UNIVERSITY OF RWANDA

COLLEGE OF SCIENCE AND TECHNOLOGY

AFRICAN CENTER OF EXCELLENCE IN INTERNET OF THINGS (ACEIoT)

ANALYSIS AND PERFORMANCE OF IoT-BASED FRAMEWORK FOR FIRE INCIDENCE APPLICATIONS

PhD. Thesis submitted in the fulfillment of requirements

of the award of PhD. Degree in the Internet of Things – Embedded Computing Systems.

Eric HITIMANA

JULY 2023

Eric Hitimana, July 2023,

UNIVERSITY OF RWANDA

COLLEGE OF SCIENCE AND TECHNOLOGY

AFRICAN CENTER OF EXCELLENCE IN INTERNET OF THINGS (ACEIoT)

ANALYSIS AND PERFORMANCE OF IoT-BASED FRAMEWORK FOR FIRE INCIDENCE APPLICATIONS

Ph.D Thesis submitted in the fulfillment of requirements of the award of PhD. Degree in the Internet of Things – Embedded Computing Systems.

Eric HITIMANA

Registration number: 2018014957

Thesis Supervisor: Dr. Gaurav Bajpai, PhD

Thesis Co-Supervisors: Dr. Louis Sibomana, PhD

Thesis Resident Co-Supervisor: Prof. Richard Musabe, PhD

JULY 2023

Eric Hitimana, July 2023,

Eric Hitimana, July 2023,

DECLARATION

I hereby declare that the dissertation entitled "**Analysis and Performance of IoT-Based Framework for Fire Incidence Applications**" to be submitted for the Degree of Doctor of Philosophy on the Internet of Things (Embedded Computing Systems) at the University of Rwanda/College of Science and Technology is my original work and has not formed the basis for the award of any degree, diploma, associateship, or fellowship of similar other titles. I also declare that all source of information used was acknowledged by a complete list of references and was well cited.

Signature: 

HITIMANA Eric

Eric Hitimana, July 2023,

Eric Hitimana, a Ph.D student of UR-ACEIoT registration ID 2018014957, successfully defended the thesis/dissertation entitled "**Analysis and Performance of IoT-Based Framework for Fire Incidence Applications**", which he prepared after fulfilling the requirements specified in the associated legislation, before the thesis examination members whose signatures are below.

Thesis Supervisor:	**Dr. Gaurav Bajpai** Kampala International University, Uganda	 ----------
Co-Supervisor:	**Dr. Louis Sibomana** National Council for Science and Technology	 ----------
Resident Co-Supervisor:	**Prof. Richard Musabe** Rwanda Polytechnic	 ----------
Viva Voce Members:	**Prof. Denis Ndanguza (Chair)** University of Rwanda	 ----------
	Prof. Vincent Havyarimana Burundi Higher Institute of Education (ENS)	 ----------
	Dr. Abdi T. Abdalla University of Dar es Salaam	 ----------

Date of Submission : 07/08/2023

Date of Defense : 10/07/2023

Eric Hitimana, July 2023,

Eric Hitimana, July 2023,

To my spouse and children,

Eric Hitimana, July 2023,

Eric Hitimana, July 2023,

FOREWORD

This thesis is the final deliverable of Ph.D. research conducted in the African Centre of Excellence in Internet of Things (ACEIoT) with the financial support of the Government of Rwanda through the African Centers of Excellence (ACEs) project.

After the opening of ACEIoT in 2017 with the initiative to provide the knowledge-based and applicability of the Internet of Things in real life, with the tragedies of people lost due to fire outbreaks in Kigali and Rwanda resulting in people's loss, description of various properties as well as infrastructure. After many investigations of how people can be saved using IoT, I decided to come up with research on IoT applied to firefighting mechanisms. We come up with the research topic entitled "Analysis and Performance of IoT Framework for Fire Incidence Applications", to contribute to the community. It is in this regard I address my thanks to the Ministry in Charge of Emergency Management, Rwanda National Policy to give us a chance to use the data related to fire incidences.

I would like to thank the Government of Rwanda as well as the University of Rwanda for their initiatives to empower and strengthen the human capacity to promote a knowledge-based economy by delivering high-quality, market-relevant postgraduate education in Rwanda.

Let me express my sincere gratitude to the ACEIoT administration team who kept consistent administration and guidance to me during this journey of more than four years. Thanks to Prof. Hanyurwimfura Damien and Dr. Gatera Omar for your guidance, inspiration, administrative support, and facilitation to accomplish different research activities.

My great thanks go to my main supervisor Dr. Gaurav Bajpai for his courage, guidance, discussions, opportunities, and other various support you provided to make this research successful. I would like to thank my co-supervisor, Dr. Louis Sibomana, for his dedicated support and guidance during this research. I also acknowledge the support of the resident co-supervisor, Prof. Musabe Richard. Your guidance and motivation are the needed ingredients to fortify me during this journey. I do remember different sessions we had together on your motivation experiences regardless of your overloaded agenda. Thanks to the other co-author who was not part of the supervision team, Prof. Kayalvizhi Jayavel, for your technical and professional contributions to research, and writing published papers.

All thanks go to our Almighty Lord for his protection in life. I would like to express my heartfelt gratitude to my parents for their moral support from the first time when joining the

University of Rwanda to date. Also, my thanks go to my classmates and workmates in the Department of Computer and Software Engineering for your encouragement, and thanks to Mr. Omar Sinayobye Janvier and Mrs. Angelique Mukasine for inspiring words when I was feeling exhausted.

Finally, thanks to my spouse Nibagwire Jacqueline for your usual patience during this long journey and your prayers. My daughters Inema H. Sheila and Imena H. Keila without forgetting my brother Byiringiro Frederick, siblings, and my parents. I apologize for being obnoxious while I was busy writing papers and could not allow you to converse as you desired. I will never forget your constant encouragement, without which I would have given up on my studies long ago. You are all a part of the accomplishments that brought this long journey to a close.

JULY 2023 **Eric HITIMANA**

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

Eric Hitimana, July 2023,

ABBREVIATIONS

3D	Three dimensions
ARM	Acorn RISC Machine
ANFIS	Adaptive Neuro-fuzzy Interface System
API	Application Programming Interface
BSON	Binary JavaScript Object Notation
BIM	Building Information Management
CPU	Central Processing Unit
CSV	Comma Separated Value
DB	Database
DCNN	Deep CNN
DevOps	Development and Operations
DDR	Double Data Rate
ERC	Emergency Responses Coordinator
NGINX	Engine X
EMQ	Erlang MQTT
FNN	Feedforward Neural Network
FFSM	Forest Fire Susceptibility Management
GHz	Gigahertz
GPS	Global Positioning System
HDD	Hard Disk Drive
HVAC	Heating, ventilation, and air conditioning
HTTP	Hyper Text Transfer Protocol
ICT	Information Communication Technology
I/O	Input and Output
IoT	Internet of Things
JSON	JavaScript Object Notation
LED	Light Emitting Diode
LAN	Local Area Network
LSTM	Long Short-Term Memory

ML	Machine Learning
MAE	Mean absolute error
MSE	Mean Squared Error
MQTT	Message Queuing Telemetry Transport
NoSQL	Not Only SQL
OS	Operating System
PDA	Personal Digital Assistant
QoS	Quality of Services
RPi	Raspberry
RHR	Rate of Heat Released
Rx	Reception
RNN	Recurrent Neural Network
RISC	Reduced Instruction Set Computer
REST	Representational State Transfer
RMSE	Root Mean Square Error
SMS	Short Message
SD	Standard Deviation
SQL	Structured Query Language
SVM	Support Vector Machine
SVR	Support Vector Regression
SBC	System Basic Chip
Tx	Transmission
UR	University of Rwanda
UAV	Unmanned Aerial Vehicle
VM	Virtual Machine
VR/AR	Virtual Reality/ Augmented Reality
WUI	Wildland-Urban Interface
Wi-Fi	Wireless Fidelity
WSN	Wireless Sensor Network

SUMMARY

Natural disasters are one of the leading causes of death worldwide. Statistics show that many people die in various catastrophic events. The damage to property and infrastructure from these types of disasters is worth millions of dollars, and many more dollars are spent on disaster recovery. In many countries, including the East African region, indoor fire outbreaks are the most common natural disasters with fatalities. Least developed or developing countries do not have efficient mechanisms to detect, count or predict human occupants in buildings in the event of a fire.

In Rwanda, common strategies to reduce the risk of fire outbreaks include emphasizing the use of firefighting equipment and following fire codes to enable firefighters to intervene and rescue a reasonable number of casualties. These methods do not guarantee that trapped people have the skills to use the device or know these guides in case of panic. The internet of things and machine learning are now available. The former will play a key role in developing data collection prototypes to collect and monitor environmental parameters and report them to the cloud in real-time. The latter helps predict occupancy using sensor-captured data from indoor environments.

This research aimed to reduce the risks of fire outbreaks in public buildings by identifying and analyzing the environmental factors to be considered as indicators for fire existence, and the recent technologies adopted to fight against the matter. After identification of parameters and devices capable to support on fire detection and human counting, the development of a scalable spruce fire IoT-based system aiming to warn and predict humans in case of fire is done. The system uses wireless sensors to collect real-time environmental data, report estimated occupancy, and then predict the human presence. By using the developed IoT system, the building environmental parameters were collected, and machine learning techniques have been used to predict human presence. The IoT framework was containerized to support system scale-up and performance metrices were assessed.

The research was conducted in three systematic phases: The data acquisition IoT prototype was developed, calibrated, and tested to gather real-time data to be used in the prediction. In this phase, the environmental parameters' thresholds were experimentally captured. In the second phase, the IoT framework has been improved with machine learning model integration. The data

generated was used to build the insight between people's presence and the environmental parameters in a room using machine learning prediction techniques. Different machine learning models were tested, and the long and short-term memory model was selected with accuracy of 96%.

In the third phase, the framework was designed using the microservices paradigm, to analyze the performance during the increase of the distributed sensor nodes over the same managerial process. The performance was measured experimentally on 3 nodes by varying different requests 0, 100, 10.000, 100.000, and 1.000,000 respectively. Using Grafana and Prometheus, the system resources considered are CPU utilization, memory usage, as well as data transmission (receiving and transmitting bytes). The results showed that due to the containers, the CPU used did not cost high by increasing the number of requests. For memory resources, the increases do not cost any attention. The transmission increases considerably to the bytes received starting from 10.000 requests. The system prototype was successfully tested at the selected site and shows the better performance in case of scaling up. The study area of this research is the University of Rwanda, Kigali City, and Nyarugenge districts in Rwanda.

Eric Hitimana, July 2023,

Chapter 1

BACKGROUND OF STUDY

This chapter presents the background of the study where the problem area is defined together with the aims and objectives. Further, it discusses the contributions of the research and the structure of the thesis and the organization of the rest of the chapters.

1.1 Introduction

In Rwanda and worldwide, natural, and human-induced disasters result in a serious loss of infrastructure, monetary costs, distress, injuries, and deaths. The internet of things (IoT) has inspired solutions that are already available for addressing problems in various application scenarios. For instance, IoT systems have been used in building automation and comfort management [1]. Among a variety of emerging technologies, the demand-driven HVAC (heating, ventilation, and air conditioning) method is mostly adopted to optimize the building operation cost, while achieving sustainable smart buildings [2].

Most buildings are managed when affected by disasters (fire, earthquakes, etc.) with smart properties. There are mechanisms to rescue and properly search trapped victims using evacuation policies [3]. However, its accuracy is not efficient. This thesis is adopting the use of IoT and machine learning (ML) technologies to propose an enhanced framework to discover efficiently, accurately, and automatically predict the occupants in a building. Further, warn the trapped people and the emergency response coordinators (ERC) for facilitating the evacuation process.

Artificial and natural disasters are unexpected daily events that are creating a big concern in worldwide nations. Every year, extreme weather conditions, hurricanes, earthquakes, droughts, floods, heat waves, and fire outbreaks cause considerable damage to infrastructure [4]. This leads to monetary loss, mass evacuations, distress, injuries, and deaths. Like any other country in the region, Rwanda experiences internally displaced persons due to environmental disasters, especially fire outbreaks [5].

Rwanda disaster management policy stated that fire hazards including unplanned and massive burning may destroy equipment, settlements, property, and lives [6]. Hazardous electric wiring, limited construction standards, and accidents are among many factors that cause fire hazards. Fires

are common in commercial buildings, industries, congested human settlements, learning institutions, and marketplaces. The community requires preparedness for the management of fire hazards by the intensification of sensitization for public awareness. Apart from the manual mechanisms of guiding people within modern buildings, there are more ways like showing signs for the position of fire extinguishers, exit routes, etc. This requires additional training for users/the public to be able to use them during an emergency. However, there is no appropriate way of warning, searching, and rescuing the victim during an emergency.

1.2 Fire Outbreak in the Context of Rwanda

Rwanda is a country with a fast-growing economy along with infrastructure development involvement [7]. Due to its cities with regular expansion, there is an increase in fire risks as more houses are constructed and a growing number of populations without knowledge about fire safety measures, laws, and regulations.

The main root cause of fire hazards includes unplanned and massive burning that destroys equipment, settlement, property, and lives. Fires are mostly present in industries, congested human settlements, learning institutions, and marketplaces. For the readiness, preparedness as well as management of fire hazards and for the intensification of sensitization with the public, there are guidelines designed to overcome those hazards [6]. The fire hazards in Rwanda are classified into different categories such as wildfires, industrial fires resulting in methane gas extraction, and fire outbreaks in public and private settlements. Forest fire in other words knows us wildfire has been affecting different regions, especially in national parks [8], and this is a serious concern in many ecosystems across the globe due to environmental and human life loss.

As per the East African Community (EAC) [9] on disaster risk reduction and management strategy in 2012–2017, the abundant natural hazards such as floods, droughts, urban fires, and earthquakes negatively contribute to the impacts on the security of lives as well as economies of their partner states. It is stated that the cause behind the fire hazards such as haphazard electric wiring, disregarding construction standards, accidents, and uncontrolled burning including bush waste materials burning [9].

Rwanda as an Eastern African country has managed to establish a law for the fire and rescue brigades under the Rwanda national police [10]. All control strategies of the buildings in the capital/secondary cities are inspected regularly to handle fire incidents and other possible disasters.

The law again enforces naming commercial houses for better allocations for smart city management.

Figure 1.1 displays all fire damage observed between the period 2010 to 2019 [5]. According to the graph, data for 2015 is not available.

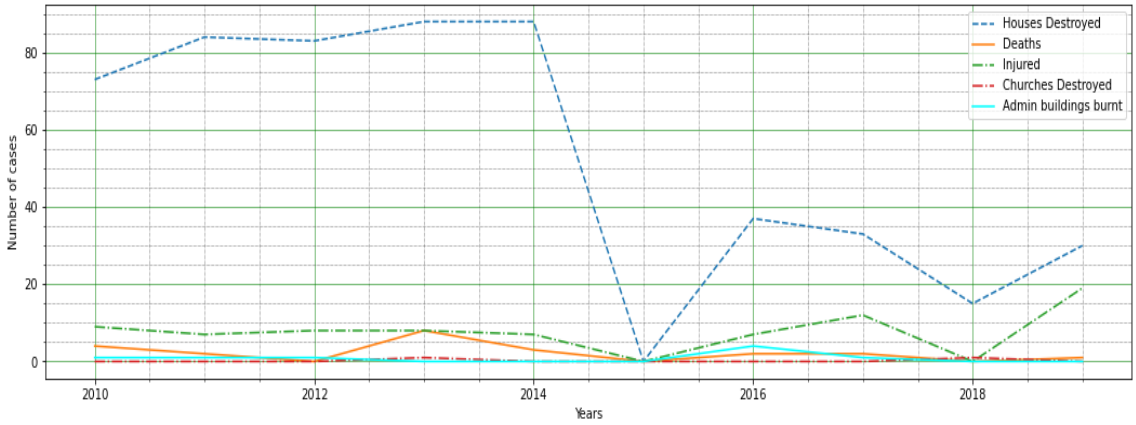


Figure 1. 1: Fire damages in Rwanda in the last 9 years.

The impact of natural disasters on remittance flows toward low- and middle-income countries had been discussed [11]. A proposal for global expected risk analysis of fatalities [12], injuries, damages by natural disasters [13], and earthquakes inclusive of fire hazards are global issues to care about [14] [15]. Most of the efforts have been done to fight against the loss of human lives, the unexpected disasters causing the buildings to collapse or be burnt. The buildings require regular evacuation drills causing expenses, and loss of productivity. Still many occupants are not experienced with the usage of firefighting equipment. Any automation to assist in these drills during real-time actual evacuations would mean saving cost, time, and life.

1.3 Motivation

Rapid urbanization and growing city centers such as Kigali present the rise of commercial growth where huge commercial buildings, trade centers, and private as well as public infrastructures have been established [16]. The growth in infrastructure and population results in high exposure to fire hazards and increasing vulnerability. After accidents involving motor vehicles, falls, and drowning, fires are the fourth-leading cause of accidental injury worldwide, accounting for over 300,000 fatalities each year [17]. In an urbanizing world, there is a strong correlation between the risk of burn injuries and densely populated urban areas [18]. Due to the

high population density and high economic value of the buildings and their contents, fires in high-rise buildings and residential complexes are one of the major urban disasters [19].

