists the offloading process for computationally intensive tasks. The cloud layer supports a decision-making process by combining the classification results, users' social media data, and weather information, in order to determine the seriousness of fire incidents and create alerts for the responsible authorities.

Another study on IoT systems for monitoring and detection of environmental incidents introduced an edge computing-based sensor network for water level forecasts in real-time, named ECOMSNet (Yang et al., 2020). The proposed network system was designed for early warning systems (EWSs) that monitor in-place and real-time water level information and produce forecasts with the collected information. Different from the most common centralized EWS framework, the system adopts the decentralized framework 'edge computing' to achieve shorter processing and response time, allowing simultaneous data collection and processing which leads to decision-supporting information with minimum delay. Initially, with a basic algorithm for data quality management, the ECOMSNet produces water level predictions using different parameters such as water discharge, channel bottom slope, and roughness coefficient. The system predicts water levels using ultrasonic sensors on the edge where the computing power is limited, and it is inessential to send the observed data to a central computing device to further process the data. Three real-time correction methods were established and embedded in ECOMSNet for the sake of improved accuracy. The water level forecasts can be supplied to the parties responsible for flood warnings or other applications.

### 2.1.2 Edge Computing and Smart City

The unprecedented developments in IoT and sensing technologies have shown that the vision of smart cities is not far from reaching. The definition of smart cities varies depending on researchers (Eckhoff & Wagner, 2018; Ianuale et al., 2016; Albino et al., 2015), thus there is no absolute definition of smart cities that has been universally agreed upon. The general definitions of smart cities mean advances in ICT infrastructure, which make it possible for residents to have a better quality of life via intelligent systems (Khan et al., 2020). Recently, the cloud computing paradigm has emerged and offered nearly limitless resources as a promising solution to the high computing complexity of implementing smart cities (Siow et al., 2018). However, because of its structural limitations of high latency, non-context-aware characteristics, and lack of support for mobility, the centralized cloud computing models for IoT fail to satisfy the requirements of real-time smart environments. In addition, cloud computing suffers from processing time inefficiency due to the large overhead of smart city data (Khan et al., 2020). Hence, these limitations called for a new paradigm that expands the computing resources to the edge of a network and offers context-awareness, less throughput, mobility support, and scalability for implementing real-time smart city designs (Shi et al., 2016; Ahmed et al., 2017; Ji et al., 2019). Figure 2.5 presents a high-level design of the IoT-based smart city enabled through the integration of edge computing.

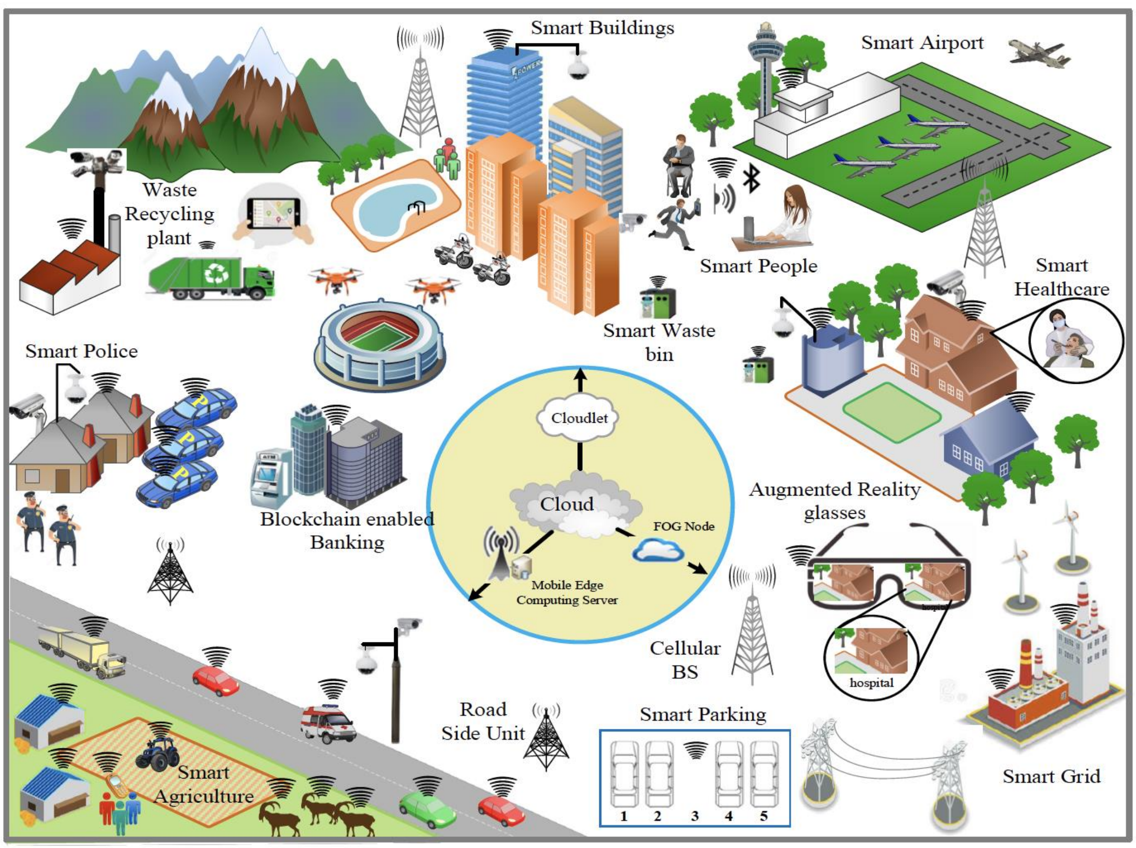


Figure 2.5. An overview of a smart city advanced by edge computing (Khan et al., 2020)

### 2.1.3 Edge Computing and Smart Building

Ferrández-Pastor et al. proposed an edge computing-based system that incorporates Internet-connected things with SoC (System on a Chip) capacity to provide the groundwork for designing smart buildings. A system on a chip, known as an SoC is an integrated circuit that encompasses components of a computer. The authors considered the integration of different systems, including electricity consumption detection, indoor environment control, security, residents' comfort, or operating costs, to develop smart facilities. The proposed model was tested in a residential home, utilizing pattern recognition and decision tree methods. Through the experiment, they proved that their approach overcomes the drawbacks of existing solutions in terms of interoperability and scalability of services.

In their work in 2014, Moreno et al. analyzed how energy is currently consumed in buildings with the aim to promote energy sustainability of the planet. Buildings are required to meet higher energy demand and energy efficiency requirements nowadays to reduce the amount of global energy consumption. Their analysis aims to help building managers figure out the most relevant parameters as input data of building control systems. The automated energy monitoring system was deployed in three reference smart buildings. Through the use cases presented by the study, the percentage of saved energy with their energy management proposal was proved to be approximately 23%.

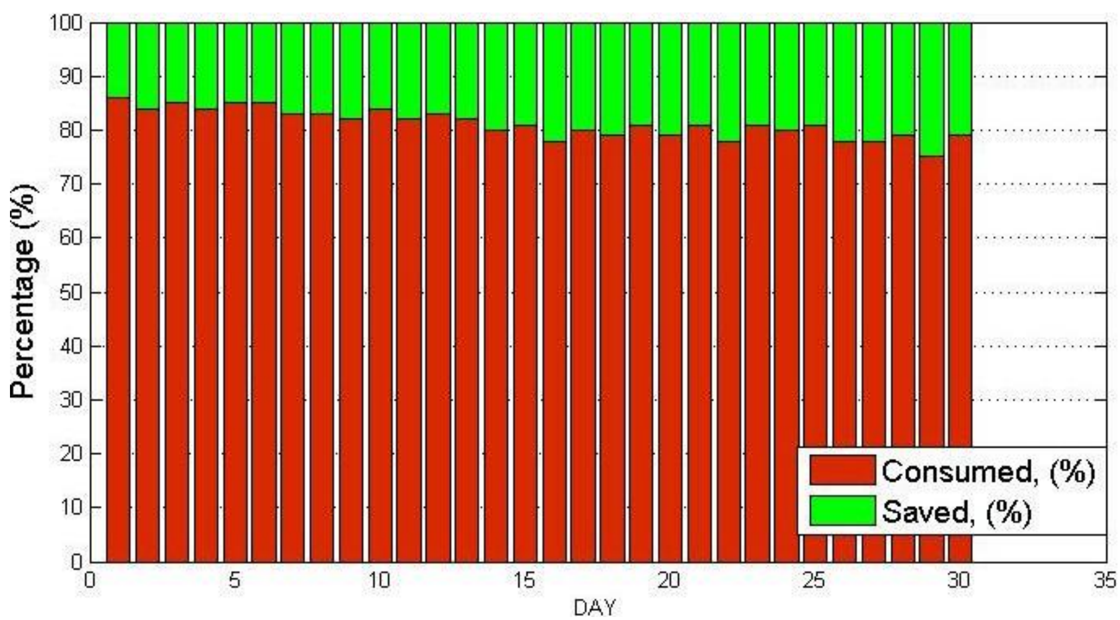


Figure 2.6. Percentage of energy saved in heating in Use Case 2 (Moreno et al., 2014)

## 2.2 IoT-based Indoor Temperature Prediction

Paul et al.'s paper published in 2018 highlighted the increasing demand for energy consumption management in the building sector. The main challenge in building energy management lies in analyzing and predicting the dynamical behavior of the building system (Nicol & Humphreys, 2002). While complex energy simulation tools are deployed to provide building administrators a general assessment of the building energy consumption, their accuracy metrics highly rely on the building parameters of the dataset used. Their work is focused on the comparison of three different supervised machine learning methods, which are used for smart building indoor temperature prediction. The following figures display the real-time and predicted values of each prediction model.