The inspections and assessments conducted in [20] revealed that most public and private buildings and houses are not equipped with firefighting equipment and the public is not aware of laws as well as regulations related to fire safety management. Firefighting and evacuation efforts are very difficult in the fire event due to the high concentration of people and property [21]. The economic value of the structure itself, the interior construction, and the possessions of the building's occupants highlight the significance of the fire crisis management issue [22].

During the emergency case in the building, the key factor is to know the number of people in the building for the evacuation assistance systems to rely on the attendance of humans in the specific building. Normally, this number is calculated by counting the number of people gathered at the assembly points to the last known number of people inside the building. The problem of the current attendance can be done in the office building by knowing the already present employees. However, the system will not consider visitors.

On the other hand, technologies have been used to make decisions about some events that may happen. Research had been conducted on commercial buildings for context-aware computing [23], here estimation of occupancy in indoor environments using different kinds of sensors data analysis had been conducted. Knowing the number of occupants in a living environment facilitates the management of buildings more intelligently for human evacuation in case of emergency. It also controls ventilation, heating, and cooling for energy consumption management. Efficiency ratings of most LEED (Leadership in Energy and Environmental Design) certified buildings require such sensor systems. The concept of smart building arose through a combination of the IoT paradigm with the smart object approaches. Smart environments focus on looking at occupancy in the specific indoor area, like tracking, detecting, and identifying occupants to offer quality services, while considering factors [24].

The IoT and Machine Learning is one of the emerging technologies that may be used to address the issue of fire outbreaks by remotely collecting data and delivering it to the cloud platform to notify concerned people of the specific area. The importance of using IoT techniques for fire disaster monitoring and prediction is its rapidness to respond to the changes in data, by providing accurate information on the imminent disaster before its occurrence. By using IoT, many parameters can be gathered and analyzed to increase the accuracy of occupant prediction. In this

thesis, various parameters of the indoor environment have been considered by conducting experiments in the respective areas of study, and different methods were considered that leverage the improved accuracy of the prediction and operationalization of the framework.

IoT deployments have grown in number at an unprecedented rate in smart buildings [25]. It helps industries and researchers to address the solutions in the direction of time management while evacuating people struck by either natural or artificial disasters. It has been observed that the ERC easily panics causing a high probability of not reducing survivability [26]. Most buildings have evacuation guidance marked with specific signs. However, they look inadequate and are not considered when people are on the intensive run for their lives. In general, Rwanda and the East African region have been known for fire outbreak cases in the past few years [27]. The "life cycle" of comprehensive emergency management, also known as the phases of the traditional disaster model to assist emergency managers, consists of four stages: mitigation, readiness, response, and recovery. The cycle of disaster framework aims to reduce fatalities in disasters by choosing the best courses of action and establishing timely and adequate levels of human and financial resources [28, 29].

1.4 Problem Statement

The big challenge among others is the lack of knowledge for building occupancy at a specific period. It seems impossible to get the precise number of people in a building. It is even hard to inform the ERC for intervention. Most recent buildings in the region lack automatic little prevention mechanisms while waiting for help with real fire-fighting equipment. The problem is how to precisely know and predict the number of present people in a building so that ERC can locate them in real-time.

This research focused on the development of an integrated machine learning model into an IoT Sensor-based system to predict and count human occupants in public buildings. It suggested performance measurement of the system due to scale-up with the large data ingestion.

1.5 Research Objectives

The main objective of this thesis was to develop an IoT-based framework and undertake a performance analysis in case of fire outbreaks in buildings in an urban area.

The specific objectives are:

1. Assess the current fire emergency scenario in Rwanda and review current trends of home automation mechanisms for detecting victims in buildings using the IoT.
2. Develop an innovative IoT architecture for occupant counting with alerting mechanism.
3. Select and integrate a machine learning algorithm into the IoT framework architecture for occupant prediction, and response applied to public buildings.
4. Test and validate an IoT framework architecture using performance metrics for scalable system scenarios.

1.6 Research Questions

The main objective of developing the framework is to contribute to preventing the cases of increased fire outbreaks with the loss of human beings. This thesis aims to provide a smart framework that can intelligently detect fire outbreaks leading to prevention and occupant prediction. The solutions proposed in this work aim to address the identified research questions listed below:

1. Conduct a literature review on what are the current fire incidences trends and their prevention measures using IoT in Rwanda?
2. What are the suitable IoT deployment technologies to be applied?
3. What is the appropriate IoT-based framework to address the machine learning models of fire prevention, discovery, prediction, and response for public buildings?
4. What are the best testing and validation methods for performance metrics to assess system scale-up and count the occupants and notify the existence of fire outbreaks?

1.7 Research Methodology

This research employs both modeling and IoT integration approaches to propose solutions for improving building occupant detection from fire outbreaks. The solution proposed in this thesis is relevant for countries struggling for improved fire prevention safety systems by highlighting the process to achieve intelligent mechanisms for fire detection and occupant prediction. To achieve

the objectives of this work, qualitative, quantitative, and experimental approaches were combined. These three approaches complemented each other to achieve the objectives of this research [30].

The research was conducted by literature review both online and offline. To capture the real scenarios, we conducted different site visits. Various stakeholders including the Rwanda National Police (RNP) and the ministry in charge of emergency involved in firefighting incidences were consulted. From the discussions, the idea was to contribute to the measures of reducing the problem of fire detection, alerting, people counting as well as predicting fire incidences before time using emerging technologies.

The data related to fire incidences, and the application of emerging technologies to detect, as well as predict fire outbreaks and alert humans has been discussed as the outcome of research objective 1. After data collection, the analysis phase was conducted to identify different parameters implying the fire outbreaks. The main outcome in this phase was to come up with a feasible sensor-enabled system to detect environmental parameters that do not expose human privacy. And develop an IoT solution for alerting fire brigades as planned in research objective two.

The third objective aims to select and propose a machine learning integrated-IoT framework architecture. The LSTM was chosen as the best model among many tested from the dataset collected with the developed IoT solution.

Due to the downtime based on system overload issues reported by most IoT solutions handling a lot of data ingestion coming from deployed sensors in real-time, continuous development and continuous integration (CD/CI) paradigms using DevOps were analyzed and applied to the developed framework. Testing of the system scale-up using varying requests with variant infrastructures was conducted as planned in research objective four.

1.8 Overview of the Major Contributions

The major contributions of this thesis are as follows:

- Literature survey for IoT and ML applied to fire outbreaks in an urban building,
- Development, and deployment of an embedded device for fire detection and data acquisition for room occupant counting with real-time data visualization,
- Prediction analysis using different ML algorithms from the gathered fire-related sensor data.

- Development of a containerized framework to handle or manage multiple concurrently deployed IoT edge devices.
- Performance evaluation based on the system scalability for IoT resources utilization.

The above contributions have been developed based on the research questions and linked to the objectives. The outcomes are grouped into three research papers published.

1.9 Thesis Outline

Chapter 2 presented the literature review on fire outbreaks, their causal factors, and the mechanisms used to reduce their risks. Chapter 3 discusses the study of fire outbreaks by designing and developing an IoT-based application for spruce fire. The implementation of the system is done and tested on the detailed parameters. The data visualization is done, and the recommendation is addressed for prediction and performance evaluation. Chapter 4 details the proposed IoT framework implementation and its occupancy prediction using deep learning methods. The chapter comprises a description of the framework and the parametric characteristics of the observed variables. The results from the experiments are presented with the recommendation. Chapter 5 describes the development of a framework with performance evaluation analysis. The components of the system are described, and the evaluation is experimented based on reliability and throughput. Chapter 6 concludes the thesis. The reference to the thesis structure is found in Figure 1.2.

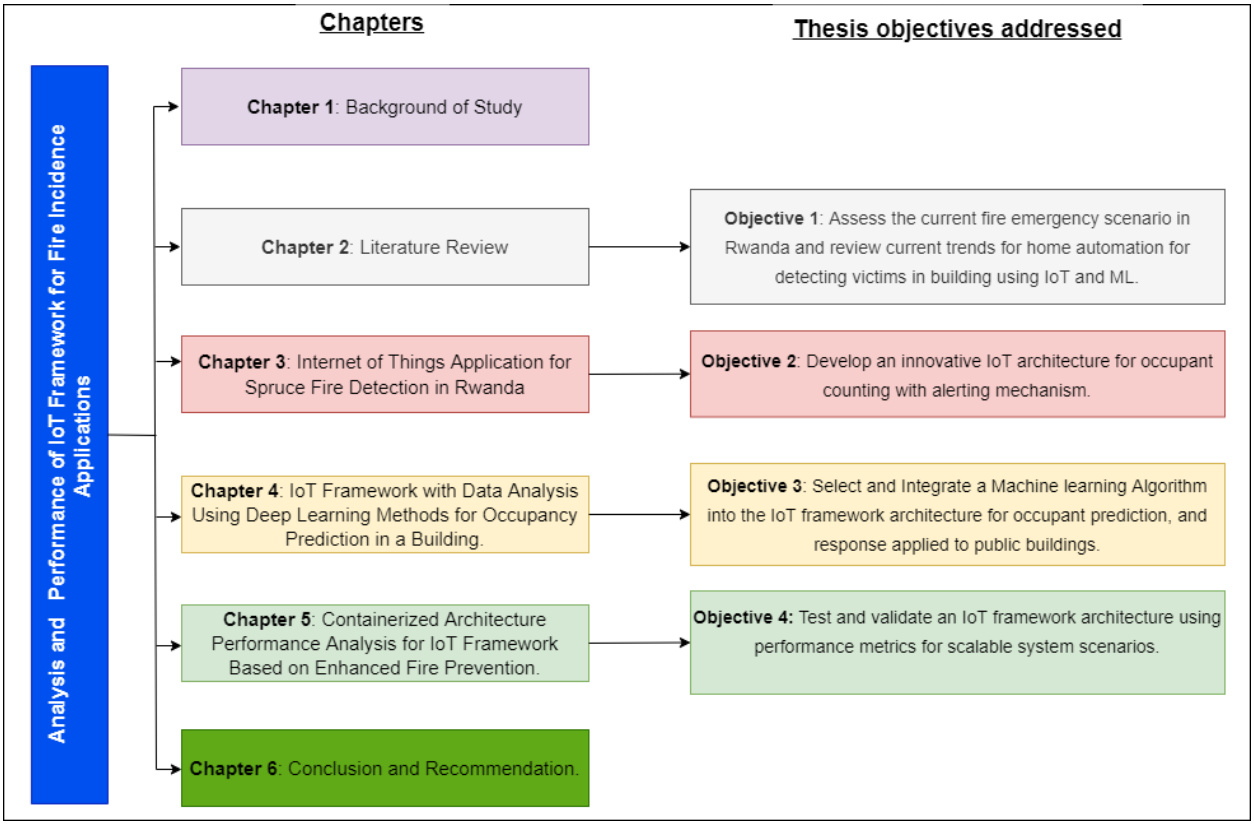


Figure 1. 2: The structure and organization of the thesis.

Eric Hitimana, July 2023,

Chapter 2

LITERATURE REVIEW

This section details the literature review regarding the fire outbreaks in Eastern Africa and Rwanda in general. It discusses the fire outbreak categorization, risk reduction strategies and use of emerging technologies such as internet of things, machine learning to fight against fire outbreaks.

2.1 Introduction

Nowadays, global observations of natural hazards show a massive increase in the number of disasters with their total economic impacts [31]. Fire hazards were reported worldwide many times as a global significant problem and the root cause of natural ecosystem degradation, with loss of people's lives [33]. The East African Sub-Region is known for natural hazards such as floods, droughts, earthquakes, landslides, fires, strong winds, lighting, and their secondary impact on diseases and epidemics [32]. Rwanda is among the vulnerable country to a wide variety of natural and manmade hazards.

Over the last decade, the frequency along with the intensity of natural hazard-induced disasters, particularly fire, and flood have significantly increased, raising the toll of human loss as well as economic and environmental losses.

One of the main problems today that seriously harms the environment and has an impact on human life is fire detection. Most of the time, fire accidents are more serious and cause environmental destruction. However, it is possible to reduce danger and save lives by identifying and detecting the high-risk factors and the early stages of a fire accident. It is important to inform the appropriate caring unit and those who are closest to the burning area to leave and extinguish the fire source.

This chapter discusses different situations related to fire risks and the technologies used to facilitate the prevention and rescue of fire victims. Its main purpose is to survey the state of the arts of the current situation and proposes the way forward for this research.

2.2 Fire Outbreak Categorization

Fire risk can be classified into different categories based on the scenario [34]. For example, it can be classified based on location. In this category, there is indoor and outdoor fire. The former is categorized into the urban fire. The latter is classified as forest fire and wildfire. The fires classes are discussed based on fire sources: (i) class A – ordinary solid combustibles, (ii) class B – flammable liquids such as alcohols, (iii) class C – electrical equipment, and (iv) class D – certain flammable metallic substances such as sodium and potassium [34].

There is the occurrence of temperature increase over time during the fire's ignition and development phases in addition to the production of smoke and toxic gases [35, 36]. Due to a high concentration of combustion (CO, CO_2, HCN, SOx, NOx), most victims pass away from asphyxiation and intoxication [37]. Even if the subject was not directly exposed to the flames, those gases, at high temperatures, can cause burns, eye irritation, and breathing difficulties, leading to death [38]. Due to the smoke's tendency to reduce visibility and its color, which is determined by the materials burning, it is difficult for the victim to flee and for the rescue team to assist [39]. Examining factors like the fire position, the quantity of combustibles, and the length of ventilation openings that may affect the potential occurrence of the flashover, which occurs just after the fire has fully developed, is crucial for maintaining a safe environment indoors [40].

When it reaches this stage, the only course of action that can be taken is to keep the fire under control until the combustible flame has been put out [41]. The values of the RHR (Rate of Heat Released), which is related to the thermal power [kW] released and to the velocity of the process [42], have an impact on the various stages characterizing the development of the fire (pre and post flashover). Determine the value of the RHRlim, which, if not exceeded, will prevent the fire from reaching the flashover. The RHR is influenced by the flame speed, amount of available air, and exchange rate. This is because of the high temperatures in the area, which must be avoided because they could break windows at a temperature of about 500°C. It is possible to distinguish between two methods for the analysis of the fire during the post-flashover phase. The first one adopts a prescriptive methodology and investigates the thermal effects of the nominal fire on the building materials. The second employs performance-based assessment codes whose complexity can be altered by the level of accuracy required by the evaluation [43].

This section discussed the fire categorizations and next section shows the risks associated as well as the strategies proposed by the concerned policymakers to reduce those risks.

2.3 Fire Outbreak's Risks Reduction Strategies

The main challenges for fire risk mitigation are the lack of information about the fire outbreak happening, and what is about to happen [44]. Broadly, some relevant measures for fire risk mitigation rely on (i) planning area, (ii) community by considering whether the structure was built under current Wildfire and the wildland-urban Interface (WUI) building codes, (iii) community design (by checking the lot size, density, and internal road system), (iv) general property hygiene [45]. In modern strategies, there is a plan for every country to codify each building to ease localization and accessibility.

Among many strategies, this section details the ways of assessing vulnerability causing the fire incidences, reducing risks by applying structural techniques, using emerging technologies such as drones, imaging, internet of things as well as machine learning methods to reduce risks caused by fire outbreaks in indoor buildings. These measurements are the key important points to assess the feasibility study of our proposed research in the context or East African and Rwanda in general.

2.3.1 Fire Outbreak Susceptibility Assessment

Even though indoor fire outbreaks are unforeseen and undetected, there are various ways to lower their risks. One of the recommended methods for a forest fire is to apply the zonation technique. For the case of indoors, the most recommended measure is to use proper construction tools and make the building accessible to help fire brigades. The other method is susceptibility assessments of electrical wiring failures in various areas. These two techniques may be used to reduce risks. These analyses provide information on which areas are highly susceptible and which have moderate to low susceptibility. Remote sensing [46, 47], geographic information systems (GIS) in conjunction with bivariate statistical analysis methods [48], data mining techniques [49], and multi-criteria decision-making [50] are all used in the mapping of fire susceptibility and other types of hazards.

The laws governing the construction of homes, schools, hospitals, commercial buildings, and other types of infrastructure are known as "building acts" and are found in many nations. It is stated that no building should be in dangerous and inaccessible areas, they should be constructed with minimal prevention measures (possession of firefighting equipment).

2.3.2 Fire Outbreak Risks Reduction by Structural Techniques

Structural techniques for fire risk outbreak mitigations rely on technical materials used in construction and applying safety measures. Modern buildings are frequently made with fewer resources and designed to be lighter, stronger, and more resilient due to global trends of population growth, the high cost of resources, and a growing need for sustainably sourced building materials [51]. Without additional treatments, however, this could also mean that the materials are less fire-resistant [52], especially if they are not strengthened to meet standards like those outlined by the National Fire Protection Agency (NFPA) [53] or in regional, national, or international building codes. From a life safety standpoint, fire-resistant materials are particularly crucial in residential construction because residential fires are the main cause of fire fatalities in Rwanda [54] and around the world [55-58].

This section focuses on catastrophic events that are out of sight right now rather than addressing worries about a disaster that is already underway [59, 60]. This could entail modifying local building codes to fortify structures, updating zoning and land use regulations, bolstering public infrastructure, and other initiatives to increase a community's resilience to a catastrophic event [61, 62]. In the meantime, risk assessment is crucial to both the planning process and disaster risk management. A risk assessment's goals are to describe the risk's nature, pinpoint communities' vulnerabilities, and determine their potential exposure to specific hazard events. Some factors go into determining the order in which risk management measures should be implemented. They include the probability and impact of potential events, the cost-effectiveness of preventative measures, and the availability of resources [63].

With the information discussed on this section, this resource learnt that with the deployment or identification of the building to assign the prototype, it is important to care about what kind of the materials used during the construction, this helped to take strategies and recommend to the building owners.

2.3.3 Digital Technology

In recent decades, however, researchers have investigated disaster management using different emerging technologies. AI techniques, along with other cutting-edge technologies like remote monitoring, high-resolution sensors, high-speed computation, and data-driven methods,

have been increasingly applied in fire safety engineering since the recent boom of big data and deep learning in the 2010s.

There are specific challenges that cannot be met without combining spatial and attribute data [64, 65]. Geospatial information science (GIS) is essential to disaster management as disasters are fundamentally spatial in nature [66, 67]. Urban fire risk assessment methodologies could be upgraded by developments in crowdsourcing using geospatial data creation, and collection as well as GIS-based risk and vulnerability assessment methods [68]. Urban planners, emergency managers, and community organizations working in resource-constrained environments can identify as well as evaluate pertinent fire risk factors with the help of vulnerability maps. However, modeling the risk of fire is necessary and requires data from numerous sources. Therefore, using information fusion techniques as a tool can help decision-makers come up with mitigation and policy plans.

In the meantime, information fusion has a wide range of applications, ranging from basic pan-sharpening to feature-based fusion, decision-based fusion, data fusion, 3-D fusion, and information fusion, as well as advanced sensor fusion. The fusion of remote sensing data with information from a GIS and other spatial data sets is one of the challenges in information fusion. In the context of determining fire vulnerability, the current study used both high-rise building characteristics and urban structural factors. For creating fire risk maps, UAV (unmanned aerial vehicle) images are combined and analyzed data from various sources using information fusion techniques in GIS [69].

The Wireless Sensor Network (WSN) with handheld devices is recommended for smart building establishments [70]. An intelligent workplace used to monitor and assess the efficiency of building systems along with user comfort was developed and employed in smart construction [71]. Electrical usage was tracked in buildings using a sophisticated integrated real-time monitoring system [72]. To improve the efficacy of the emergency protocol the human-centric theory with WSN was used to collect and deliver information [73]. For detecting temperature and humidity in a subway, the proposed integration between BIM-based models and facility management processes using WSN is applied [74].

An intriguing example of a multifunctional building (offices, laboratories, lecture hall) in Riyadh City used a thermal simulation for a building by applying an as-built BIM model, as well as a comparison of sensor data [75]. Each space had a defined purpose, and the settings were changed by that purpose. The solar path was defined based on the building orientation after the

building was accurately located on the map in its precise location using Google Earth. The weather file was then attached to the 3D model. Due to their sensitivity to changes in the external environment, critical spaces were identified. Building efficiency in terms of energy use and carbon emissions was tracked using energy analysis [75].

2.3.4 Internet of Things (IoT) for Fire Outbreak Data Acquisition and Notification

Due to its popularity, WSN is used in a variety of applications, including localization [76], smart transpiration [77], target tracking [78], healthcare [79], industrial automation, and environment [80]. Different techniques are used to implement its technologies. WSN is used for surveillance monitoring both automatically and with the assistance of users [81]. Applications based on WSNs aid in monitoring humans, animals, and industry monitoring (underground applications) [82].

IoT has not been left behind for fire monitoring as well. Applications for detecting fires used sensors and sensor-based circuits [83]. A new approach to mine fire detection was put forth by using the WMSS (WSN-Based Mine Safety System) sensor network. By analyzing the gas level, gas sensors are used for fire detection in mines. For detecting fires in remote forest areas, the author of [84] proposed a Zigbee-based WSN. Temperature sensors are used to measure the temperature to assess the severity of the forest fire that occurred there. To accomplish this, the sensor network's hardware uses the CC2430 chip.

For fire detection, the use of a variety of clustering techniques along with communication protocols was used [85]. To prevent false alarms, a novel method for fire detection using multi-sensor and cameras connected wirelessly via IP is implemented. The method used a gateway in a cloud environment, for data transmitting, uploading, and downloading [86].

The safety monitor system is implemented to keep an eye on the release of hazardous gases from a building in northern Taiwan. The system could be utilized for extra tasks like temperature and humidity measurements [87]. An integrated system needing to use a BIM-based interface is developed that communicates with Bluetooth, wireless networks, and other sensors. The system provides the following data: (a) environmental perception; (b) fire sense; (c) locating/evacuation rescue; (d) fire location; and (e) global information [88].

It was observed that most of the used CCTV cameras mapped to the WSN network was reported as privacy exposure as it may expose occupant privacy. Also, other techniques focused mostly on fire detection and alarming.

2.3.5 Machine Learning for Fire Risk Prediction

The fundamental idea behind AI techniques is multi-dimensional fitting using artificial neural networks (ANNs). It is possible to identify and forecast real-time fire information, such as HRR (heat release rate) and the probability of flashover, based on the sparse sensor data by training the dataset of fire tests [89, 90] and numerical fire simulations [91- 94].

To date, various machine-learning algorithms [95, 96] have been put forth to train databases and establish the intricate connection between sensor data and fire to forecast fires in real-time. The long short-term memory (LSTM) RNN loop structure can be unfolded into a chain of cells. Recurrent neural networks (RNNs) are specifically designed for the prediction or classification of temporal data. The LSTM-RNN model has been used in many domains as a deep learning model to predict future effects.

In the air pollution domain, the prediction model based on the fine Particulate Matter (PM) concentrations demonstrated in 24 stations is developed in Seoul [97]. In environmental disasters, the LSTM neural network model for flood forecasting is implemented. They used daily discharge and rainfall data for prediction purposes [98]. It was used in traffic forecasting [99]. A design of a novel custom power demand forecasting algorithm based on the LSTM regarding the recent power demand patterns [100]. They collected the power usage data every five-minute resolution in the day from some groups of residential, public offices, hospitals, and industrial factories buildings for one year. The feasibility of LSTM in the water quality analysis from samples across the Yangtze River in Yangzhou has been explored [101]. The LSTM is more feasible and effective in applications like this research, where the data comes from various IoT sensors contributing to big data analysis.

For identifying fire scenario information and predicting upcoming critical fire events, RNNs-based AI algorithms are especially helpful [102, 103]. Convolutional neural network (CNN) algorithms are better suited to support decision-making by fire commanders and field firefighting operations when it comes to forecasting fire scenes and images [104, 105]. To produce images

displaying the distribution of temperature, smoke visibility, and gas concentrations in fire scenes, an LSTM model, and a Deep CNN (DCNN) model are combined.

The adaptive neuro-fuzzy inference system (ANFIS) and its combinations with meta-heuristic algorithms have been a group of the most widely used models for forest fire susceptibility management (FFSM) among the various attempted forest fire susceptibility (FFS) modeling approaches [106, 107]. As opposed to the ANFIS, support vector regression (SVR), a more recent modeling methodology, has been one of the most effective models in hazard mapping as well as other applications [108, 109].

A combination of more technologies has been conducted [110]. A novel technology-integrated framework for developing a building information management (BIM), IoT, and augmented reality/virtual reality (AR/VR) proof-of-concept system based on the theory of situational awareness is developed. To assess the framework's performance, a pilot test based on a fake fire scenario is carried out. The results show that the firefighting department can use the data produced by the system to quickly find the locations of indoor fires, and the VR gamification scenarios can hasten the trainees' development of situational awareness.

The introduction of a cost-effective and automatic system design for fire detection and monitoring in IoT was discussed [2011]. Their solution enabled users to remotely keep track of a building's status through an easy-to-use dashboard that presents all relevant data on a single page. If any monitored parameter surpasses its predefined limit, the homeowner receives a notification on their mobile phone, including a buzzer, a red light, and a text message. This prompt allows them to quickly contact the authorities for immediate assistance. Additionally, the approach included a feature to count the number of people present at the site of the fire, simplifying the evacuation process. The work looks like ours done in objective 2 presented in Chapter 3. However, the main emphasis of our objective 4 as discussed in Chapter 5 was to improve the performance due to the system overload which is different from the work mentioned in [2012] too.

The main contribution of this chapter is to assess the state of the arts regarding the application of emerging technologies to reduce fire outbreaks in indoor buildings. The main gap figured out is to find the IoT time-series dataset well enough to be applied to our context. The data should be exclusive to any human privacy as more technologies such as security cameras expose human privacies. This chapter gave us insight to design an efficient calibrated IoT-based prototype to gather environmental parameters as discussed and developed in Chapter 3.

Chapter 3

INTERNET OF THINGS (IoT) APPLICATION FOR SPRUCE FIRE DETECTION IN RWANDA

In Rwanda, installing fire extinguishers is mandatory in all buildings. The number of staff trained in its efficient usage is still insufficient. Many are not aware of fire extinguisher handling. Although, alerting a fire outbreak detection on a real-time basis is still a big challenge. Furthermore, as people delay informing fire brigades, the administrative procedures are also delayed resulting in several people having fire injuries with asset damage. Developing a system that detects fire smartly and records the building's occupants by sending alert information to the fire brigade with other stakeholders on a real-time basis is the motivation of this research. This research presents a system based on the Internet of Things (IoT) that detects spruce fire in a building. The IoT system consists of various sensors that measure important factors like recording the number of occupants in the buildings through an Infrared proximity sensor. The smoke sensor sends the notification updates to the fire brigade's dashboard providing an alert in case of fire. The actuators (LEDs) are initialized for a quick exit from the spruce fire. The developed system helps to evacuate buildings' occupants against fire outbreaks in real-time information. This reduces the number of casualties, decreasing the extent of property damage to save resources during fire outbreaks.

3.1 Chapter's Article Details with Authors Contribution

Hitimana, E., Bajpai, G., Musabe, R., Sibomana, L., Kayalvizhi, J. "IoT Application for Spruce Fire Detection in Rwanda". In: Fong, S., Dey, N., Joshi, A. (eds) ICT Analysis and Applications. Lecture Notes in Networks and Systems 2022, vol 314. Springer, Singapore. https://doi.org/10.1007/978-981-16-5655-2_79.

Personal Contribution: The idea of writing this feasibility study arose out of a meeting with Gaurav Bajpai, Louis S. and Richard M. Formal analysis was done by B. Gaurav, I did methodology together with B. Gaurav; I implemented the application for data acquisition under supervision of Louis S. M. Richard brought the idea of sensor calibrations with B. Gaurav. I wrote the first version under guide with B. Gaurav. B. Gaurav, S. Louis and M. Richard performed paper

review and editing. I produced all the figures and tables. All authors read and agreed to the published version of the manuscript.

3.2 Introduction

Rwanda is a fast-growing economy country with increasing infrastructure development. Its cities are gradually expanding with city dwellers [111]. This leads to an increase in fire risks as more buildings are constructed. There are a growing number of people with inadequate knowledge of fire safety measures, regulations, and laws. This results in high exposure to fire hazards leading to increased vulnerability. The firefighting equipment installed in the building could handle small fire incidences. Most buildings are unequipped with upgraded fire safety equipment. Sensitization of the population on fire safety-related matters is irregular. Building codes for geographical identification based on fire safety regulations are not fully reinforced.

Real-time information is critical to fire emergency response for recovery measures. The collation, assessment, verification, and dissemination of information require an appropriate information management system. This system's decision-making provides external information that allows members of the public to make informed decisions to ensure their safety. Fire outbreaks have increased in recent times with 326 fires registered across the country from 2010 up to 2019. Investigation of causes of the fire outbreaks found that 61%, 22%, 9%, 4%, 3%, and 1% were due to short circuit causes, arson, road incidents, hazardous domestic activities, and chemical explosion respectively [112]. Fire outbreaks cause loss of life with property damage. Early detection of fire sources with effective countermeasures is an issue in fire prevention [113]. Efforts to develop fire management systems worldwide by applying information communication technology (ICT), are being conducted in practice. The utilization of ICT includes cloud computing with the IoT.
The main characteristics of the smart city are integrating scalability, interoperability, limited power consumption, multi-modal access, and robustness [114].

The main objective of this chapter is to detail the development of a highly efficient system that detects a fire and records the building occupants by notifying the stakeholders in real-time. This also cares about users' privacy by detecting them using an Infrared proximity sensor.

3.3 Related Works

The building monitoring system using Raspberry Pi, radio-frequency identification (RFID) technology, temperature sensor, and video monitoring compromises privacy [115]. It is capable of motion and disturbance detection at entry points along with a security alarm system having alerts containing pictures with privacy concerns. However, this system avoids static individuals such as people sleeping or in places that have no cameras such as toilets. It has been observed that cameras do not provide privacy [116]. A non-intrusive approach for indoor occupancy detection enhances smart environments by focusing on the detection, localization, and identification of people. Many techniques to detect human presence by using wearable devices and cameras also suffer from intrusiveness issues.

The development of an algorithm for the detection of carbon dioxide (CO_2) based occupancy was implemented. It detects CO_2 concentration monitored in buildings to assess indoor air quality. It has the drawback of detecting the concentration of CO_2 or other gases in places such as cooking areas. The use of different methods for inferring occupancies such as data streaming and video streaming [117] to handle ingesting along with analyzing with the IoT sensors. Different platforms such as Intel Edison, Raspberry Pi, and Arduino with sensor parameters including vibration, light, CO_2, temperature, and humidity, with live video streams were used. The reasons can be to improve space utilization, locate people during an emergency, or understand flows within buildings over time [118].

Personal detection techniques and emergency evacuation assistance were developed to minimize the loss of mishaps [119]. However, some corporations trained employees to be emergency response coordinators (ERCs) recording manual entries of occupants. Nowadays, emergency evacuation is linked to electronic attendance providing baseline numbers in the building. It scanned ID of a person to detect movement. This assumes that moving machinery is not interfacing too. The system has the drawback of detecting static people or visitors. Counting all people in the whole building or people based on some predefined zones [120] is to know the total number of occupants. The zones are defined using heating ventilation, and air conditioning (HVAC) zones, offices, or wireless fidelity (Wi-Fi) access point coverage areas. The granularity of the zones differs case by case.

The IoT connection control has four common methods: internet control (Wi-Fi), area network control (WebSocket), Bluetooth, and serial port. It has nodes connected with multi-hop to transmit

and reinforce continuous monitoring in real-time [121]. Mainly fire alarm systems use wireless sensor networks (WSN) [78]. The key characteristics of an IoT setup include the following: (1) instrumentation, (2) interconnectivity, (3) intelligence, and (4) facilitation of smart services through instrumentation and networking technologies. The IoT enables various devices that are attached to microsensor chips. Earlier, fire detection used visual analysis through video and image processing techniques to detect fires [122]. Nowadays, advances in sensor microelectronics are used in fire detection technology monitoring [123].

Traditionally, single-point heat along with smoke detectors was widely used for limited areas. In large rooms with high-rise buildings, smoke particles, and heat take a longer detection time. Detection of fires preferably uses video-based fire [124]. They can also perform high-rise building fire monitoring along with outdoor fire detection [125]. These detections use visual characteristics of fire, including color, texture, geometry, and flicker, with motion [126]. Since smoke is translucent, the edges of the high-frequency image frame lose clarity as an indicator of smoke. The distinction between different types of smoke is through the examination of changes in background tones, smoke pixels, and blurred backgrounds, with illumination segmentation. Establishing a sensor-based safe cooking kitchen environment requires in-depth risk analysis. Kitchen disaster risk consists of three factors: fire, burn, and intoxication. The sensor-based fire detection technology is highly accurate. It is easy to install the hardware and deploy the software. Integrating sensor and vision-based visual fire detection algorithms for fire color modeling with motion detection sensors seem to be an appropriate combination. Although it increases system overhead with complexity.

3.4 Proposed Methodology for Internet of Things Deployed Prototype.

In the system, the proximity sensor will detect building occupants. This is recorded in persistent database (Firebase) and data displayed in the dashboard with the process of sending messages through a short message service (SMS). Also, a buzzer and light-emitting diode (LED) shall be activated for warning and show direction exiting in the building. In case of smoke in a building, it has smoke sensors that will provide a warning light alert when surpass the threshold.

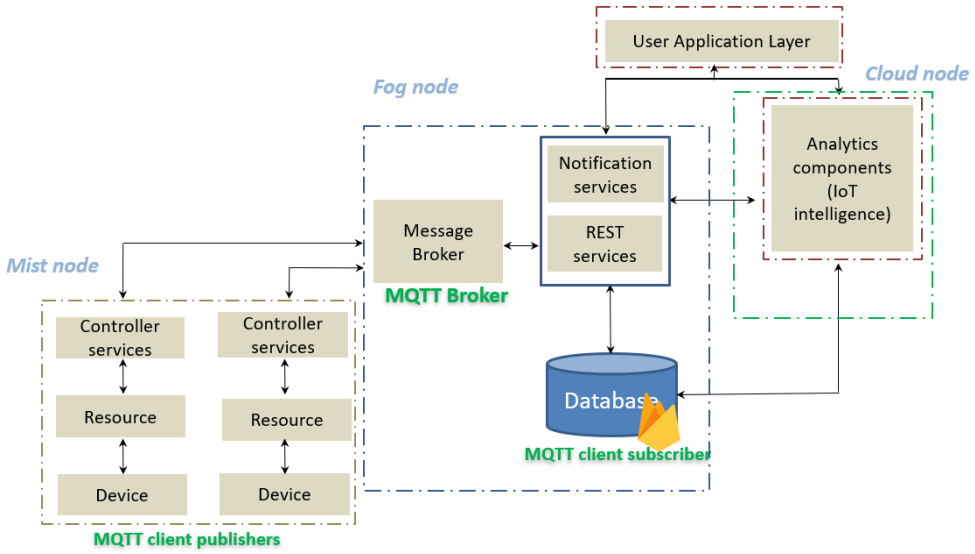


Figure 3. 1: The architecture of the system's overall diagram

The proposed system as shown in Figure 3.1 will enhance the accuracy of detecting building occupants by using an IR proximity sensor. It will ensure the privacy of people by knowing the location of each person in the building without using the camera. Consequently, the system provides an alert to deliver real-time information to the fire brigade. This reduces the number of injuries during a fire outbreak.