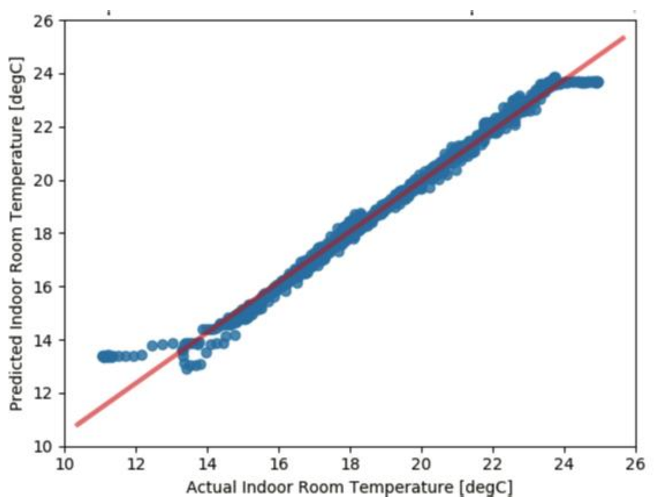


Figure 2.7. Predicted vs actual results for Random Forest (Paul et al., 2018)

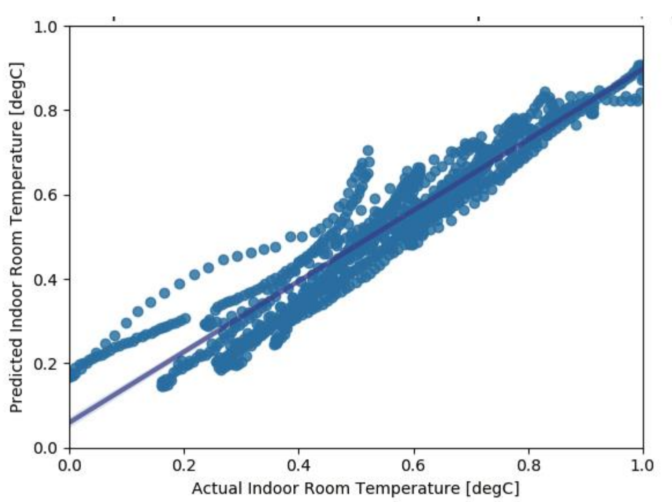


Figure 2.8. Predicted vs actual results for SVM method (Paul et al., 2018)

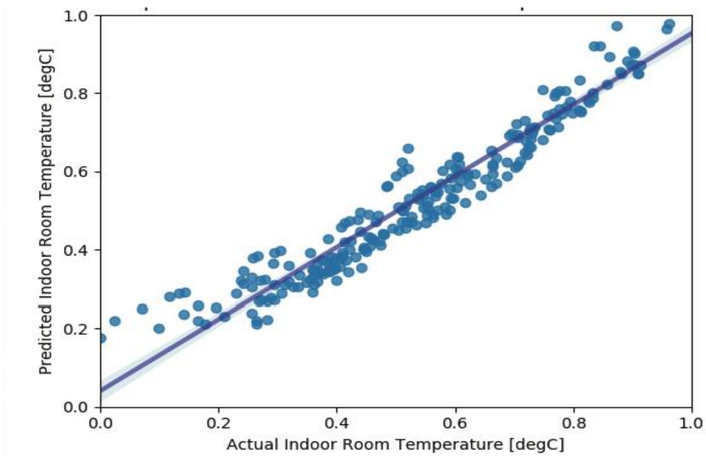


Figure 2.9. Predicted vs actual results for Neural Networks (Paul et al., 2018)

In their research in 2018, Monteiro et al. explained that the integration of the IoT can benefit indoor temperature forecasting systems by connecting smart devices and sharing sensor data over a local network. They claimed that fewer research efforts have been devoted to an IoT context despite the advantages of utilizing environment parameters, for which IoT sensors gather the information. In their IoT scenario, various machine learning models have been considered and the Mean Squared Error (MSE) metric was used to assess each method. Figure 2.10. displays the errors obtained from each model when applied to the data collected from the internal sensor of a refrigerator.

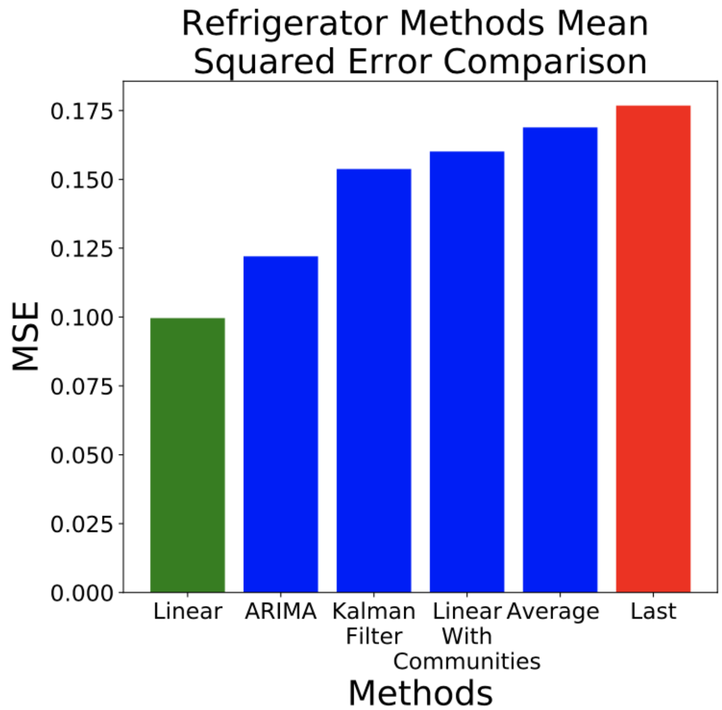


Figure 2.10. MSEs by the forecasting models based on the refrigerator's temperature data. The color of the bars represents the accuracy of each model, green being the best and red being the worst. (Monteiro et al., 2018)

The results indicate that the refrigerator can use machine learning-based prediction models to generate temperature forecasts. The best algorithm performed the temperature prediction within two and half hours with only the error of ≈ 0.09 ºC, proving the feasibility of the scenario.

# CHAPTER 3. METHODOLOGY

In this study, an edge computing-based IoT system architecture for indoor temperature forecasts in a smart building will be proposed and tested as a platform for indoor temperature prediction. The system used a set of machine learning-based prediction models, and they were evaluated against two real datasets retrieved from the UCI Machine Learning Repository (Zamora-Martinez et al., 2014). The following sections consist of the description of the datasets, algorithms used for prediction, accuracy metric, and the proposed system architecture.

## 3.1 Description of the Datasets

The datasets used in this research were collected in a residential building in Spain, where a monitor system was mounted and collected multiple building parameters. The residential building participated in the Solar Decathlon 2013 competition (Zamora-Martínez et al., 2013). They will be used for time series forecasting, where a time series is a collection of sequential observations across time (Chatfield, 2001). Table 3.1 contains the entire list of the collected parameters which will be termed as attributes in the datasets.

Table 3.1. Attributes and descriptions of the datasets (Zamora-Martinez et al., 2014)

| **Attributes** | **Units** |
|---|---|
| Date | UTC |
| Time | UTC |
| Indoor temperature (dining room) | ºC |
| Indoor temperature (room) | ºC |
| Weather forecast temperature | ºC |
| Carbon dioxide (dining room) | ppm |
| Carbon dioxide (room) | ppm |
| Relative humidity (dining room) | % |
| Relative humidity (room) | % |
| Lighting (dining room) | Lux |
| Lighting (room) | Lux |
| Rain, the proportion of the last 15 minutes where rain was detected (a value in range [0,1]) | |
| Wind velocity | m/s |
| Sunlight in west façade | Lux |
| Sunlight in east façade | Lux |
| Sunlight in south façade | Lux |
| Sun irradiance | $W/m^2$ |
| Enthalpic motor 1, 0 or 1 (on-off) | |
| Enthalpic motor 2, 0 or 1 (on-off) | |
| Enthalpic motor turbo, 0 or 1 (on-off) | |
| Outdoor temperature | ºC |
| Outdoor relative humidity | % |
| Day of the week, 1=Monday, 7=Sunday | |

Each dataset corresponds to the period from March 13 to April 11 and from April 18 to May 2 in 2012, respectively. The data was sampled with a period of 15 minutes, where each sample equals a mean of the last 15-minute data. The heatmaps in Figure 3.1 and Figure 3.2 display the correlation between all the numeric parameters in Dataset 1 and Dataset 2, respectively.

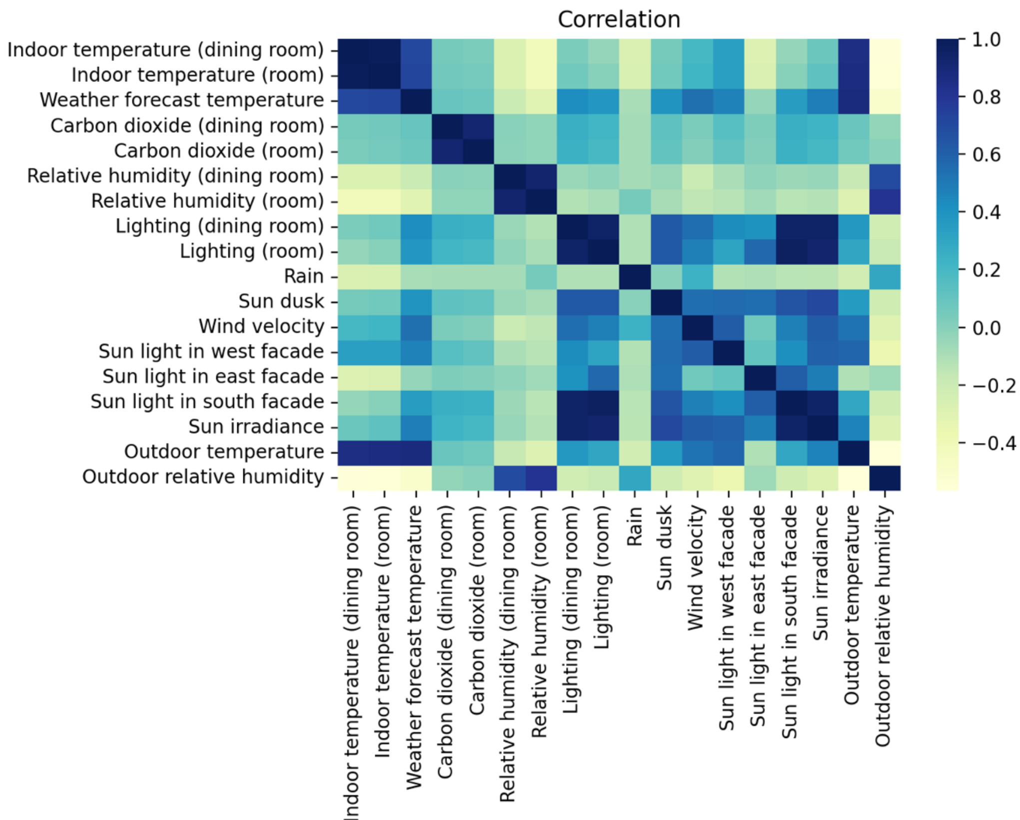


Figure 3.1. Correlation between the numeric parameters in Dataset 1

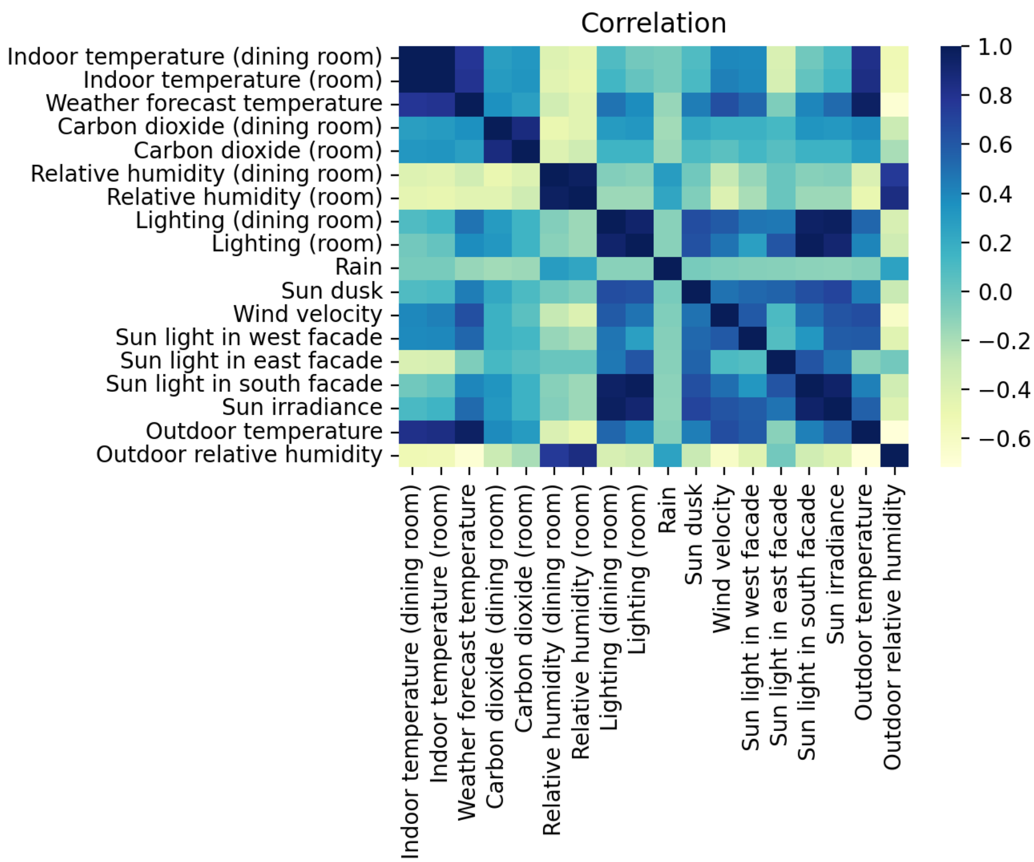


Table 3.2. Correlation between the numeric parameters in Dataset 2

The following tables contain a brief description of the two datasets. Dataset 1 has a total of 2764 data points and Dataset 2 has 1373 data points.

Table 3.3. Statistical summary of Dataset 1

| | **Mean** | **Std** | **Min** | **25%** | **50%** | **75%** | **Max** |
|---|---|---|---|---|---|---|---|
| Indoor temperature (dining room) | 19.1997221 | 2.85331471 | 11.352 | 17.4508 | 19.37365 | 21.229975 | 25.54 |
| Indoor temperature (room) | 18.8248521 | 2.82117789 | 11.076 | 17.06035 | 19.021 | 20.8287 | 24.944 |
| Weather forecast temperature | 13.8973955 | 4.17199073 | 0 | 10.783325 | 15 | 16.6667 | 26 |
| Carbon dioxide (dining room) | 208.479123 | 27.032686 | 187.339 | 200.89325 | 207.0455 | 211.2455 | 594.389 |
| Carbon dioxide (room) | 211.065844 | 28.4691435 | 188.907 | 202.68275 | 209.408 | 213.21875 | 609.237 |
| Relative humidity (dining room) | 44.8784197 | 6.58743994 | 27.084 | 40.351975 | 45.43465 | 49.352675 | 60.9573 |
| Relative humidity (room) | 47.3212199 | 7.55779527 | 29.5947 | 42.531325 | 47.5347 | 52.685975 | 62.5947 |
| Lighting (dining room) | 26.7453813 | 23.2984405 | 10.74 | 11.5887 | 11.8013 | 31.224 | 110.693 |
| Lighting (room) | 40.7325708 | 42.3260865 | 11.328 | 13.2653 | 17.69 | 52.05735 | 162.965 |
| Rain | 0.04703328 | 0.20670498 | 0 | 0 | 0 | 0 | 1 |
| Sun dusk | 325.369289 | 305.062614 | 0.606667 | 0.65 | 611.797 | 619.21075 | 624.96 |
| Wind velocity | 1.10853124 | 1.161283 | 0 | 0.0948333 | 0.659 | 1.9714975 | 6.32133 |
| Sunlight in west facade | 14936.6177 | 25964.0495 | 0 | 0 | 0 | 15088 | 95278.4 |
| Sunlight in east facade | 12248.0001 | 21758.5505 | 0 | 0 | 0 | 11131.275 | 85535.4 |
| Sunlight in south facade | 22047.5258 | 32709.3871 | 0 | 0 | 0 | 38736.575 | 95704.4 |
| Sun irradiance | 215.010017 | 297.234046 | -4.16467 | -3.38133 | 3.922 | 435.4345 | 1028.27 |
| Outdoor temperature | 16.7578468 | 3.88586933 | 9.22333 | 13.662025 | 16.49035 | 19.3978 | 29.908 |
| Outdoor relative humidity | 55.9819884 | 13.0196104 | 22.2607 | 46.430675 | 57.47735 | 65.649325 | 83.8053 |
| Day of week | 3.95443804 | 1.99179884 | 1 | 2 | 4 | 6 | 7 |