3.4.1 The Protype Design for the Proposed Internet of Things Detection Methodology

IoT-based applications use sensors for sensing the environment factors. NodeMCU Amica is an ESP8266 Wi-Fi Module-based development board. Proximity level is reported as the distance from the sensor to the closest visible surface. A smoke sensor can detect smoke by alerting through a sound alarm while LED gives an alert to the building occupancy. A piezo buzzer is a tiny speaker connected directly to an Arduino. LED is a semiconductor that illuminates when charged and satisfy the condition.

The use case identifies the users that are called actors for implementation. Figure 3.2 describes the actors of the proposed model. The system is implemented to notify concerned users. Sound alerts to notify the occupant of the threat to evacuate. Also, the LED will be blinking to direct the evacuation path to be followed during an event. Light indicators with sound alarm notifications are for occupants in range of the threat. However, the remote administration units will get SMS notifications for action to be foreseen. The information assimilated through data sent on the

dashboard generates analytics that is visualized promptly through dashboard management. Furthermore, the fire brigade will log in for information access about rescue during an event.

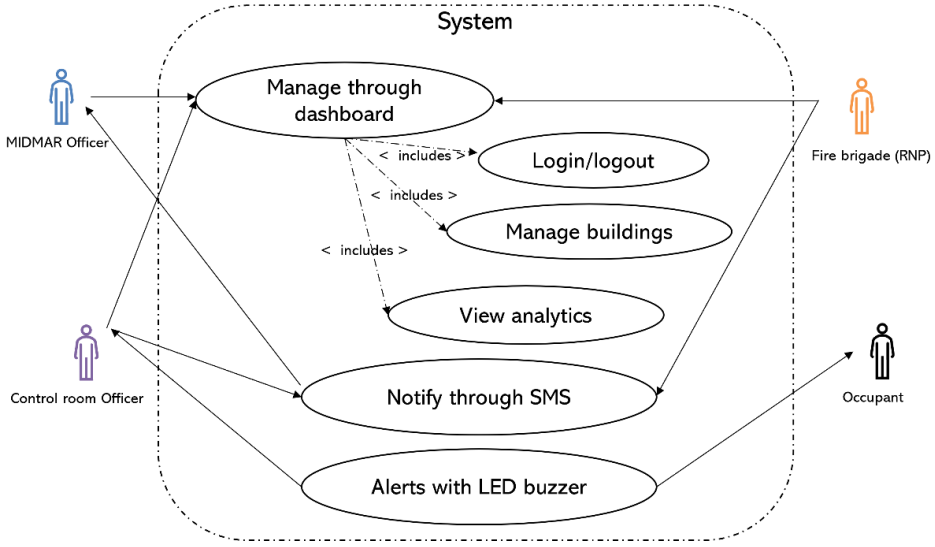


Figure 3. 2: Use case for the proposed systems.

The activity diagram in Figure 3.3 shows the data workflows with activities followed by actions made according to the choices taken.

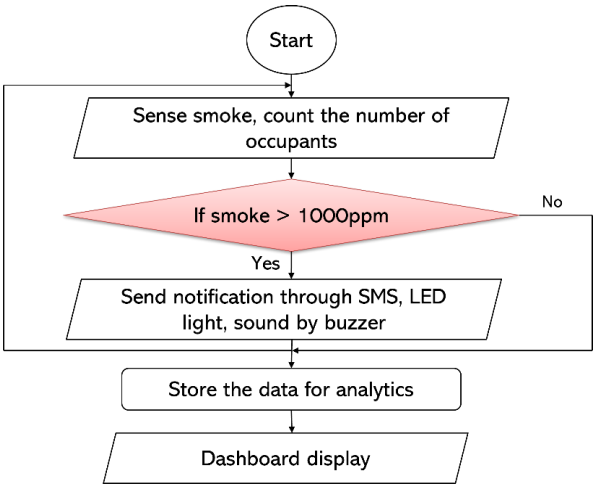


Figure 3. 3: Activity diagram for the proposed systems.

The sequence diagram of the proposed system describes sequenced interactions as shown in Figure 3.3 among classes in terms of the exchange of tasks over time. This sequence diagram contains seven (7) objects having different names.

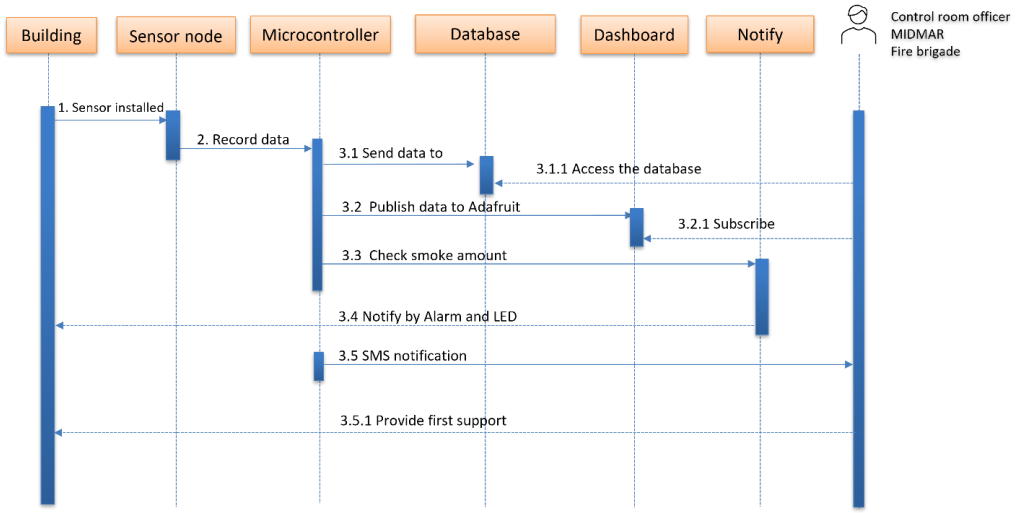


Figure 3. 4: Sequence diagram of the proposed system.

3.4.2 Acquired Results and Discussions

This research used IoT techniques to develop smart fire detection. The system architecture integrates the microcontroller, sensor elements, and actuators, and send notifications through SMS to concern stakeholders. Further, the data is stored in Firebase DB later to display events on the dashboard for rescue and evacuation of occupants in the building. This leads to data visualization through the Adafruit I/O Platform. The process diagram of the proposed model follows in Figure 3.5.

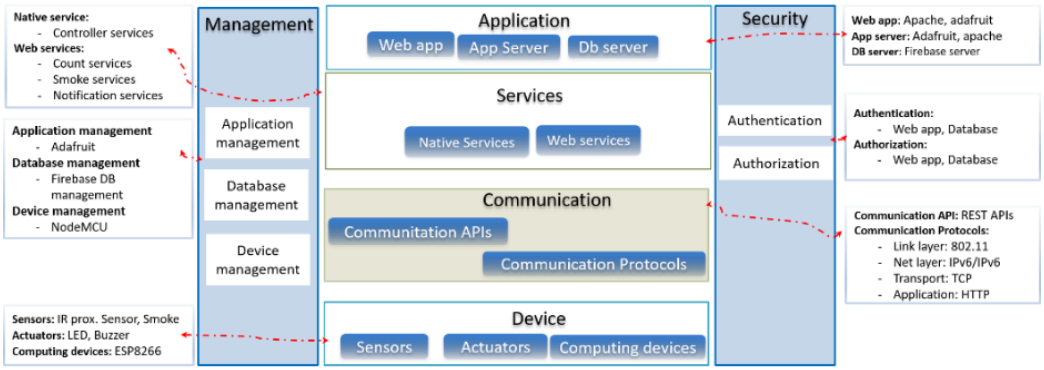


Figure 3. 5: Process diagram of the proposed model.

The system implementation contains six parts described in the IoT framework structure shown in Figure 3.5. It has an application layer, management layer, services, communication, security, and device layer.

The system developed and shown in Figure 3.6 collects data and stores them in the centralized database. Here, the client occupancy is detected with two sensors, proximity sensors, and smoke

sensors in every room in the building. This assists in calculating the exact number of people on-site.

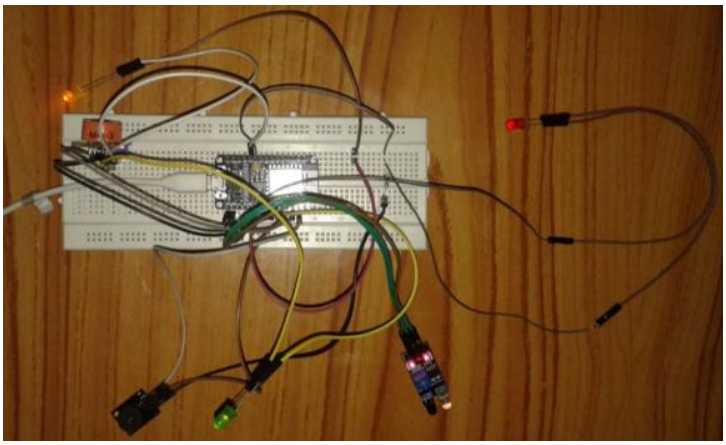


Figure 3. 6: System setup.

The smoke sensor detects the presence of CO_2 concentration if greater than 1000 ppm. The data is sent every 2 minutes to the cloud storage service as shown in Figure 3.7.

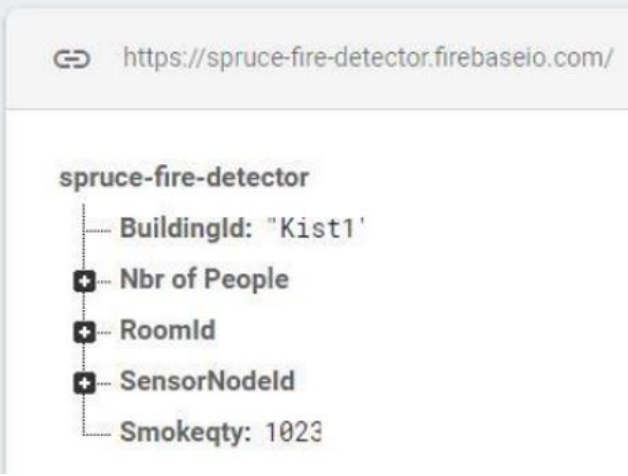


Figure 3. 7: Firebase data store.

The global system for mobile communications (GSM) module is used to alert the user by sending an SMS. In this research, the notification is sent from the microcontroller directly to administrators and the ERCs. The users access that real-time information by subscribing to the MQTT broker. MQTT allows them to subscribe to the system. MQTT is a lightweight publish/subscribe protocol. It allows administrators to publish and receive the message as the client after subscription.

In this section, the input from sensors is kept in a firebase like the number of people in a building from the proximity sensor including the amount of smoke from the smoke sensor.

Consequently, in case of a fire outbreak, the fire brigade will receive records of real-time information for occupants trapped. This database always records the data and the content that is received from the microcontroller.

This provides real-time information that is recorded in Firebase. Administrators feed information statistics to ERCs. The research proposes including a control room manager for each building. The fire brigade unit's focus is to quickly rescue victims depicted on the dashboard. It also archives the record for predictive analysis. Adafruit I/O platform is a cloud service used to manage traffic information in real-time. The administrators view real-time data online by using any PDA (Personal Digital Assistant) device as depicted in Figure 3.8.

Figure 3. 8: Sample real-time data for building occupants and smoke display dashboard.

Table 3.1 shows all information accumulated from the system, such as building Id, Room Id, NodeSensorId, building occupancy, smoke quantity, time, and status. The system analyzes the status of any spruce fire detection to ease the ERCs for automatic recognition.

Table 3. 1: Sample dataset generated by the system.

Building ID	Room ID	Sensor Node ID	Number of occupants	Smoke Quantity	Time	Status
50	21	34	1	650	1:02: 50	Normal
50	21	34	1	650	1:02: 52	Normal
50	21	34	1	650	1:02: 54	Normal
50	21	34	1	650	1:02: 56	Normal
50	21	34	2	750	1:02: 58	Normal
50	21	34	2	750	1:03: 00	Normal
50	21	34	2	750	1:03: 02	Normal
50	21	34	2	750	1:03: 04	Normal
50	21	34	2	890	1:03: 06	Normal
50	21	34	2	1000	1:03: 08	Fire
50	21	34	3	1060	1:03: 10	Fire

This chapter details the detection of spruce fires in buildings. The research objective of developing an IoT algorithm to detect fire and count the occupants which may contribute to the data sharing with the responsible entities is achieved. The system implemented was deployed to send a notification to the institutions in charge such as the fire brigade and building control room officers.

Future research work should focus on android applications for notifying using email over the cloud instead of using SMS notifications. Moreover, advanced prediction methods could be used for predictive analysis.

This section discusses the implementation of the IoT prototype to detect environmental parameters, while the next section details the full framework application with deep learning algorithms on the selected data.

Chapter 4

INTERNET OF THINGS (IoT) FRAMEWORK WITH DATA ANALYSIS USING DEEP LEARNING METHODS FOR OCCUPANCY PREDICTION IN A BUILDING

Many countries worldwide face challenges in controlling building incidence prevention measures for fire disasters. The most critical issues are the localization, identification, and detection of the room occupant. IoT along with machine learning proved to increase the smartness of the building by providing real-time data acquisition using sensors and actuators for prediction mechanisms. This research proposes the implementation of an IoT framework to capture indoor environmental parameters for occupancy multivariate time-series data. The application of the LSTM deep learning algorithm is used to infer the knowledge of the presence of human beings. An experiment is conducted in an office room using multivariate time series as predictors in the regression forecasting problem. The results obtained demonstrate that with the developed system it is possible to obtain, process, and store environmental information. The information collected was applied to the LSTM algorithm and compared with other machine learning algorithms. The compared algorithms are Support Vector Machine, Naïve Bayes Network, and Multilayer Perceptron Feed-Forward Network. The outcomes based on the parametric calibrations demonstrate that LSTM performs better in the context of the proposed application.

4.1 Chapter's Article Details with Authors Contribution

Hitimana, E.; Bajpai, G.; Musabe, R.; Sibomana, L.; Kayalvizhi, J. "Implementation of IoT Framework with Data Analysis Using Deep Learning Methods for Occupancy Prediction in a Building". Future Internet 2021, 13, 67. https://doi.org/10.3390/fi13030067.

Personal Contribution: The idea of writing this feasibility study arose out of a meeting with Gaurav Bajpai, Louis S., and M. Richard. Formal analysis was done by B. Gaurav, I did methodology together with B. Gaurav; I implemented the algorithms for data analytics under supervision of Louis S. M. Richard brought the idea of sensor calibrations with B. Gaurav. I wrote the manuscript under guide with B. Gaurav. B. Gaurav, S. Louis and M. Richard performed paper

review. I produced all the figures and tables. All authors read and agreed to the published version of the manuscript. All together worked on review feedbacks and agreed for submission.

4.2 Introduction

Several fields are involved in the influence of the advanced use of IoT techniques by taking advantage of the way these embedded devices offer efficient and affordable functionalities. The monitoring and actuating functionalities facilitate the development of various solutions in real-life [127].

Among the areas that have received great attention from the used IoT is smart building, which has been considered mostly for energy conservation and people's comfort [128, 12]. While people are in buildings something disturbing can happen. Those issues in the category of disasters like earthquakes, and fires take a huge toll in terms of both money and life even though most governments have special organizational units that handle them. When it happens, there are considerable expenses comprising construction costs, equipment, recruitment, retention, and training. As reported by the East African Community [9] on disaster risk reduction and management strategy in 2012–2017, it declares that abundant natural hazards happened.

Those hazards such as floods, droughts, urban fires, and earthquakes negatively contribute to the impacts of security of lives as well as the economies of their partner states [129]. Their observations mentioned the cause behind the fire hazards such as haphazard electric wiring, disregarding construction standards, accidents, and uncontrolled burning including bush waste materials burning.

Rwanda as an Eastern African country has managed to establish a law for the fire and rescue brigades under the Rwanda National Police [130]. All control strategies of the buildings in the capital/secondary cities are inspected regularly to handle fire incidents and other possible disasters. The law again enforces naming commercial buildings for better allocations for smart city management.

Most of the efforts have been done to fight against the loss of human lives the unexpected disasters causing the buildings to collapse or be burnt. The buildings require regular evacuation drills causing expenses, and loss of productivity. Still many occupants are not experienced with the usage of firefighting equipment. Any automation to assist in these drills during real-time actual evacuations would mean saving cost, time, and life [131].