Table 3.4. Statistical summary of Dataset 2

| | Mean | Std | Min | 25% | 50% | 75% | Max |
|---|---|---|---|---|---|---|---|
| Indoor temperature (dining room) | 23.0981075 | 2.55210714 | 16.9833 | 21.1827 | 23.192 | 24.964 | 28.924 |
| Indoor temperature (room) | 22.8057664 | 2.53452913 | 16.7973 | 20.9253 | 22.852 | 24.664 | 28.548 |
| Weather forecast temperature | 17.4989079 | 3.73982822 | 9 | 15 | 17 | 20 | 29 |
| Carbon dioxide (dining room) | 202.816621 | 8.30308641 | 189.195 | 197.867 | 203.115 | 206.219 | 278.645 |
| Carbon dioxide (room) | 206.684117 | 10.8628665 | 192.107 | 199.541 | 207.925 | 210.731 | 313.216 |
| Relative humidity (dining room) | 37.3801744 | 5.65367745 | 26.1733 | 32.484 | 35.8413 | 42.9253 | 52.5893 |
| Relative humidity (room) | 38.9593849 | 6.76501465 | 27.256 | 32.5693 | 38.812 | 44.7013 | 52.624 |
| Lighting (dining room) | 33.4491484 | 29.4189663 | 10.838 | 11.5407 | 20.8233 | 51.5733 | 111.797 |
| Lighting (room) | 45.5623596 | 42.9882965 | 13.5093 | 14.9067 | 24.416 | 67.52 | 157.157 |
| Rain | 0.02209274 | 0.1382033 | 0 | 0 | 0 | 0 | 1 |
| Sun dusk | 354.671855 | 302.567396 | 0.606667 | 0.65 | 615.36 | 620.437 | 625.003 |
| Wind velocity | 1.69937811 | 1.25134467 | 0 | 0.614667 | 1.57267 | 2.716 | 5.354 |
| Sunlight in west facade | 14371.7555 | 23932.658 | 0 | 0 | 2802.69 | 13545.5 | 93121.2 |
| Sunlight in east facade | 16220.1373 | 25966.8031 | 0 | 0 | 3024.9 | 17013.4 | 92367.5 |
| Sunlight in south facade | 15447.7853 | 20945.8505 | 0 | 0 | 2609.15 | 26637.7 | 77359.8 |
| Sun irradiance | 266.816322 | 338.532417 | -3.708 | -3.042 | 35.816 | 566.293 | 1094.66 |
| Outdoor temperature | 20.5588378 | 3.91009286 | 11.9333 | 17.6007 | 20.2647 | 23.4653 | 29.8707 |
| Outdoor relative humidity | 47.7551238 | 12.7940452 | 22.2467 | 37.3387 | 46.8453 | 57.872 | 75.8907 |
| Day of week | 3.9788784 | 1.98374605 | 1 | 2 | 4 | 6 | 7 |

### 3.2 Algorithms Used for Prediction

To understand the patterns behind the progression of temperature, machine learning algorithms were adopted to develop prediction models based on historical data. In this subsection, the algorithms used for indoor temperature prediction will be introduced. Considered algorithms include linear regression and neural networks.

#### 3.2.1 Linear Regression

As a straightforward approach, a linear regression model was made based on the assumption that future temperature depends on the past temperature data. A future temperature at a data point t is calculated as in (1) where n is the number of data inputs.

$$T_t = \beta_0 + \beta_1 T_{t-i} + \cdots + \beta_{i-1} T_{t-i} + \varepsilon, \qquad i = 1, \ldots, n, \tag{1}$$

#### 3.2.2 Artificial Neural Network (ANN)

Artificial Neural Networks (ANNs) were inspired by the idea that we need to look at the brain's architecture for building an intelligent machine. Puri et al. defined an ANN as a computational model motivated by networks of biological neurons, wherein neurons compute output values from inputs. ANNs have been widely used in energy systems modeling (Ruano et al., 2006, Ferreira et al., 2012, Zamora-Martínez et al., 2012). Through the ANN approach, relations between input and output values can be constructed, and they consist of an input layer, an output layer, and one or more hidden layers (Khayatian et al., 2016).

Additionally, a baseline model was used to set a basis for the comparison of the other models. The baseline model just returns the current temperature as the prediction assuming that temperature does not change rapidly.

### 3.3 Performance Metrics

In this study, the error metrics used for measuring and comparing the accuracy of the prediction models include mean absolute error (MAE). It is computed as (2) where n is the number of predictions, $y_i$ is a prediction and $x_i$ is a true value.

$$MAE = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} = \frac{\sum_{i=1}^{n}|e_i|}{n} \qquad (2)$$

Another performance metric used is mean squared error (MSE). The MSE is the mean of the squares of the
errors.

$$MSE = \frac{\sum_{i=1}^{n}(y_i - x_i)^2}{n} = \frac{\sum_{i=1}^{n}(e_i)^2}{n} \qquad (3)$$

### 3.4 Edge Computing-based System for Indoor Temperature Forecasting

The proposed system makes use of the three-tier software-defined fog networking architecture introduced in Tomovic et al. (2016): sensor layer, edge computing layer, and application layer. The sensing layer is the foundation of the indoor temperature forecasting system, which is principally responsible for temperature sensing and data transfer. The main units of the layer are groups of indoor temperature monitoring nodes, and each monitoring node is comprised of a microcontroller, a temperature sensor, and a network device. The network device enables the data generated by the monitoring nodes to be transmitted to edge computing modules in the following layer.

The edge computing layer is based on a low-power wide-area network (LPWAN) and provides connectivity to the monitoring nodes through wireless APs. In addition, a single-board microcontroller, Raspberry Pi is included and fulfills the function of the edge computing paradigm, providing limited computing power at the edge of a network. Hereon, the embedded Raspberry Pi will be referred to as an edge computing device (ECD). In this layer, processing for the raw data is performed by the ECD before it is transferred to the IoT cloud.

The roles of the application layer are defined but not limited to data storage, further data processing, information delivery, and provision of user-interactive services. In the proposed system, the pre-processed sensor data transmitted from the edge computing layer is stored in the cloud. The main components of the application layer consist of a cloud service and a user dashboard where data visualizations are provisioned.

## 3.5 Hypotheses

- $H_0$: The ANN model shows better performance than the simple linear regression model in the proposed edge computing system.
- $H_1$: The ANN model does not show better performance than the simple linear regression model in the proposed edge computing system.

# CHAPTER 4. RESULTS

In Chapter 4, the results of the experiment are explained. First, the data processing tasks are included in order. Then, it is followed by the analysis of the performances of the prediction models.

## 4.1 Data Processing

In this paper, the attributes of the data that were selected for prediction will be termed as features, and the following features were chosen to produce the best predictions based on the correlation matrix.

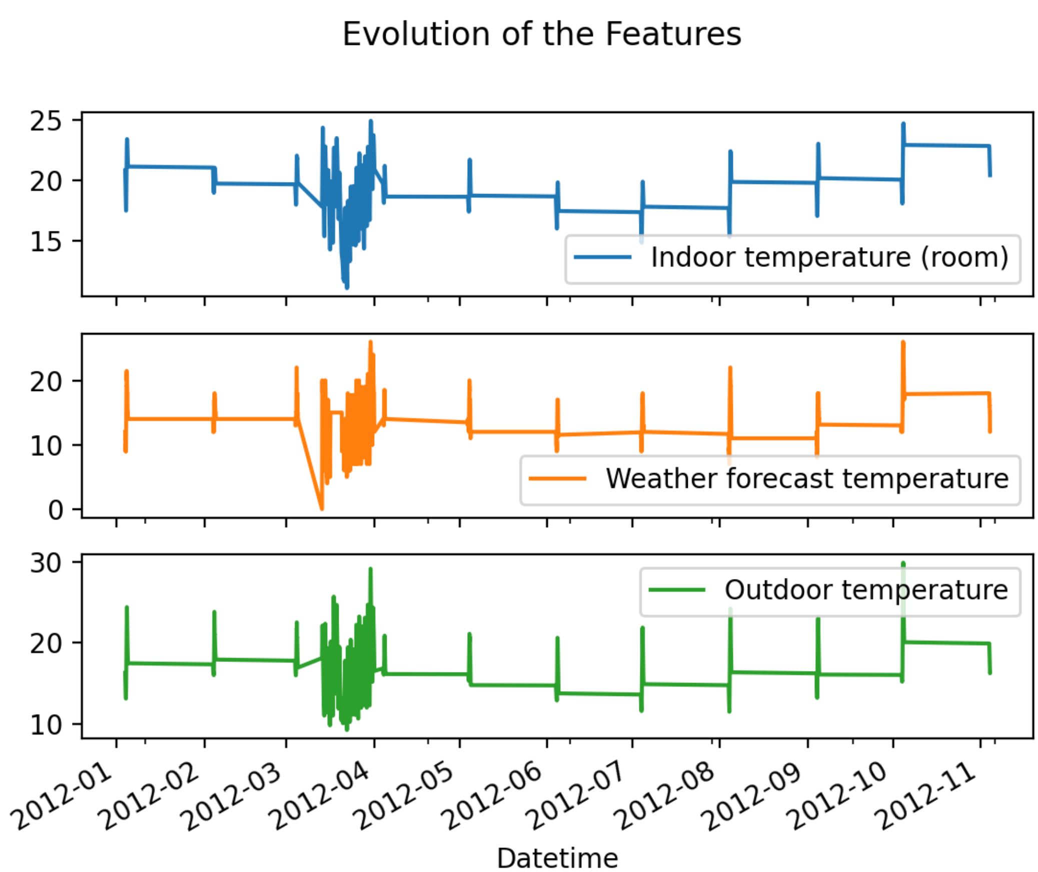


Figure 4.1. Evolution of the selected features in Dataset 1

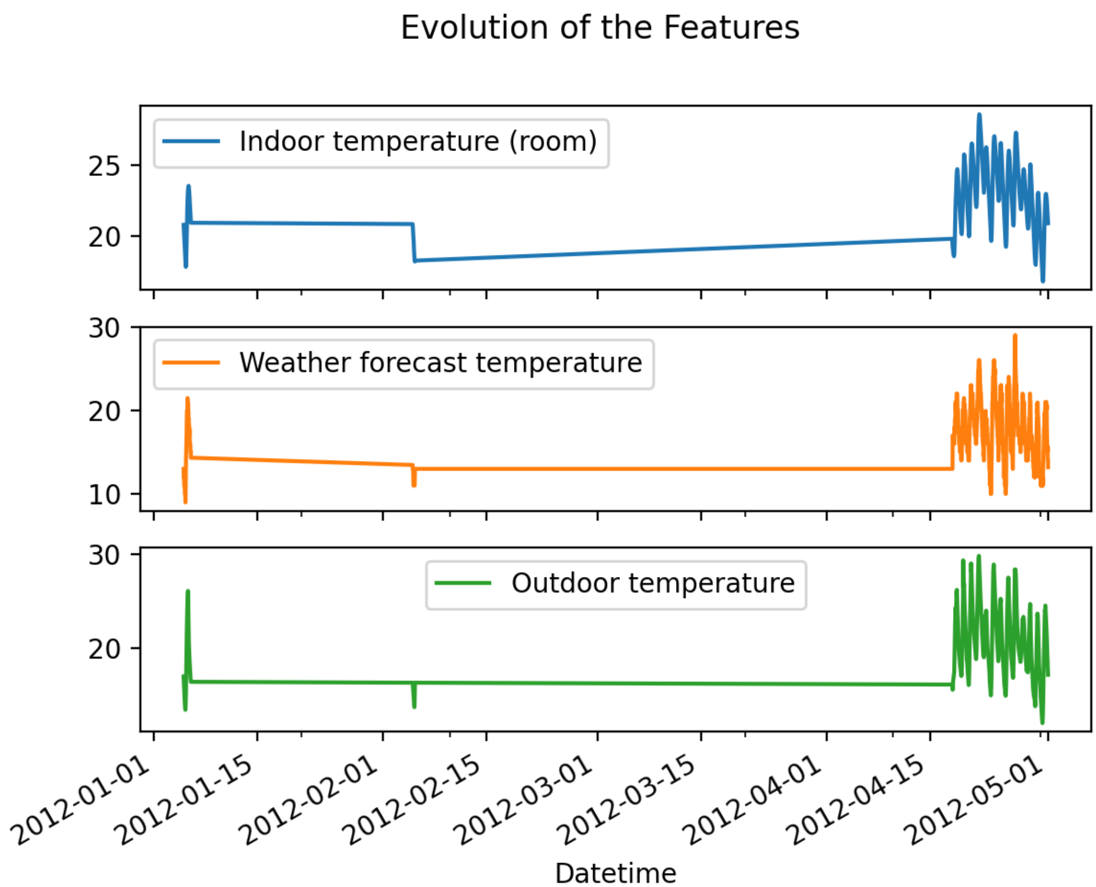


Figure 4.2. Evolution of the selected features in Dataset 2

The data processing tasks in this subsection were implemented using Python and TensorFlow. They were performed on a Raspberry pi 3, which is the edge device in the proposed system.

### 4.1.1 Data Preprocessing

Any missing data points in the datasets were filled through linear interpolation. They were obtained by passing a straight line between two known data points.

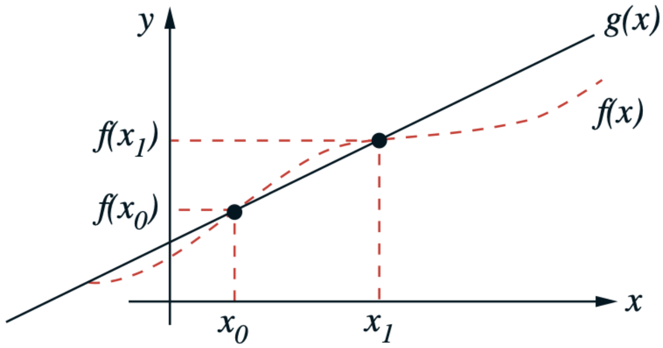


Figure 4.3. Linear interpolation between two known points. Given the data points $x_0$, $x_1$, the straight line is the linear interpolant. (Westerink, 2018)

In addition, the datasets were normalized to scale the features before training the models. Only the training data was used when calculating the mean and standard deviation so that the validation and test sets do not affect the training.

### 4.1.2 Data Windowing

In this study, a prediction produced by the models is made on consecutive samples from the datasets selected by data windows with different sizes. The following figures represent the data windows used in the experiment. The interval between the data points is 15 minutes.

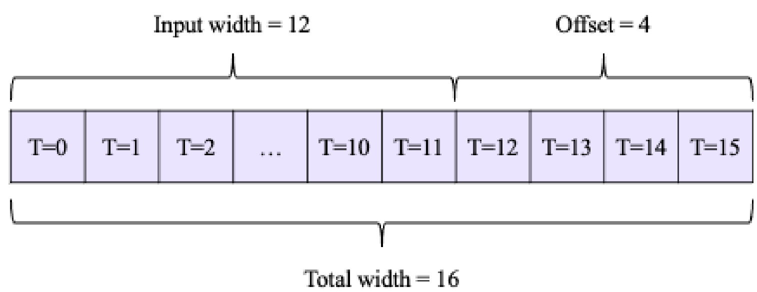


Figure 4.4. Data Window 1 makes a prediction an hour into the future based on 3 hours of history.

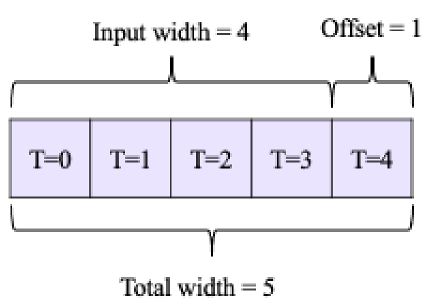


Figure 4.5. Data Window 2 makes a prediction 15 minutes into the future based on an hour of history.

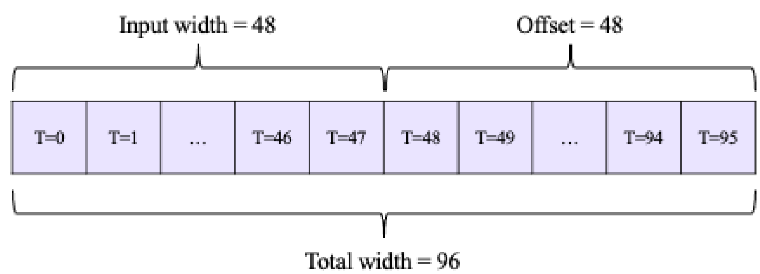


Figure 4.6. Data Window 3 makes a prediction 12 hours into the future based on 12 hours of history.

### 4.1.3 Data Splitting and Training

Data splitting was performed on the two datasets to divide them into the training, validation, and test sets. Before the splitting, the datasets were not randomly shuffled to ensure that the windowed inputs were still consecutive. 70% of the data was used for the training set and the remaining 20% and 10%, for the testing and validation set, respectively. The algorithms worked through the training set 100 times, which was arbitrarily assigned as the maximum number of epochs.