As far as the occupancy monitoring of a building is concerned, it can be categorized into group-based and individual monitoring. The group-based occupancy method refers to the estimation of aggregated space occupancy, while the individual based is to track each specific occupant's location. In the latter case, the system extracts the identity of an occupant to provide tracking information showing the locations of a past time-period [132, 133]. Group-based occupancy monitoring is divided into two categories according to occupancy granularity. The first is occupancy detection, referring to the detection of presence in a controlled environment. The second is occupancy estimation referring to the inference of the number of occupants. Occupancy estimation refers to the estimation of the exact number of occupants along with an estimation of occupancy density. Occupancy detection is the most feasible task but the estimation of the exact number of occupants in space is the most challenging task.

To respond to occupancy detection, intrusive methods were used, like cameras and wearable devices, but these methods suffered from the privacy of occupants, later, non-intrusive mechanisms were adopted by using sensors to capture the human presence based on different considerable characteristics. It is observed that the deployment of a sensor network in the whole building is cost-effective. This idea led researchers to start focusing on the estimation and prediction methods. At this point, identifying good features considered to increase accuracy depends on two essential criteria. The first is to select a high-accuracy algorithm. The second is the computation time. In the case of fire rescue, human life and resources need accurate methods to be efficient [134].

In case of emergency, especially in densely populated buildings with complex floor plans, it is a challenging task to evacuate people to a safe place. The sensor data must be sent in such a way that it will ease the information complexity. This research is mainly reflecting on the building occupant prediction, as it is tackling the problem of indoor victims trapped when there is a natural disaster (fire or earthquake). The proposed mechanism is to use environmental parameters captured using a designed IoT system. The IoT time series multivariate data captured in a timely manner allows the LSTM deep learning model to infer the occupant of the specific rooms so that the fire brigade can respond automatically without any delay.

The main objective of this chapter is to detail the development of an IoT system to gather time-series-based data from an indoor building. Further is to aggregate those data in the cloud storage by using a deep learning prediction model with multivariate predictors resulting to draw

the inference of building occupancy. The motivation for choosing deep learning is (1) the use of ordered time-series data requiring the previous sequential information to work on the current and next sequences, (2) keeping the memory state of the previous layer, and (3) the expectation of the huge amount of data as they are accumulated in every one-minute time interval.

The experiment used LSTM as a deep learning model to predict the building's occupancy by considering multivariate parameters that infer the knowledge of human presence in a room. The prediction model is tested using the acquired sensor data captured by the developed IoT system deployed on the University of Rwanda site.

It is composed of four main components such as (1) the sensor nodes deployed in different rooms of the building with wireless capabilities to subscribe to topics and sent real-time messages (data) through the broker (EMQ) to the database. (2) The broker's role is to allow real-time data transmission from the sensor node to the persistent database (resides in UR-CST Datacenter) which acts as a subscriber to the broker. The data acquired from the rooms are CO_2, temperature, humidity, and occupancy. (3) The data stored in the persistent database are ingested into the LSTM model for prediction purposes. The model is selected among many tested machine learning models. (4) The data such as the location of the incidences and the amount of data at that location, are visualized through the responsive user dashboard.

Figure 4.1 is the view of the proposed framework for gathering room parameters such as temperature, humidity, lighting conditions, CO_2, and occupancy.

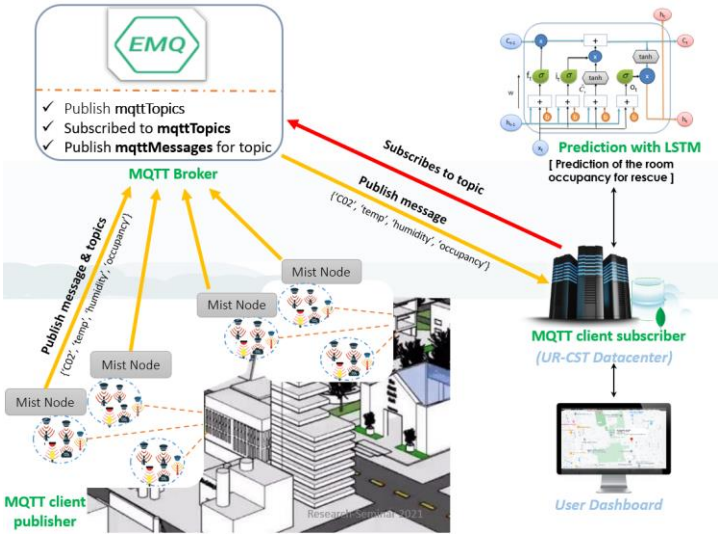


Figure 4. 1: Proposed architecture of data gathering.

The purpose of this research is to provide data acquisition architecture using IoT technology. It also conducts a comparative analysis of other occupancy prediction models.

4.3 Materials and Methods

4.3.1 Node Layer

The conducted research had been implemented in the University of Rwanda building located in Kigali city Rwanda. The developed protype consisted of the selected sensors according to the designed architecture to acquire environmental parameters to be used for further prediction. These sensors are namely temperature sensor, humidity sensor, lighting sensor, CO_2 sensor, and the passive infrared sensor also known as a motion sensor for occupant detection as listed in Table 4.1

A global positioning system (GPS) is used to locate the building. Besides, all buildings with their corresponding rooms are labeled to allow precise locations for the quick rescue of victims done by emergency response coordinators.

Table 4. 1: Sensors used in the development of prototype.

Component	Characteristics	Manufacturing
DHT11	Temperature, humidity	Adafruit
PIR Motion	Occupancy	Adafruit
MQ-135	CO_2	Adafruit
LM393 Light Detector	Light	Adafruit

All sensor modules are connected to an ESP 8266 microcontroller NodeMCU. It has an inbuilt Wi-Fi module to allow data transmission to the gateway followed to the cloud respectively as shown in Figure 4.1. Table 4.2 describes each component used with its characteristics in this research context.

Table 4. 2: Components of the developed architecture

Component	Characteristics
Mist node	- Composed of sensors and a microcontroller, - Quick decision for controlling actuators, - Allow data access control and privacy mechanism.
Fog node	- Located in the University of Rwanda Datacenter, - Allow real-time data storage, data processing, data monitoring.
Cloud node	- Uses third-party cloud services, - Data is sent there for public access and decision making, allow prediction analysis

The reason for choosing this architecture is to make a flexible framework to decide at any point in time based on the criticalities of the situation. In case of fire, the mist node takes a quick reaction by activating the actuators in a specific room, for example, the sprinklers act by spraying water. At the fog node, all possible collected data is stored in the data center of the institution for real-time analysis/data processing by reducing the latency of the wide-area network.

At the cloud node, third-party cloud services are selected, allowing public data access for decision-making followed by prediction analysis. The mist node had been deployed in a selected room for 15 days. Data was transmitted using Wi-Fi technology to the back-end server database within a one-minute time interval. The collected data are extracted into comma-separated value (CSV) format for prediction purposes.

4.3.2 Global Positioning System (GPS) Sensor

GPS is a satellite-based radio navigation system. GPS-enabled devices are used as navigation and orientation tools as well as to capture traveled routes [135, 136]. It has been used in several sectors such as travel mode detection [137], physical activity detection and informing classifiers for detecting physical activity types [138]. GPS data has composite types of parameters that are used to allocate a specific object relying on the earth. GPS data include latitude, longitude, date, time, horizontal dilution of precision, vertical dilution of precision, number of satellites, altitude, and instantaneous speed [135].

The latitude and longitude parameters captured by the GPS allow the localization of the affected building. To alleviate the problem of locating victims in the vertical direction, the sensor node is labeled based on the room number along with the building's floor number indicating the deployed microcontroller.

4.3.3 Message Queuing Telemetry Transport Protocol

Message queuing telemetry transport (MQTT) is an application layer protocol adopted for different applications such as monitoring data [139], user energy management [137], notification systems [140], handling mobility [141], heart and electrocardiogram monitoring [142, 143], it was adopted in the past by large-scale companies for messenger application [144, 145].

Building an IoT system requires IoT data transmission using IoT communication middleware. A topic is identified by a unique string. That string represents the hierarchy of topics themselves. All information is organized on different topics. Each client connected to the broker could be either

an information subscriber or publisher. As the protocol exchanges data through topics while all devices with the same topic are connected to the broker, its location is important for MQTT broker based IoT middleware [146].

Every time a publisher (mist node) posts a new message on a topic (sensor data), the MQTT broker (EMQ) forwards that message to all subscribed clients (MongoDB) to that topic. In this way, publishers and subscribers could send or receive data ignoring the existence of the locations of other clients. Another important MQTT feature is the possibility of setting different Quality of Services (QoS) in client connection to the broker. A different level of QoS determines a different security level regarding the delivery of the message to the destination [147, 148].

The MQTT protocol has been chosen from HTTP even though they have a considerably smaller footprint. The MQTT protocol is suitable for resource-constrained environments. All MQTT-based brokers have similar or comparable abilities. The mosquito broker by default does not provide security for its messaging scheme. Here authentication information is sent in plaintext therefore, it requires security mechanisms to protect the transferred information [149].

In this context, from the available MQTT brokers explored, the two open sources (Mosquito, and Erlang MQTT) had been explored, the latter is chosen. EMQ broker provides a scalable, reliable, enterprise-grade PUB/SUB MQTT message Hub for IoT. It is written in Erlang/OTP under Apache Version 2.0 that is cross-platform, which was deployed on different operating systems and even Raspberry Pi [150]. EMQ implements both MQTT V3.1 and V3.1.1 protocol specifications and supports MQTT-SN, CoAP, WebSocket, STOMP, and SockJS at the same time [151].

In the framework displayed in Figure 4.2, the environmental parameters are acquired from a room, including the room name, floor number, and the geo-location of the building. This data is sent to the EMQ broker followed by the MQTT client subscriber.

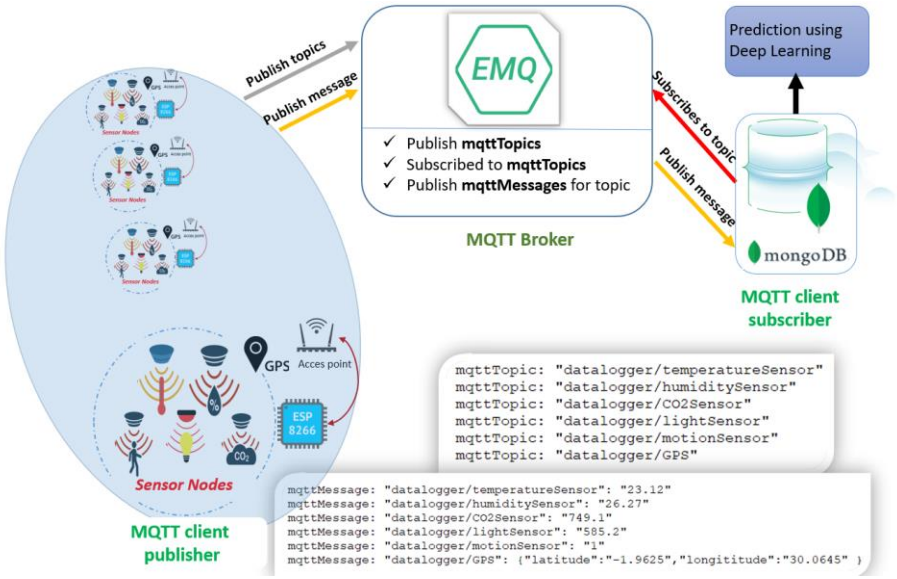


Figure 4. 2: The proposed framework for data acquisitions and prediction mechanism.

4.3.4 Data Storage

Mongo is an open-source schemeless, not only SQL (NoSQL), but document also-based database that stores and retrieves documents as extensive markup language (XML), JavaScript Object Notification (JSON), binary JSON (BSON) [152]. In this research, Mongo is selected as it can support the variety, volume, and velocity features of big data. The system implemented includes horizontal scalability, linearization, high availability, and fault tolerance. The database server is responsible for recording room physical environmental parameters every 1-minute resulting in a huge amount of data. The retrieval is simple for the different real-time user interfaces.

4.3.5 Prediction Model Based on Long Short-Term Memory Networks

RNN methods are a special type of neural network designed for sequence problems. They have been used to create sequential information data for deep learning applications such as object tracking [153], image classification [154], speech recognition [155], and language translation [156]. One of the RNN types is LSTM. LSTM overcomes the limitations of RNN with the lack of full information on how to train the network with backpropagation. It also enables stopping gradients from vanishing or exploding during training.

A common LSTM unit shown in Figure 4.3 is composed of a cell, an input gate, forget gate, and an output gate. The cell remembers values over arbitrary time intervals. The three gates regulate the flow of information into and out of the cell.

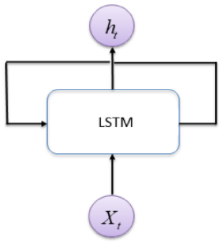


Figure 4. 3: Common LSTM unit without details.

The LSTM cell as shown in Figure 4.4 is composed of inputs, outputs, vector operators, and nonlinearities that are responsible for keeping track of the dependencies between the elements in the input sequence. The input gate denoted by I_t controls the extent to which a new value flows into the cell. The forget gate controls the extent to which a value remains in the cell. The output gate denoted by O_t controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit. The candidate gate denoted by $\check{C}_t$ controls what data to write to the cell state.

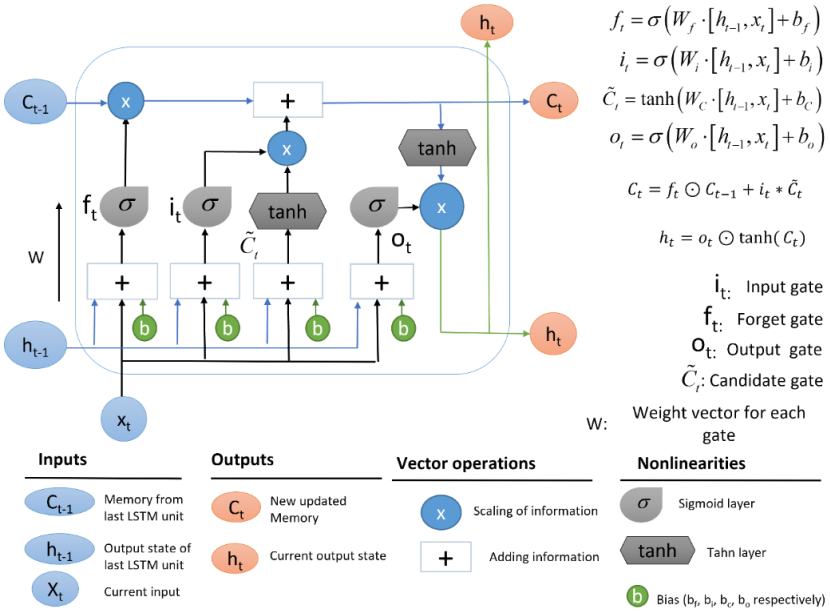


Figure 4. 4: LSTM cell gate complexity.

The inputs are of three types. The first is C_{t-1} that indicates memory from the previous/last LSTM unit. The second is h_{t-1} which indicates the output state of the previous/last LSTM unit. The third is X_t indicating the current input. The outputs are of two types. The first is C_t which indicates new updated memory. The second is h_t that indicates the current output state. The vector operations

are of two types. The first is x indicates the scaling of information while the second + indicates adding information. The nonlinearities are of two-three types. The first one is the sigmoid layer denoted by σ. The second is the tanh layer denoted by tanh. The third is the bias denoted by b_i.

The activation function of the LSTM gates is often a logistic sigmoid function denoted by σ. There are connections into and out of the LSTM gates, a few of which are recurrent. The weights of these connections that need to be learned during the training of the model determine how the four gates inter-operate.

The first step in LSTM is to decide the information that needs to be discarded away from the cell state. This decision is made by a sigmoid layer called the forget gate layer. It takes h_{t-1} and X_t, and outputs a number between 0 and 1 for each number in the cell state C_{t-1}. Here, 1 represents keeping it while 0 represents discarding it.

The next step is to decide the new information to be stored in the cell state. This has two parts. First, a sigmoid layer called the input gate layer decides the values to be updated. Next, a tanh layer creates a vector of new candidate values, $\check{C}_t$, which could be added to the state. The next step is to combine it and $\check{C}_t$ to create an update to the state.

The next step is to update the old cell state, C_{t-1}, into the new cell state C_t. In order to obtain a new cell state C_t, the old state is multiplied by f_t, forgetting the things decided to be forgotten earlier. Then add it $* \check{C}_t$. These are the new candidate values, scaled to update each state value.

This output will be based on the cell state but will be filtered. The sigmoid layer decides the parts of the cell state to output. Then the cell state is put through tanh (to push the values to be between −1 and 1) and multiplied by the output of the sigmoid gate so that the decoded output is achieved.

The operator 0 presented in the formula for C_t and ht denotes element-wise multiplication (labeled as the scaling of information in the diagram with a multiplication symbol as shown in Figure 4.4). Thus, the output at each time step depends on previous inputs and past computations.

In the context of occupancy prediction from the acquired data as the research is proposing the use of multivariate features, the input of data features is denoted as $X = (x_1, x_2, x_3, x_4, x_5)$ where 1, 2, 3, 4, and 5 represent temperature, humidity, CO_2, occupancy, and date respectively. The hidden state of memory block $h = (h_1, h_2, h_3, ..)$, depends on the number of LSTM state networks used with the real output sequence $O = (o_1, o_2)$ (where 1 and 2 denote occupant and time

respectively) are been iteratively calculated by the equations mentioned in Figure 4.4. In case the data needs iteration bias is added.

In the context of environmental time-series data, the prediction process, and the construction of an LSTM neural network shown in Figure 4.4 using the LSTM series of the cell are unfolded in Figure 4.5.