## 4.2 Performance

### 4.2.1 Dataset 1

The error metrics of the prediction models for Dataset 1 are graphed in the below figures.

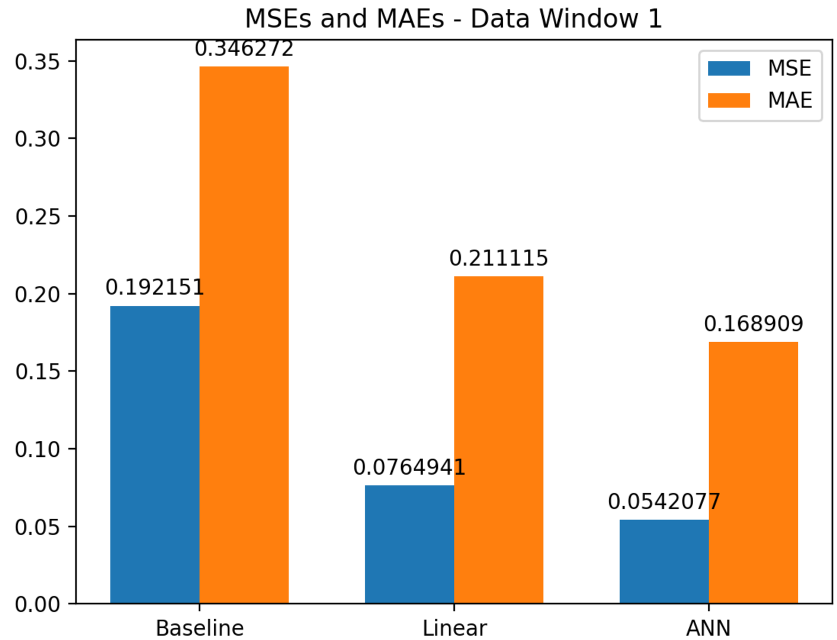


Figure 4.7. The error metrics of the prediction models using Data Window 1 based on MSE and MAE

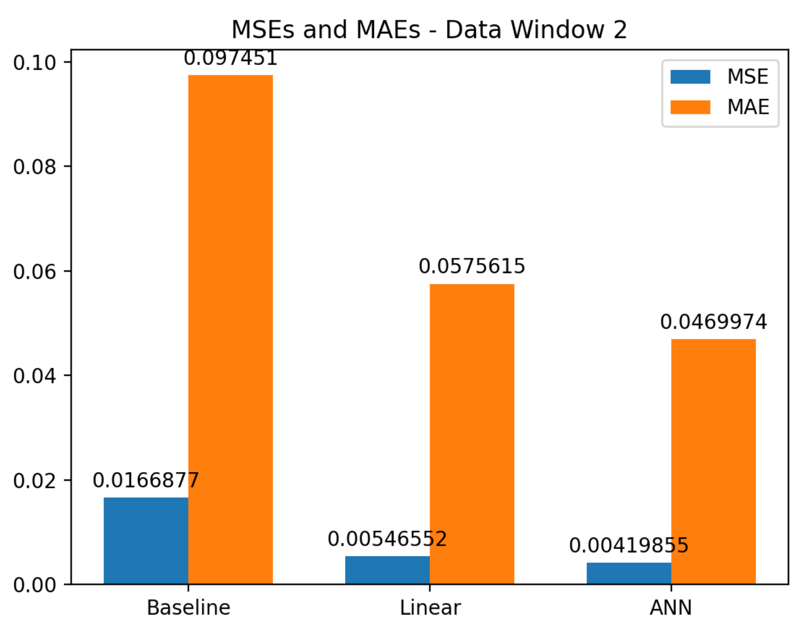


Figure 4.8. The error metrics of the prediction models using Data Window 2 based on MSE and MAE

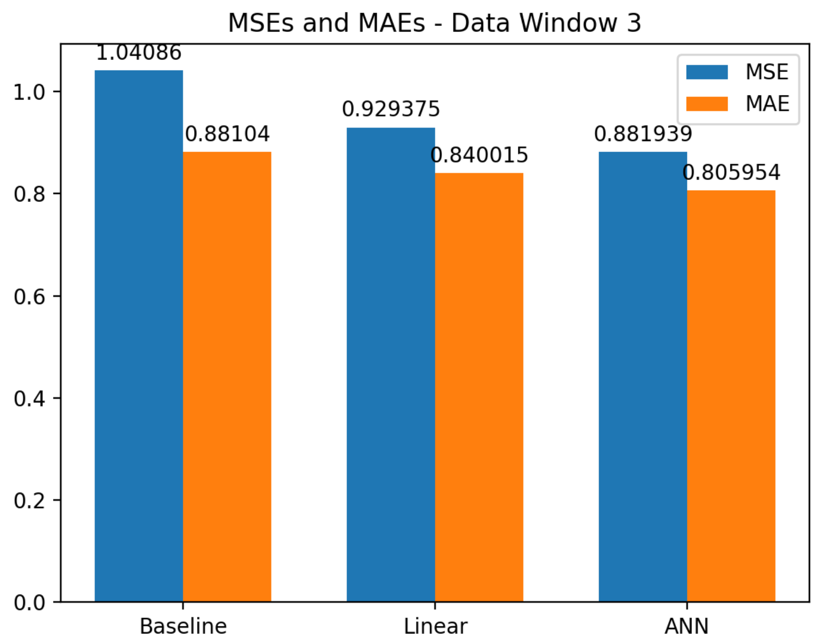


Figure 4.9. The error metrics of the prediction models using Data Window 3 based on MSE and MAE

Overall, the ANN model showed the smallest error in all the cases where predictions are generated every 15 minutes, an hour, and 24 hours. The obtained MSEs and MAEs follow similar performance trends and do not exceed the error of 1.0 ºC in most cases, except the MSE of the baseline model. The wider the data window got, the more error in predictions increased. While all the models displayed the best performance when producing predictions every 15 minutes based on an hour of temperature data, the ANN model was able to reduce the MAE value down to 0.047 ºC, which is half the error of the simple baseline model. The linear regression model displayed the second-best performance in most cases.

### 4.2.2 Dataset 2

The error metrics of the prediction models for Dataset 2 are graphed in the below figures.

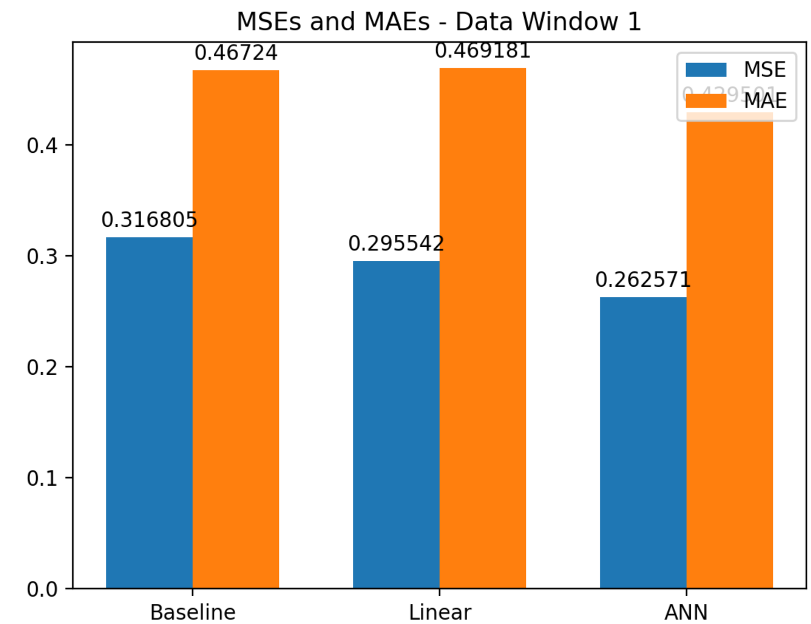


Figure 4.10. The error metrics of the prediction models using Data Window 1 based on MSE and MAE

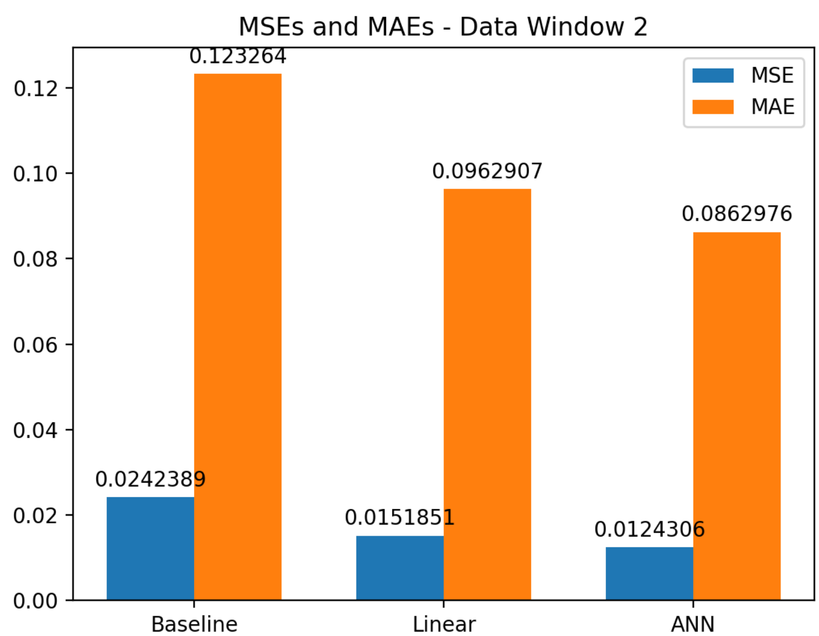


Figure 4.11. The error metrics of the prediction models using Data Window 2 based on MSE and MAE

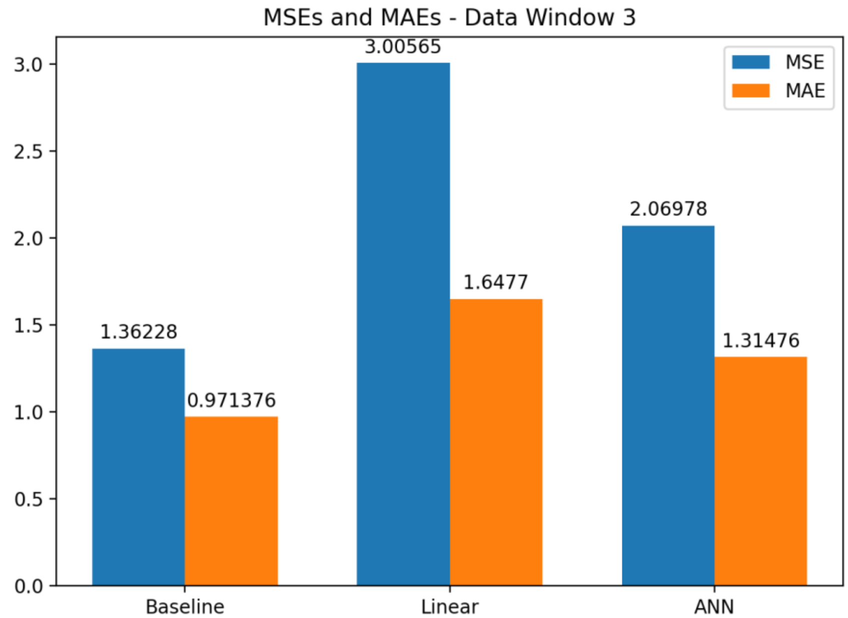


Figure 4.12. The error metrics of the prediction models using Data Window 3 based on MSE and MAE

Overall, the models performed better with fewer errors as they did in Dataset 1. One thing that stands out in Dataset 2 is that the linear regression model and the ANN model do not always produce a better performance than the baseline model. The baseline model produced the least error when tested with Data Window 3. While the baseline model showed a stable prediction capability for long-term forecasting, it can be drawn that the linear regression model and the ANN model were significantly affected by the length of the prediction intervals. Other than that case, the smallest errors were achieved by the ANN model, as small as ≈ 0.086 ºC of MAE.

# CHAPTER 5. CONCLUSION

In this study, an edge computing-based system for indoor temperature forecasting is presented, and machine learning-based prediction models are tested, and their errors were measured using mean squared error (MSE) and mean absolute error (MAE). As much as ANNs have been considered superior compared to the rest of considered models, in this study, the ANN prediction model excelled under different data constraints in most cases. However, as it was mentioned in the previous chapter, the ANN model showed conflicting performances when applied to Dataset 2, whose behavior is presumably similar to Dataset 1's. More careful inspection into the behavioral difference between Dataset 1 and Dataset 2 is needed to discover the root cause behind the conflicting results. The lowest MAE it was able to achieve was ≈ 0.047ºC, and the lowest MSE was ≈ 0.004ºC. The ANN model always indicated lower errors than the linear regression model in all the cases, proving the hypothesis to be true. Based on this, a conclusion can be drawn that ANNs are a suitable model for producing indoor temperature forecasts close to real-time.

As each data point in the datasets was smoothed and collected in a 15-minute interval, 15-minute termed predictions were as the most frequent as the experiment could achieve. The temporal resolution of the experiment can be even more improved by reducing the interval between data points and producing predictions more frequently.

It also stands out that all the prediction models for producing short-term forecasts outperform the models for long-term forecasts. More efforts would be needed to improve long-term forecasts in the future. For future work, a prototype of the presented IoT system will be implemented and build more prediction models utilizing different machine learning technologies.

# REFERENCES


Aazam, M., Zeadally, S., & Harras, K. A. (2018). Deploying Fog Computing in Industrial Internet of Things and Industry 4.0. *IEEE Transactions on Industrial Informatics*, *14*(10), 4674–4682. https://doi.org/10.1109/tii.2018.2855198

Afroz, Z., Shafiullah, G. M., Urmee, T., & Higgins, G. (2017). Prediction of indoor temperature in an institutional building. *Energy Procedia*, *142*, 1860–1866. https://doi.org/10.1016/j.egypro.2017.12.576

Ahmed, E., Ahmed, A., Yaqoob, I., Shuja, J., Gani, A., Imran, M., & Shoaib, M. (2017). Bringing Computation Closer toward the User Network: Is Edge Computing the Solution? *IEEE Communications Magazine*, *55*(11), 138–144. https://doi.org/10.1109/mcom.2017.1700120

Albino, V., Berardi, U., & Dangelico, R. M. (2015). Smart Cities: Definitions, Dimensions, Performance, and Initiatives. *Journal of Urban Technology*, *22*(1), 3–21. https://doi.org/10.1080/10630732.2014.942092

Avgeris, M., Spatharakis, D., Dechouniotis, D., Kalatzis, N., Roussaki, I., & Papavassiliou, S. (2019). Where There Is Fire There Is SMOKE: A Scalable Edge Computing Framework for Early Fire Detection. *Sensors*, *19*(3), 639. https://doi.org/10.3390/s19030639

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing - MCC '12*, 13–16. https://doi.org/10.1145/2342509.2342513

Buyya, R., & Srirama, S. N. (2019). *Fog and edge computing: principles and paradigms*. Wiley.

Chatfield, C. (2001). *Time-series forecasting*. Chapman & Hall etc.

Chekired, D. A., Khoukhi, L., & Mouftah, H. T. (2018). Industrial IoT Data Scheduling Based on Hierarchical Fog Computing: A Key for Enabling Smart Factory. *IEEE Transactions on Industrial Informatics*, *14*(10), 4590–4602. https://doi.org/10.1109/tii.2018.2843802

Chiang, M., & Zhang, T. (2016). Fog and IoT: An Overview of Research Opportunities. *IEEE Internet of Things Journal*, *3*(6), 854–864. https://doi.org/10.1109/jiot.2016.2584538

Cisco. (2020, March 10). *Cisco Annual Internet Report (2018–2023) White Paper*. Cisco Annual Internet Report. Retrieved October 24, 2020, from https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

Durresi, M., Subashi, A., Durresi, A., Barolli, L., & Uchida, K. (2018). Secure communication architecture for internet of things using smartphones and multi-access edge computing in environment monitoring. *Journal of Ambient Intelligence and Humanized Computing*, *10*(4), 1631–1640. https://doi.org/10.1007/s12652-018-0759-6

Eckhoff, D., & Wagner, I. (2018). Privacy in the Smart City—Applications, Technologies, Challenges, and Solutions. *IEEE Communications Surveys & Tutorials*, *20*(1), 489–516. https://doi.org/10.1109/comst.2017.2748998

Environmental Protection Agency. (n.d.). *Sources of Greenhouse Gas Emissions*. EPA. Retrieved October 5, 2021, from https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions.