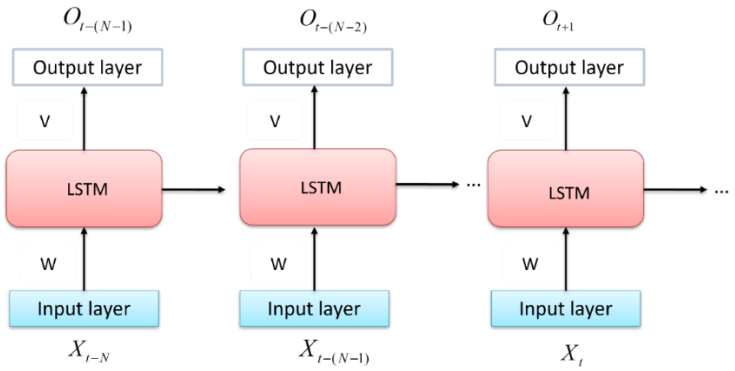


Figure 4. 5: The structure of series prediction LSTM neural network.

At time t − 1, the input of the network is the observed historical data X_{t-1} and the output is the predicted future data where X is the observed occupancy time sequence. There are two weights namely W stating the weights for transforming X to LSTM hidden state (not prediction) while V states the weights for transforming LSTM hidden state to prediction. Figure 4.5 describes the series of LSTM gates to construct the prediction architecture.

4.4 Results for the Implemented Model

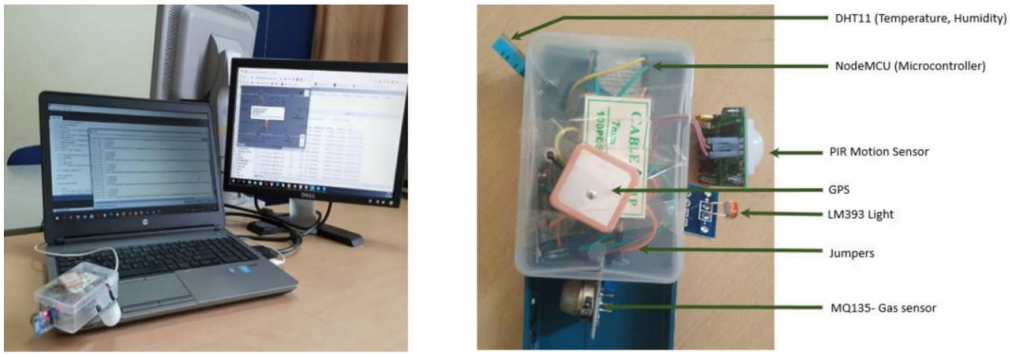
4.3.1 Data Acquisition

In the experimental setup configuration, the developed system including humidity, temperature, CO_2, light, and proximity sensors is implemented in a working office and gathers data for 15 days. The sensors are connected to an ESP8266 microcontroller board with its WiFi-enabled module enabled to send data to the locally configured MongoDB database server. The selected indoor environmental parameters are considered by different researchers to detect the presence of human beings. However, the key additional parameter is the use of a proximity sensor to detect any human presence along with the other captured variables. The geolocation information is captured to allow localization of the victims in case of a fire or any incidents that occur. The

output of the system displays on the web interface linked to the google map of the experimented building location.

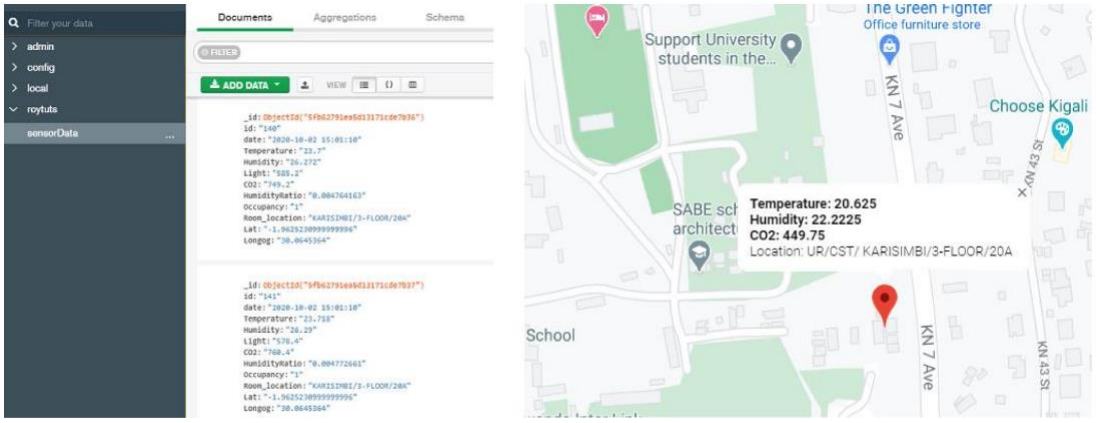
The presence of human beings is indicated by the increase of CO_2 as it is the gas produced during exhalation [157]. The other is also used to infer the knowledge of human presence in a specific room. The experiment did not consider the windows open or closed which can increase or reduce the amount of CO_2 concentration in the room. The position of the prototype was centered in the room. The room is a shared room with a proximity sensor placed near the door to capture any movement of humans.

Figure 4.6 shows the developed system in its operational mode during the testing phase. The data are sent to the back-end database in 1-minute time intervals to avoid redundancy of similar records. Figure 4.7 shows the close-up of sensor node localization on the google map with minimum average parameters.



(a) (b)

Figure 4. 6: (a) Prototyping setting up for data acquisition. (b) Closeup side of the data acquisition system.



(a) (b)

Figure 4. 7: (a) Sample of captured data appealing on MongoDB user interface. (b) Close-up geographical location map of the sensor node placement with some parameters omitted.

4.3.2 Preparation of Data for Multivariate Forecast Model

The time series acquired data from the experiment to the gateway server are converted to CSV format to be used in LSTM as input data. Before LSTM data preparation in Anaconda package of Python, a separate environment was created by the installation of Keras with Theano, as TensorFlow proved the slowness of the system used for implementation. Scikit-learn, Pandas, NumPy, Seaborn, and Matplotlib libraries are also installed for data preprocessing, visualization, and machine learning algorithms implementation support. Preprocessing of the data was conducted as an initial phase for the prediction that is required for missing values tracking and elimination. At this step, the time-series data is to fill the missed values. The interpolation of the variables before and after a timestamp is used for missed values. Table 4.3 shows the sample of the first five rows retrieved using the head() method in pandas among 19,879 records of the dataset.

Table 4. 3: Sample captured data in a selected room.

	date	Temperature	Humidity	Light	CO2	Occupancy	Room location	Lat	Long
0	02/02/2020 14:19	23.7000	26.272	585.200000	749.200000	1	KARISIMBI/FLOOR/3F-20A	-1.92197	30.149986
1	02/02/2020 14:19	23.7180	26.290	578.400000	760.400000	1	KARISIMBI/FLOOR/3F-20A	-1.92197	30.149986
2	02/02/2020 14:21	23.7300	26.230	572.666667	769.666667	1	KARISIMBI/FLOOR/3F-20A	-1.92197	30.149986
3	02/02/2020 14:22	23.7225	26.125	493.750000	774.750000	1	KARISIMBI/FLOOR/3F-20A	-1.92197	30.149986
4	02/02/2020 14:23	23.7540	26.200	488.600000	779.000000	1	KARISIMBI/FLOOR/3F-20A	-1.92197	30.149986

4.3.3 Collinearity of the Data

In the assessment of the correlation between the variables for better prediction, the collinearity is computed to check how they are associated with each other. In this operation, the heatmap function of the seaborn library of Python is used. Figure 4.8 shows how data are correlated for the selected Karisimbi building in the experimental office, omitting some variables (geo-location data).

Figure 4.8(a) some parameters are correlated in the sense that gives an insight into their relationship. The fully correlated data has a value of 1, while the worse correlated data has around zero value. The acquired data shows an obvious correlation between the occupancy variable and the light data. Moreover, there is a better correlation between occupancy with CO_2 and temperature parameters respectively. Figure 4.8(b) describes how all parameters are related to occupancy data.

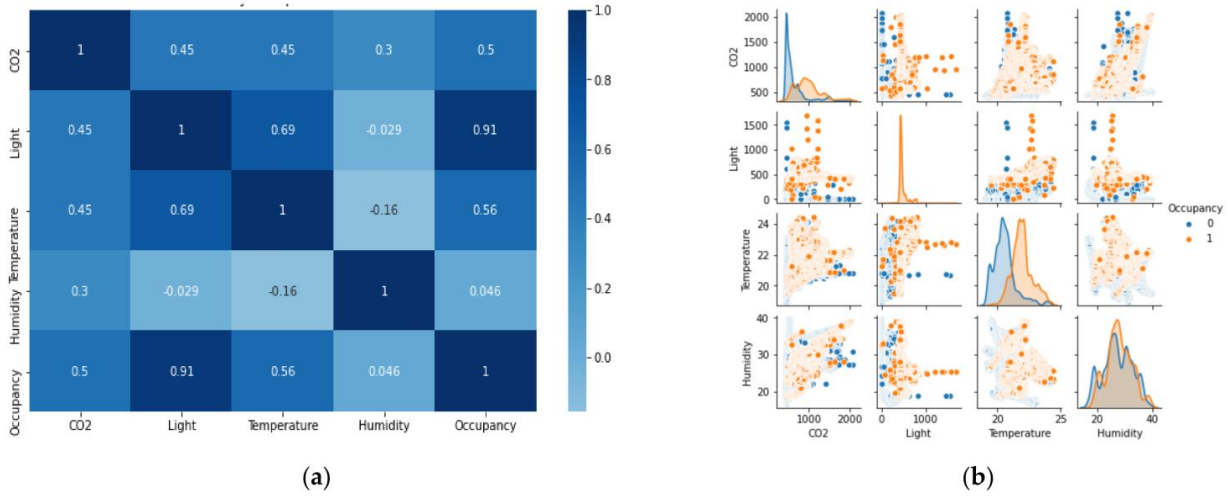


Figure 4. 8: Relationship between variables: (a) Showing the collinearity of the data set; (b) Describing the pairplot of the data set used.

4.3.4 Data Preparation for Long Short-Term Memory Networks Model

The occupancy dataset preparation for the LSTM model must be framed as a supervised learning problem. The data are multivariate data that require input variables to be normalized [158]. The data set is composed of multiple variables of different units. The reason for the re-scaling technique to be used as a normalization technique is to stretch and squeeze the values in the datasets to fit on a scale from 0 to 1. The model setup is considered to predict the occupancy at the next hours from the previous measurements at the prior time steps of five input variables used [temperature, humidity, light, occupancy, and CO_2] in the experiment.

The dataset is split into two parts which are train and test sets with 70% and 30% respectively. Then the train and test are split into input and output variables. The LSTM model expects 3D dimension variables namely [samples, timesteps, features] reshaped from input variables. Different configurations are done progressively, which is to define the first hidden layer with 1 neuron for predicting occupancy. The input shape will be a 3-times step with 5 features. The efficient Adam version of stochastic gradient descent is used with the Mean Squared Error (MSE) loss function. This research used 50 training epochs with a batch size of 64 of four iterations for a better-fit model. The research explored the appropriateness justification of deep learning towards the usage of automatic feature extraction. The adoption of real-time experimental data is something that was inspired to adopt in this research [159]. The importance of multivariate or multimodal data has been explored [160].

4.3.5 Performance Metrics

In this research, the evaluation of the experiment was conducted using Naïve Bayes Classifier, SVM, and Multilayer Perceptron Feed-Forward Network against LSTM, which is one of the RNN methods LSTM for occupancy prediction. In each experiment, the results were assessed by the following evaluation measures such as root square score, mean absolute error, root means squared error, accuracy, and standard deviation. The modeling scenario used in this research is considered a regression problem that is a set of statistical processes for estimating the relationships between a dependent variable (often called the outcome variable) and one or more independent variables (often called predictors, covariates, or features).

4.3.5.1 Root Square Score (R^2)

The performance metric of the model is evaluated using a root-squared score. It is the coefficient of determination [161], that represents the proportion of variance that has been explained by the independent variables in the model. It provides an indication of the goodness of fit and therefore a measure of how well-unseen samples are likely to be predicted by the model through the proportion of explained variance. The metric has crucial importance in measuring the efficiency of the model which means that its values range from the closed interval of zero to one [0, 1]. Here zero (0) illustrates that the proposed model does not improve the prediction over the mean model while one (1) implies the perfect prediction.

$\hat{y}_i$ is the predicted value of the i[th] sample and y_i is the corresponding true value for the total samples, the estimated R^2 is defined as follows.

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (1)$$

4.3.5.2 Mean Absolute Error (MAE)

The mean absolute error is a risk metric corresponding to the expected value of the absolute error loss. It measures the distinguished values between two continuous variables [162]. If $\hat{y}_i$ is the predicted value of the i[th] sample and y_i is the corresponding true value for total n samples, then the MAE estimated over n samples is defined as follows.

$$MAE(y,\hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}|\, y_i - \hat{y}_i \,| \quad (2)$$

4.3.5.3 Root Mean Squared Error (RMSE)

The RMSE is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are. RMSE is a measure of how spread-out these residuals are [163]. In other words, it says that the concentrated data is around the line of best fit. RMSE is commonly used for experimental results verifications [159]. A framework capable of solar power forecasting in a smart grid is proposed [164]. It is achieved through Autoregression (univariate) framework, and the optimization is achieved through RMSE. In this research incorporation of multivariate data analysis was done for better result accuracy.

If $\hat{y}_i$ is the predicted value of the i[th] sample and y_i is the corresponding true value for the total n samples, then RMSE estimated over samples are defined as follows.

$$RMSE(y,\hat{y}) = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2} \quad (3)$$

4.3.5.4 Standard Deviation (SD)

It is the measure of the dispersion of statistical data. Dispersion computes the deviation of data from its mean of average position. A large standard deviation indicates that the data points can spread from the mean and a small standard deviation indicates that they are clustered closely around the mean. Let the sample size be n with a mean of $\bar{Y}$, the standard deviation (SD) is calculated using the following equation:

$$SD = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1} | y_i - \bar{Y} |^2} \quad (4)$$

These performance measures give the indicators for the determination of the effectiveness of the model for forecasting the daily stock index [165-167].

A new forecasting model is proposed to outperform conventional autoregression and vector autoregression models [168]. This was made possible by incorporating the least absolute shrinkage and selection operator (LASSO) framework, as the case study involves sparse placements of wind power. This paper has inspired our future work when we are planning to expand the work across various places in the cities of Rwanda. The importance of spatio-temporal autocorrelation arriving from various data sources will be an added advantage to avoid inaccurate predictions [169]. The application explored is the energy domain, as mostly in these domains many papers have usually explored only temporal data. This work is reflected in this current research work through the adoption of spatiotemporal data in our data set.

The change detection in the application that involves streaming data is significantly described [170]. The approach has improved the quality of results by overcoming false positive detection, and additional changes in new incoming data. The research will be crucial in the future when deployed across cities, which creates the possibility for more and more new data, whereby the training and test datasets need to be rewired accordingly.

4.3.6 Training Results

The learning process for the recurrent neural network that uses LSTM cells is configured as an iterative process of the optimization of the weights. The weights are updated in each epoch. Once the training starts, the aim is to generate predictions by minimizing the loss function. The performance of the network is then evaluated on the test set. The training process ends when the error on the validation set begins to increase as this could mark the beginning of a phase of overfitting the data.

The model training and validation loss of the implemented LSTM recurrent neural network is presented in Figure 4.9. The observation shows that as the number of epochs increases the training model becomes the best fit.

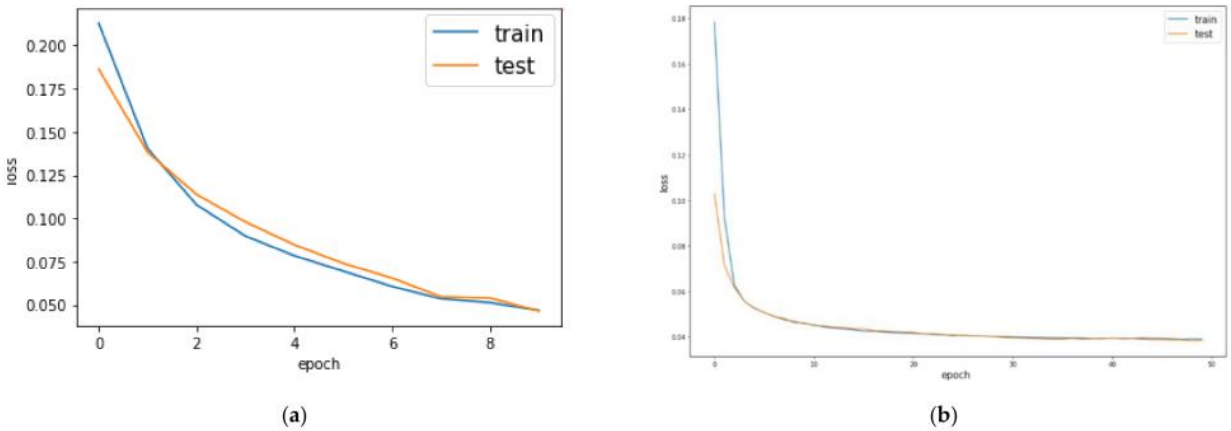


Figure 4. 9: LSTM model training versus test loss graph: (a) showing the training model with 8 epochs, (b) showing the training model with 50 epochs.

4.3.6.1 Comparative Analysis of Machine Learning Results

The results of the LSTM model are compared to two machine learning algorithms and one deep learning algorithm tested against the same dataset to evaluate the efficiency of the proposed model.

Table 4. 4: Model parameters optimal values in the experiment.

Parameter	Optimal Model Values for Occupancy Dataset
Train dataset lot	70%
Test dataset lot	30%
Input layer	1
LSTM cells	2 cells
Activation function	Rectified Linear Unit (ReLu)
Dropout wrapper	0.4
Dense Layer	1
Optimizer	Adam
Number of Epochs	50
Batch size	100
Look back window	8
Loss function	MAE

Figure 4.9 shows the training and test loss of the proposed LSTM model at optimal parameters values described in Table 4.4. The diagrams show that the train and test losses decrease for larger epoch values and that train as well as test losses both stabilize at closer points. The loss instability in the occupant model resulted from data complexity due to the continuous changes of light intensity during one or more combinations of the daylight or/and the windows open.

Figure 4.10 describes the actual and predicted values of the selected parameters such as temperature, humidity, CO_2, and light from the LSTM model described in Figure 4.4. As shown in the legends the true values are in orange while the predicted values are in blue.

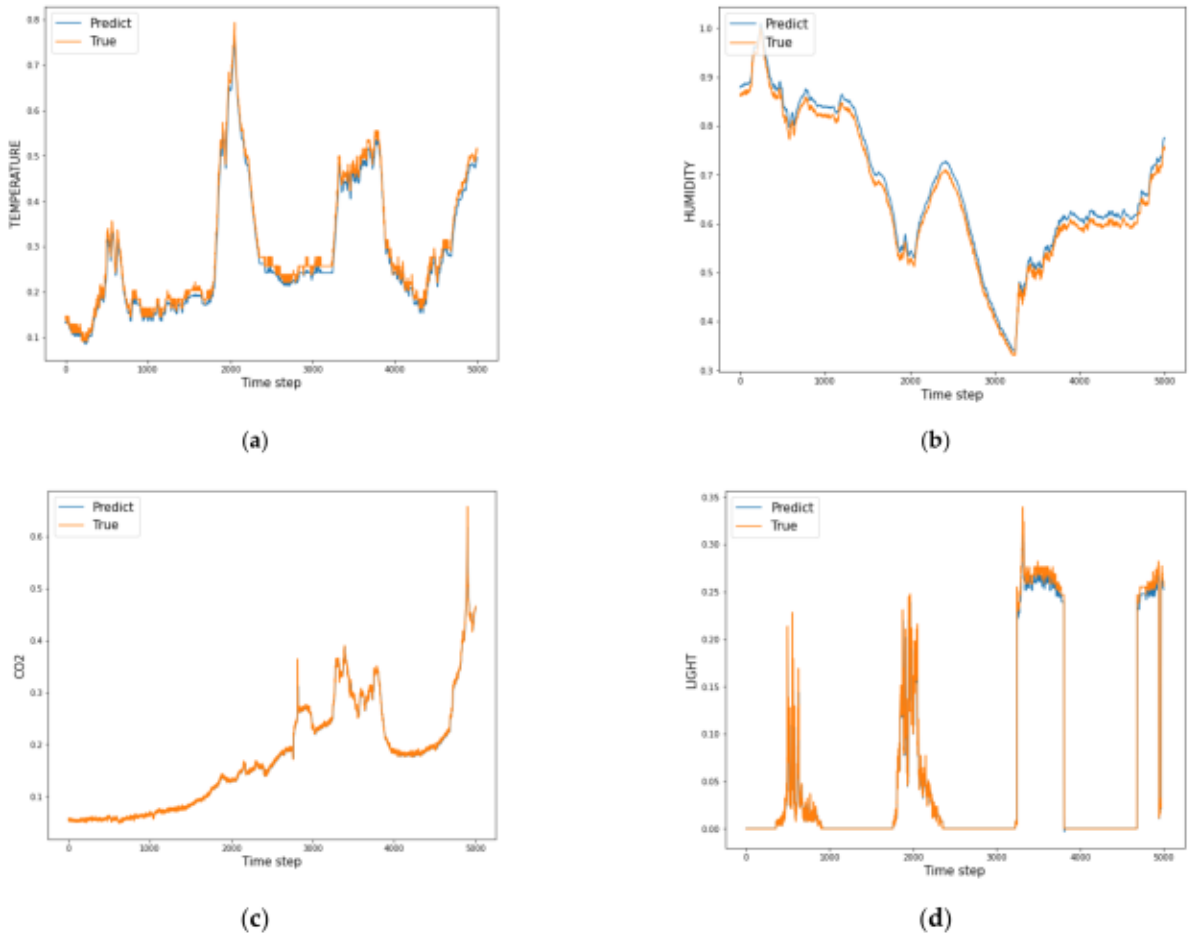


Figure 4. 10: Actual and predicted results: (a) for temperature, (b) for humidity, (c) for CO2, and (d) for light models.

Table 4.5 presents the comparative minimum error values as well as model performance accuracy of the proposed model with other models (Naïve Bayes Classifier, Support Vector Machine, and Multilayer Perceptron Feed Forward Network). The results show that LSTM has a high accuracy compared to other models and the error values are minimized in comparison to them.

In the selected model to compare the obtained errors, the minimum model loss that is attained on the hyperparameter values is shown in Table 4.4.

Table 4. 5: Performance measure values.

Approaches	Accuracy	MAE	RMSE	R^2	SD
LSTM	0.968	0.02317	0.08397	0.9566	0.347
Naïve Bayes Classifier	0.867	0.13294	0.36461	0.0908	0.4626
Support Vector Machine	0.822	0.17785	0.42172	−0.2163	0.0000
Multilayer Perceptron FFN	0.967	0.03193	0.17871	0.7815	0.44669

Table 4.5 is described in another form to represent the algorithmic comparison. The graphical representation is shown in Figure 4.11.

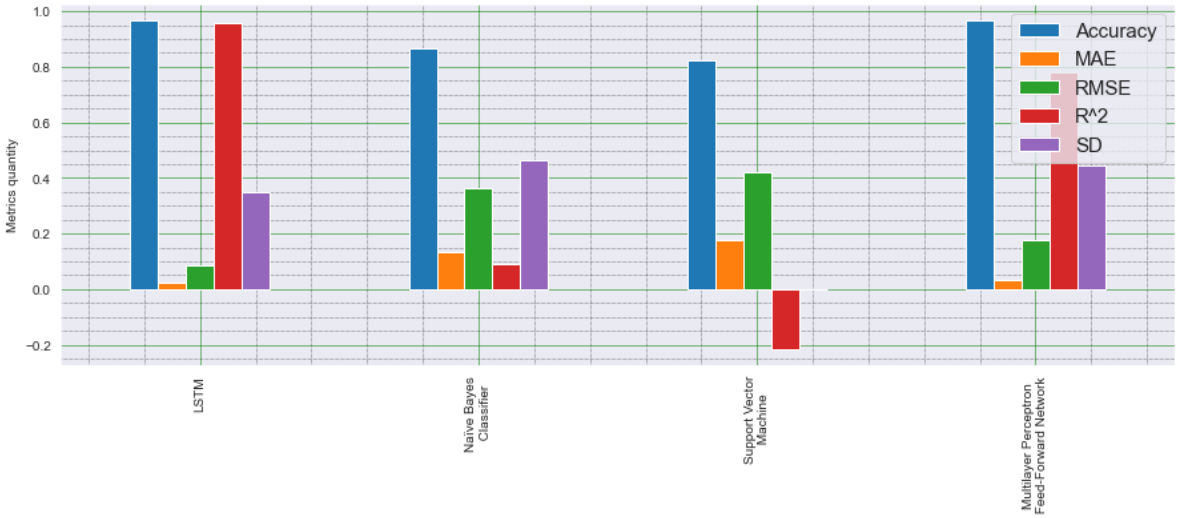


Figure 4. 11: Prediction results: Comparative analysis of LSTM with Naïve Bayes Classifier, Support Vector Machine, and Multilayer Perceptron Feed-Forward Network.

In Figure 4.11, the LSTM network presents better performance than the other selected algorithms namely, Naïve Bayes, Support Vector Machine, and Multilayer Perceptron Feed-Forward Network in all error measures. The predicted RMSE of those compared machine learning algorithms is 0.36, 0.42, and 0.17 respectively, while the predicted RMSE of the proposed model is 0.083. The prediction result of those compared machine learning algorithms enhanced to 77.57%, 80%, and 53.37% respectively compared to the deep learning recurrent neural network LSTM model. While the root means squared error (R^2) of the LSTM model is 95% indicating to be amongst the best. The support vector machine algorithm is showing the worst results that seem like it is not responding to the data set parameters used in this research. Multilayer Perceptron FFN results are presentable with its accuracy almost like LSTM as both algorithms for deep learning

algorithms use FFN to train the data. The Standard deviation comparisons also confirm LSTM to provide the least deviated predicted values.

4.4 Discussions and Conclusion

This research implements a method for an IoT-based framework to collect environmental indoor room parameter-based multivariate time-series data to be uploaded on the central gateway database using MQTT protocol. The collected data was analyzed with deep learning algorithms for the presence of human beings to be rescued from the building. This is to inform the fire brigade of the location of the presence of human beings in a building. Comparative analysis of deep learning algorithms was also conducted with error measurement techniques.

The reason for choosing MQTT over HTTP communication protocol is to allow real-time data ingestion into the database as HTTP protocol considered as client/server presents a communication overhead. The system was configured in the publish-subscribe paradigm. The sent parameters are considered as topics and the continuous sent data are considered as messages, in another word it is topic/message policy. The configuration of the mist node is an MQTT-client publisher to the MQTT-broker with the role of publishing data to the broker. The database is configured as an MQTT-client subscriber to be precise.

LSTM shows to over-perform the widely used machine learning algorithms such as Naïve Bayes and Support Vector Machines model. The implemented LSTM model is compared with another deep learning algorithm such as Multilayer Perceptron FNN (Feedforward Neural Network) that confirms a reasonable result compared to the other machine learning techniques as shown in Figure 4.11. In conclusion, the proposed LSTM model using captured real-time collected data is compared for performance metrics with other algorithms. The LSTM model's accuracy is 96.8% on the tested multivariate captured data that is nearly 16% better than the best of the others. The LSTM model does not over-fit from the used data and minimizes the loss that can infer their prediction.

The various limitations in this research use different parameters such as temperature, humidity, light, humidity ratio, and occupancy as data to predict the presence of an occupant within the room. A lot of observations have been explored as limitations of variables or biases to affect the prediction results. CO_2, pressure, and humidity parameters were reported to be noisy when the

experimental setup is not conducted in a controlled environment hence, it brought unconvincing prediction results.

Research shows that the light parameter can be noisy too due to daylight differentiation. While other researchers used dust sensors, noise sensors, and cameras to precisely predict dust, noise can be interrupted by other external and internal factors like dragging chairs and so on. Data captured by the camera shows that they can expose privacy and thus are currently eliminated from the discussion.

The experiment took advantage of a mixture of different parameters as multivariate data in the prediction models instead of univariate data for each parameter among temperature, humidity, light, humidity ratio, and occupancy to enhance the experimental accuracy of the models.

In future research, the consideration of collecting data from additional similar enclosed environments using deployed IoT sensor kits at various places across cities of Rwanda will be taken care of. Further, FIWARE, an open-source platform for smart environments will be explored. Moreover, the temporal dependence of data on the occupancy level should be examined. Besides, the impact of external factors (weather conditions, noise, and airflow) on prediction accuracy should be considered. The extension of both univariant and multivariant data with noise-canceling/avoiding techniques to precisely measure real-time prediction models with even better accuracy is also planned. Furthermore, future work will explore how the deeper layers of LSTM could affect the performance of the model concerning the tested multivariate time series data set. Finally, further evaluations with other deep-learning models will be performed to investigate the gates that have the most influential role.

Chapter 5

CONTAINERIZED ARCHITECTURE PERFORMANCE ANALYSIS FOR IOT FRAMEWORK BASED ON FIRE PREVENTION

Nowadays building infrastructures are pushed to become smarter for the future due to the environmental comforts of living. Enhanced safety, energy-saving, and comfort upgrades are taking advantage of the new technologies that are evolving. Normally most buildings are configured to respond to the safety concerns of the occupants. The advanced IoT techniques in a combination of Edge computing with lightweight virtualization technology is being used to improve users' comfort in their home. It improves resource management, and service isolation without affecting heterogeneous hardware deployed. In this chapter, the containerized architecture framework for supporting multiple concurrent deployed IoT applications for the smart building is proposed. The developed prototype uses advanced sensor networks as well as containerized microservices, centrally featuring the DevOps paradigm. The proposed prototype is tested in different academic buildings for data acquisition for three months. Different deployment architectures are tested to ensure the best cases based on efficiency and resource utilization. The results shown that with DevOps paradigm, the increases of sensors node requests shall not affect the server resources.

5.1 Chapter's Article Details with Authors Contribution

Hitimana, E.; Bajpai, G.; Musabe, R.; Sibomana, L.; Kayalvizhi, J. "Containerized Architecture Performance Analysis for IoT Framework Based on Enhanced Fire Prevention Case Study: Rwanda". Sensors 2022, 22(17), 6462 https://doi.org/10.3390/s22176462.

Personal Contribution: The idea of writing this feasibility study arose out of a meeting with Gaurav Bajpai, Louis S., and M. Richard. Formal analysis was done by B. Gaurav, I did methodology together with B. Gaurav; I implemented the algorithms for data analytics under supervision of Louis S. M. Richard brought the idea of sensor calibrations with B. Gaurav. I wrote the manuscript under guide with B. Gaurav. B. Gaurav, S. Louis and M. Richard performed paper

review. I produced all the figures and tables. All authors read and agreed to the published version of the manuscript. All together worked on review feedbacks and agreed for submission.

5.2 Introduction

The software development challenges arise due to the variety of IoT device-as-a-services, IoT infrastructure-as-a-service, and edge infrastructure-as-a-service [171]. In IoT, smart sensing and actuating objects are coupled with information communication technology. They are embedded in digital environments to produce a huge volume of data. The data needs to be extracted, processed, and analyzed efficiently to arrive at some meaningful insights to support IoT applications.

The implementations are done using different platforms as services. The applicability of these platforms was challenged by their efficiency, performance, and variating scalabilities. This happens in case there is a system escalation, along with data analytics resource consumption. The stated issues mentioned above are characteristics of monolithic applications [172]. A monolithic application is defined as a single-tiered software application that integrates both users' interfaces along with data access code as a single program of the same platform.

IoT systems provide important services, sometimes critical ones requiring an uninterrupted operation or high levels of availability. Although their complexity makes them difficult to integrate along with monitoring in-depth to prevent anomalies as well as failures [173]. DevOps methodology has shown its power by allowing system availability, scalability, and anomaly handling while the system is running [174].

DevOps is well established as an approach to ensure a rapid and efficient value-added delivery of the software product to the market. It happens through a tight collaboration between the developer (Dev) with the team that operates the software system (Ops). The idea is to decrease the gap between product design with its operation through a continuous automation process. This is supported by different tools at various stages of the product life cycle [175 - 177].

DevOps and Agile are two pillars efficient in achieving business strategy. It also checks the overlap of the traditional operational as well as a development team to create an environment that is continually improving operations. It is achieved through a cross-functional team of developers and operators [178].

The continuously rapid integration of new devices with feedback acquisition to instrument maintenance and updates leads to job accumulation. To meet customer satisfaction, it requires a

multidisciplinary collaboration between development and operations teams. The collaborative team has to test the platform in production, build new releases, as well as deploy them [179]. DevOps [180] is a solution to facilitate that collaboration [181] by accelerating along with improving the cycles of maintenance [182].

The motivation of this chapter was to contribute to the improvement of building safety prevention (specifically, from fire outbreaks) using IoT technologies. The main question was how to integrate various IoT sensor nodes with scalable characteristics to provide a sound IoT system that provided real-time communication to the stakeholders (fire brigades) in case of a fire outbreak. The idea arose due to various challenges, such as IoT applications' scalability, updatability, integration, and maintainability that affect performance and efficiency.

This chapter highlight the experiment conducted to design and develop an IoT containerized microservices architecture using docker technology. The architecture proposed the measurement of performance, efficiency, as well as escalation of the IoT framework. It allows the docker swarm to manage all available and running services based on the available resources. The IoT framework will be trimmed into different services such as device management services, data management services, user authentication management services, and application services.

5.3 Related Works

The three-layer classic IoT reference architecture was extended by applying context-aware, serverless, microservices-based cloud-centric techniques for IoT and people applications [183]. Here, the focus was on frameworks that offered a high separation of concern degrees by splitting the application layer into different sub layers based on their responsibilities. It also traced atomic components to serverless microservices to facilitate the design, development, and deployment of applications in the healthcare domain.

5.3.1 Usage of DevOps Methodology

The three-tiered DevOps model was used to build an information system capable of providing real-time analytics [184]. Management within a smart city project was proposed to support the heterogeneous environment. However, the data visualization was conducted on an existing open-

source platform, namely KIBANA[1] from Elastic-search. The integration of DevOps into the design of IoT frameworks presented an ongoing effort toward provision, with validation of solution security and privacy, along with the quality of complex software systems [175]. The solutions were built on top of an established ENACT[2] IoT DevOps concept and framework.