Ferreira, P. M., Ruano, A. E., Silva, S., & Conceição, E. Z. E. (2012). Neural networks based predictive control for thermal comfort and energy savings in public buildings. *Energy and Buildings*, *55*, 238–251. https://doi.org/10.1016/j.enbuild.2012.08.002

Ferrández-Pastor, F.-J., Mora, H., Jimeno-Morenilla, A., & Volckaert, B. (2018). Deployment of iot edge and fog computing technologies to develop smart building services. *Sustainability*, *10*(11), 3832. https://doi.org/10.3390/su10113832

Gao, Y., Liu, L., Hu, B., Lei, T., & Ma, H. (2020). Federated Region-Learning for Environment Sensing in Edge Computing System. *IEEE Transactions on Network Science and Engineering*, 1–1. https://doi.org/10.1109/tnse.2020.3016035

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010

Hong, C.-H., & Varghese, B. (2019). Resource Management in Fog/Edge Computing. *ACM Computing Surveys*, *52*(5), 1–37. https://doi.org/10.1145/3326066

Hong, C.-H., Lee, K., Kang, M., & Yoo, C. (2018). qCon: QoS-Aware Network Resource Management for Fog Computing. *Sensors*, *18*(10), 3444. https://doi.org/10.3390/s18103444

Ianuale, N., Schiavon, D., & Capobianco, E. (2016). Smart Cities, Big Data, and Communities: Reasoning From the Viewpoint of Attractors. *IEEE Access*, *4*, 41–47. https://doi.org/10.1109/access.2015.2500733

Ji, W., Xu, J., Qiao, H., Zhou, M., & Liang, B. (2019). Visual IoT: Enabling Internet of Things Visualization in Smart Cities. *IEEE Network*, *33*(2), 102–110. https://doi.org/10.1109/mnet.2019.1800258

Khan, L. U., Yaqoob, I., Tran, N. H., Kazmi, S. M. A., Dang, T. N., & Hong, C. S. (2020). Edge Computing Enabled Smart Cities: A Comprehensive Survey. *IEEE Internet of Things Journal*, 10200–10232. https://doi.org/10.1109/jiot.2020.2987070

Khayatian, F., Sarto, L., & Dall'O', G. (2016). Application of neural networks for evaluating energy performance certificates of residential buildings. *Energy and Buildings*, *125*, 45–54. https://doi.org/10.1016/j.enbuild.2016.04.067

Lee, J.-yi, Bala, G., Cao, L., Corti, S., Dunne, J., Engelbrecht, F., Fischer, E., Fyfe, J., Jones, C., Maycock, A., Mutemi, J., Ndiaye, O., Panickal, S., Zhou, T., Kanikicharla, K. K., Kattsov, V., & Kimoto, M. (2021). Chapter 4: Future global climate: scenario-based projections and near-term information. In J. Marotzke (Ed.), *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change* (pp. 952–1147). essay, Cambridge University Press.

Mendiboure, L., Chalouf, M.-A., & Krief, F. (2019). Edge Computing Based Applications in Vehicular Environments: Comparative Study and Main Issues. *Journal of Computer Science and Technology*, *34*(4), 869–886. https://doi.org/10.1007/s11390-019-1947-3

Mocnej, J., Miškuf, M., Papcun, P., & Zolotová, I. (2018). Impact of edge computing paradigm on energy consumption in IoT. *IFAC-PapersOnLine*, *51*(6), 162–167. https://doi.org/10.1016/j.ifacol.2018.07.147

Monteiro, P., Zanin, M., Menasalvas Ruiz, E., Pimentão, J., & Alexandre da Costa Sousa, P. (2018). Indoor Temperature Prediction in an IoT Scenario. *Sensors*, *18*(11), 3610. https://doi.org/10.3390/s18113610

Moreno, M., Úbeda, B., Skarmeta, A., & Zamora, M. (2014). How can We Tackle Energy Efficiency in IoT Based Smart Buildings? *Sensors*, *14*(6), 9582–9614. https://doi.org/10.3390/s140609582

Nicol, J. F., & Humphreys, M. A. (2002). Adaptive thermal comfort and sustainable thermal standards for buildings. *Energy and Buildings*, *34*(6), 563–572. https://doi.org/10.1016/s0378-7788(02)00006-3

Pardoe, I., Simon, L., & Young, D. (2018). *2.1 - what is simple linear regression?* 2.1 - What is Simple Linear Regression? | STAT 462. Retrieved September 15, 2021, from https://online.stat.psu.edu/stat462/node/91/.

Paul, D., Chakraborty, T., Datta, S. K., & Paul, D. (2018). IoT and Machine Learning Based Prediction of Smart Building Indoor Temperature. *2018 4th International Conference on Computer and Information Sciences (ICCOINS)*. https://doi.org/10.1109/iccoins.2018.8510597

Puri, M., Solanki, A., Padawer, T., Tipparaju, S. M., Moreno, W. A., & Pathak, Y. (2016). Introduction to artificial neural Network (ANN) as a predictive tool for drug Design, discovery, delivery, and disposition. *Artificial Neural Network for Drug Design, Delivery and Disposition*, 3–13. https://doi.org/10.1016/b978-0-12-801559-9.00001-6

Puri, M., Solanki, A., Padawer, T., Tipparaju, S. M., Moreno, W. A., & Pathak, Y. (2016). Introduction to artificial neural Network (ANN) as a predictive tool for drug Design, discovery, delivery, and disposition. *Artificial Neural Network for Drug Design, Delivery and Disposition*, 3–13. https://doi.org/10.1016/b978-0-12-801559-9.00001-6

Pérez-Lombard, L., Ortiz, J., & Pout, C. (2008). A review on buildings energy consumption information. *Energy and Buildings*, *40*(3), 394–398. https://doi.org/10.1016/j.enbuild.2007.03.007

Rajagopalan, R., & Varshney, P. (2006). Data-aggregation techniques in sensor networks: a survey. *IEEE Communications Surveys & Tutorials*, *8*(4), 48–63. https://doi.org/10.1109/comst.2006.283821

Ruano, A. E., Crispim, E. M., Conceição, E. Z. E., & Lúcio, M. M. J. R. (2006). Prediction of building's temperature using neural networks models. *Energy and Buildings*, *38*(6), 682–694. https://doi.org/10.1016/j.enbuild.2005.09.007

Satyanarayanan, M. (2017). The Emergence of Edge Computing. *Computer*, *50*(1), 30–39. https://doi.org/10.1109/mc.2017.9

Shahzadi, S., Iqbal, M., Dagiuklas, T., & Qayyum, Z. U. (2017). Multi-access edge computing: open issues, challenges and future perspectives. *Journal of Cloud Computing*, *6*(1), 1–13. https://doi.org/10.1186/s13677-017-0097-9

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, *3*(5), 637–646. https://doi.org/10.1109/jiot.2016.2579198

Siow, E., Tiropanis, T., & Hall, W. (2018). Analytics for the Internet of Things. *ACM Computing Surveys*, *51*(4), 1–36. https://doi.org/10.1145/3204947

Sittón-Candanedo, I., Alonso, R. S., García, Ó., Muñoz, L., & Rodríguez-González, S. (2019). Edge Computing, IoT and Social Computing in Smart Energy Scenarios. *Sensors*, *19*(15), 3353. https://doi.org/10.3390/s19153353

Sun, X., & Ansari, N. (2016). EdgeIoT: Mobile Edge Computing for the Internet of Things. *IEEE Communications Magazine*, *54*(12), 22–29. https://doi.org/10.1109/mcom.2016.1600492cm

Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017). On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. *IEEE Communications Surveys & Tutorials*, *19*(3), 1657–1681. https://doi.org/10.1109/comst.2017.2705720

Tomovic, S., Yoshigoe, K., Maljevic, I., & Radusinovic, I. (2016). Software-Defined Fog Network Architecture for IoT. *Wireless Personal Communications*, *92*(1), 181–196. https://doi.org/10.1007/s11277-016-3845-0

Truong, H.-L., & Dustdar, S. (2015). Principles for Engineering IoT Cloud Systems. *IEEE Cloud Computing*, *2*(2), 68–76. https://doi.org/10.1109/mcc.2015.23

Varghese, B., & Buyya, R. (2018). Next generation cloud computing: New trends and research directions. *Future Generation Computer Systems*, *79*, 849–861. https://doi.org/10.1016/j.future.2017.09.020

Wang, W., Feng, C., Zhang, B., & Gao, H. (2019). Environmental Monitoring Based on Fog Computing Paradigm and Internet of Things. *IEEE Access*, *7*, 127154–127165. https://doi.org/10.1109/access.2019.2939017

Westerink, J. (2018). CE 30125 Computational Methods Course Notes. Notre Dame; University of Notre Dame.

Yang, T.-H., Wang, C.-W., & Lin, S.-J. (2020). ECOMSNet – An edge computing-based sensory network for real-time water level prediction and correction. *Environmental Modelling & Software*, *131*, 104771. https://doi.org/10.1016/j.envsoft.2020.104771

Zamora-Martinez, F., Romeu-Guallart, P., & Pardo, J. (n.d.). UCI Machine Learning Repository: SML2010 data set. Retrieved from https://archive.ics.uci.edu/ml/datasets/SML2010.

Zamora-Martínez, F., Romeu, P., Botella-Rocamora, P., & Pardo, J. (2013). Towards energy efficiency: Forecasting indoor temperature via multivariate analysis. *Energies*, *6*(9), 4639–4659. https://doi.org/10.3390/en6094639

Zamora-Martínez, F., Romeu, P., Pardo, J., & Tormo, D. (2012). Some Empirical Evaluations of a Temperature Forecasting Module based on Artificial Neural Networks for a Domotic Home Environment. *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*. https://doi.org/10.5220/0004133502060211