The development of a service model structure consisted of several functions within its services. The defined hardware compatibilities tested an automation system in a building for emergency evacuation services [185]. However, the concerns of continuous integration using DevOps techniques were not considered. The development of service contracts for IoT microservices from DevOps perspectives was considered to address the diversity of service contracts using common languages for IoT data and programming [186].

5.3.2 Application of Microservices

Rapidly developing technologies enable ubiquitous connectivity. This has resulted in the continuous growth of connected devices. On the other hand, it has also revealed challenges related to the development and architecture of information systems. The wide variety of proposed solutions enabled by interacting with devices makes it practically impossible to accurately manage the predicted ratio of data traffic patterns.

In the management of a growing amount of delivered services [187], it is essential to implement scalability mechanisms. Specific strategies were proposed herein to maintain system stability and scalability by ensuring the basic scaling mechanism (such as vertical and horizontal). Vertical scaling requires the addition of resources to a single processing node, resulting in more work handling with additional capacities. It is, however, limited by cost-effectiveness, physical constraints, and the availability of specialized hardware. Horizontal scaling requires a set of design practices and planned activities, leading to an inherent distribution within the system [188]. The horizontal scaling approach defines the unique ability as the keystone for large-scale architecture design.

---

[1] https://en.wikipedia.org/wiki/Kibana

[2] https://www.cyberwatching.eu/projects/1048/enact/news-events/enact-enabling-devops-realm-trustworthy-smart-iot-systems

The initial concepts for software developments used monolithic architecture [189]. The services were developed on a unique repository shared among multiple developers. In these concepts, developer teams had to guarantee that all other services would continue working as they did before. Another disadvantage was the presence of new updates, ready for deployment in the production environment. Updates require that all services be restarted, causing severe issues, from the users' perspective. One final challenge is that each time one part/function fails, the entire set of services gets compromised.

Due to the wide variety of scenarios created in IoT and their different challenges (such as the heterogeneity of IoT devices, communication, and platforms supporting large numbers of connected devices), the monolithic pattern is not recommended [190].

The microservices-based architecture proposed the distribution of the applications in a set of services in such a way that each was independent of the others. These new patterns have been used to solve specified issues with new features, such as scalability and reusability [191]. The use of an IoT agnostic architecture, highlighting the role of the IoT platform using a system with a broader ecosystem of interconnected tools, aiming at increasing scalability, stability, interoperability, and reusability. Herein, the solution used microservices architecture and serverless computing applied to smart solutions.

Microservices, at a high level, can be considered a black box that has been divided into layers. Depending on the tasks to be performed, the layers can be reorganized. Microservices communication is possible through interfaces exposed by the Microservices themselves. Hence, each microservices offers an application programming interface (API) to facilitate different types of connections.

Microservices are useful for the development of IoT-based platforms. They allow services-oriented development, which is growing in popularity due to growing interest in parallel computation. The microservices architecture, designed and built using the Jolie programming language, was developed to work directly with a service-oriented paradigm [192]. Here, the prototype platform for supporting multiple concurrent applications for smart buildings was tested using an advanced sensor network, as was a distributed microservices architecture [191].

5.3.3 Container Management

Most of the capabilities of containers are first executed on constrained devices with limited and relatively inexpensive storage and computing resources, like Raspberry Pi [193]. They aim to provide the capability of transversal interoperability among different networks, integrating devices with limited capacity. Container-based services provisioning has demonstrated diverse benefits that allow applications to run on a diverse set of devices, with heterogeneity in terms of hardware, software, and network. To improve the performance of the containers in the described constrained devices within an environment, better solutions are needed to simplify and improve their management.

At this layer of the architecture, it is necessary to ascertain the performance of container schedulers or brokers, along with allocation among resources. Multiple containers, running in a cluster, support management activities. A container scheduler in a cluster has multiple goals, such as ensuring the use of resources efficiently; user access is restricted based on work and location. Additionally, it ensures that applications are quickly scheduled to avoid massive waiting times. It provides a high degree of balance between resources. It also provides an error-handling mechanism that ensures the most convenient allocation, both timewise and power-wise [194]. The existence of diverse container schedulers for clusters supports the notion that there is no single solution to all problem paradigms. Among these schedulers, the following can be highlighted: Apollo Microsoft, Aurora Twitter, Borg Google, Fuxi Alibaba, Kubernetes Google, Omega Google, Swarm Docker, and YARN Apache, to name a few [195].

There are two key techniques used by containers to provide isolation of each container during execution environments. Those techniques are namespace and cGroup [196]. Namespace is used for isolation between execution environments, including process trees, networks, user IDs, and file systems of containers. cGroup allows for the management of resources, such as CPU, memory, network, and file I/O for containers. In the case of cGroup, resources can be controlled by assigning specific bounds to containers. It has also been shown that a specific number of resources for a container can be allocated to prevent resource contention between containers running concurrently.

The implementation and administration of diverse container-based applications are rapidly developing nowadays. This development could facilitate physical or virtual host distribution, built on top of central components, such as containers' schedulers in general, and Dockers containers' schedulers.

5.3.4 Docker Swarm

Among container schedulers, this research used docker swarm to address the problem of container management. Docker swarm is a container management solution under the development of docker organization providing a standard docker API. Its framework consists of two main components; (1) a node that acts in the role of administrator executes an image of swarm responsible for allocation of the containers on the remote machines (called agents, or nodes); (2) the nodes with a remote docker API capability are available to the administrator after the verification of its port correctness if it is open by the time the docker daemon starts [197].

Docker swarm has different major scheduling strategies that ensure the possibility of the scheduler selecting a node for executing a container. It includes strategy name or direct node selection, and selection of the node with the lower set of containers in execution spread. Regardless of the load of each container, or binpack, the selection of the most packaged node (lower available CPU/RAM) and the random selection of the node are also included. In the latter case, if the previous strategies select several nodes according to the previous criteria, the scheduler selects a target node among them randomly.

The resource allocation strategy must be determined at the time of setup by the administrator, or the spread strategy will be used by default. However, the spread method is the most extended scheduling strategy in most applications on the docker swarm mode cluster.

In this research, apart from microservices concepts which were used together, the architecture had additional containerization mechanisms built on top to ease scalability, flexibility, and transparency. This mechanism was achieved using visualization in a container-based approach, through docker. The proposed concept allowed for the robust development of a framework operating in a distributed way, with greater speed and increased independence over underlying operating systems.

5.4 Materials and Methods

This section details the materials and methods used in the implementation of the proposed architecture. The materials were classified into hardware and software components, data formats, and performance metrics.

5.4.1 Underlying Hardware Components

This subsection details different hardware components used during the implementation of the proposed framework architecture. The choice of criteria for each component depended on its availability and openness.

5.4.1.1 Embedded Board

ARM embedded architecture uses lower-power characteristics [198]. The single board computer (SBC) family raspberry Pi (RPi) takes advantage of these characteristics. RPis are small SBCs with 4 advanced RISC machine (ARM) cortex-A53 1.2 GHz CPU and 1 GB RAM, with Wi-Fi integrated [199].

In this research, easily obtainable off-the-shelf hardware was used to facilitate reproducible results. The setup was composed of three RPi Alarm ARMv7 for Broadcom [200].
Table 5.1 describes the configurations of the experimental testbed. The desktop simulated the fog node while the RPis acted as the edge node and gateway.

Table 5. 1: Configuration of the experimental testbed.

Device	Desktop	Raspberry Pi 3 B+
CPU	4.5 GHz Intel Core TM (4 cores)	Quad core Cortex-A72 (ARM v8), 1.5 GHz
Memory	4 GB DDR3	1 GB LODDR4 RAM
Storage	500 GB HDD	32 GB (microSD card)
OS	Linux x86_64	Raspbian GNU/Linux (Jessie)
Installed Software	Docker Swarm, NGINX, PostgreSQL, Redis, cAdivisor, Prometheus, Grafana	Docker, python

5.4.1.2 Edge Sensor Nodes

In fire detection, prevention, and occupant detection use cases, this experiment used a couple of sensors to gather specific parameters. In a specific room, sensors were used to detect the presence of fire temperature, humidity, CO_2, and smoke. Motion sensors were used in conjunction with obstacle sensors to detect the presence of human beings. Table 5.2 details the sensors used, and their operating ranges based on the limits of the value for sensitivity and distance they cover.

Table 5. 2: Sensor's operation ranges.

Name	Description	Min Value	Max_Value	Min_Distance	Max Distance
DHT11	Temperature	0 ± 2 ℃	50 ± 2 ℃	1 m	10 m
	Humidity	$20\% \pm 5\%$	$80\% \pm 5\%$		
PIR	Motion sensor	0 V [0]	3 V [1]	1 m	7 m
IR	Obstacle sensor	0 V [0]	5 V [1]	1 m	5 m

MQ-2	CO_2 and Smoke sensor	300 ppm	10,000 ppm	2 cm	4 m

5.4.1.3 Network Connectivity and Protocols

The main tasks of a node were split into sensing/actuation, computation, storage, and communication, among the other nodes [201]. The node architecture was a bundle of multiple sensors which gathered environmental parameters to be sent to the cloud. The controller was the part of the node that fetched data from the sensor, exchanged them with the communication device, and controlled the behavior of the actuators. The microcontroller added the ability to interact with it by programming it. It had memory, enabling it to control the behavior of the node [202].

The implementation used the RPi board, which allowed LAN internet connectivity. The communication used two protocols: MQTT and HTTP. MQTT operated through data messaging architecture that used the publish/subscribe paradigm to ensure that the data was sent smoothly if the sensor node was subscribed to the broker and the database. At the end-user data access, the web services used HTTP so that the user could access the information through a browser application.

5.4.2 Underlying Software Components

This section details the software components used in the implementation of the proposed framework, including the docker engine as services containers manager, server setup for web and database management, data formats, protocols for transmission, services patterns, and performance monitoring tools.

5.4.2.1 Dockers

Containerization is a technique used to abstract applications from the environments that they execute. Docker containers wrap up the software and its dependencies into a standardized unit for software development that includes everything it needs to run [203]. Containers are a lightweight alternative to virtual machines (VMs) for multi-tenancy within a single host. Here, computing devices were modeled as containers and managed using the docker automation framework. This was accomplished in two parts: firstly, for resource allocation, and secondly, for software configuration.

The key advantages of using docker for distributed applications included lower CPU overhead, version control, portability, and network performance improvement [204]. Docker developed a distributed service platform by considering the fault tolerance of services [205, 206].

Dockers-based gateways and architecture have flexibility with scalability through the microservices paradigm [192, 207].

This research took advantage of docker's features. The idea was to combine docker container services and microservices strategies for modular, scalable edge computing frameworks for IoT smart buildings.

5.4.2.2 Data Management and Data Format

All sensor-generated data were periodically saved in the database, with data management conducted on the web server. The database module was implemented using PostgreSQL over the web server used as Nginx. Postgres was chosen as an object-relational database management system (RDBMS). It emphasized extensibility and standards compliance and could provide replication of the database for scalability and security.

Nginx is an open-source reverse proxy server for HTTPS, with a load balancer. Nginx was chosen as the web server for its security, reliability, and load balancing when there was huge traffic with connected users. Its characteristics focused on high concurrency, high performance, and low memory usage. The data exchange from sensor nodes to the cloud was formatted to allow efficient data transmission with low bandwidth. The data was converted into JSON (JavaScript Object Notation). JSON has been adopted as a standard text-based format for representing structured data.

5.4.2.3 Application Services: Microservices

The proposed architecture in this research targeted the microservices architecture applied to the IoT framework. The architecture is made of different components including all real-time data acquisition and processing architectures in the IoT domain. It had three key components: (1) data sources, (2) data processing, and (3) end-user data access.

5.4.2.4 Monitoring Parameters

The performance of edge computing applications followed different runtime variations depending on the running conditions. Conditions could have included the number of arrival requests and the availability of virtual resources. Additionally, the network connection quality between different interoperating application components distributed over the communication network was an additional key to consider.

A mechanism to allow the resources to be intelligently provisioned with little or no effort by an application provider was required. The container manager was configured to ensure that the orchestration process of provisioning resources was done. The resources considered for provisioning in edge/cloud computing were the usage of CPU, memory, storage, and network.

5.5 Proposed Architecture

The proposed framework architecture consists of a set of services distributed across different computing sites. Each application is made of separate independent deployable components/containers with all possible functionalities hiding its implementation details. The provision of those separate modular services provides the characteristics of the services to be small, and loosely coupled. They improve both testability and management using containerization packages configured in each end device node. Those configurations provide minimum environmental capabilities for the application to run properly.

The proposed solutions of microservices for smart buildings in general and fire detection with occupancy detection are detailed in Figure 5.1.

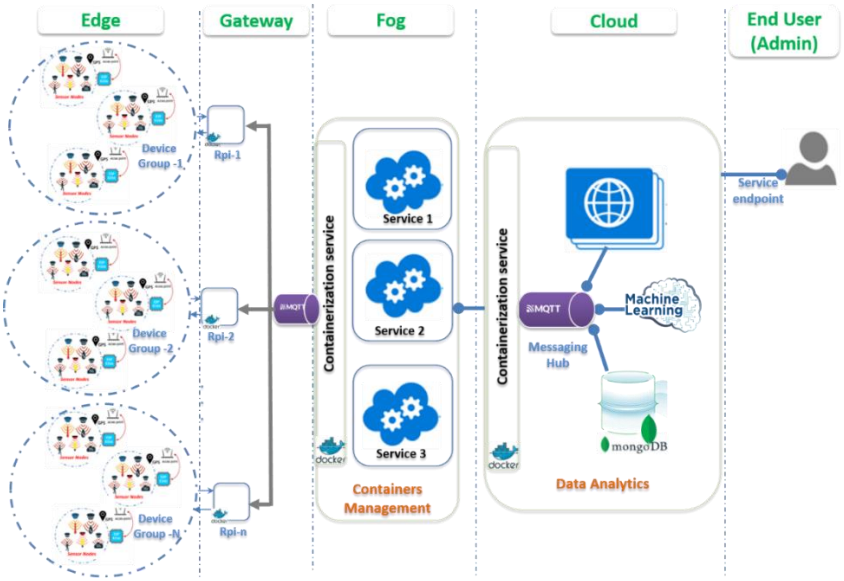


Figure 5. 1: Proposed deployment architecture.

The concept considered allowed independent mounting and unmounting of the new nodes. Every node was considered one service for data acquisition and transmission. It allowed any connected node to be updated and integrated with the management of the services manager played by docker swarm. The transmission protocol, data management, web services, and visualization

services were managed as microservices. This allowed services reusability without affecting the whole structure.

Different layers of the proposed architecture are shown in Figure 5.2, describing each layer with its computational resources.

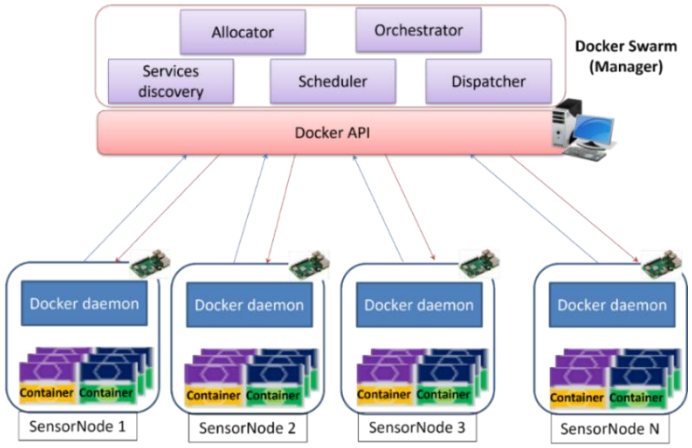


Figure 5. 2: Architecture with containerization principles.

The device layer known as the edge layer holds all capabilities of sensing and communicating with their surrounding environment. The sensor nodes are grouped to allow their connection and disconnection management. More details are explained below.

5.5.1 Edge Layer

The sensing layer describes different sensor nodes deployed in the testing environment. The sensor nodes include but are not limited to temperature, humidity, CO_2, proximity sensor, and IR sensor. They are all deployed to a powered raspberry Pi board connected to a LAN network. Each Pi board is configured with a docker machine, and all sensor services are deployed on it in a container-based approach.

5.5.2 Gateway Layer

This layer is designed to manage the device groups deployed in different buildings to ensure the security and scalability of the framework. Once the new device is mounted, its corresponding gateway will update the container manager in the next layer (Fog layer) to update its discovery service, to allow other containers to be updated. The gateway layer ensures that all containers are

running efficiently and the decision that requires little computations are done at this level as this Pi board also has constrained resources.

5.5.3 Fog Layer

In a large-scale container deployment over numerous servers, tracking all the containers and managing their life cycles is a complex task. This layer is composed of a configured server with a machine learning Linux operating system (OS). On top of the OS, the docker swarm is installed. Its role is to ensure the proper administration of service lifecycles in massive deployments of docker containers. The layer allows data management as well. The security with redundancy paradigm, the database server as well as access are done at this layer to maintain the system. All deployed services in the docker environment are cooperating hands-in-hand with the docker hub to update new versions of container images used just in case.

5.5.4 Cloud Layer

The research is taking advantage of cloud services by allowing all the public management and access to the implemented microservices resources to be managed. Main message hub services are fully managed in the cloud to control data transmission through queuing discipline. The database servers are deployed to keep persistent data for further usage. Overall web services to allow data access and user permissions at different user services are managed at this layer. Finally, data analytics are performed in the cloud to use cloud infrastructures that are limited at the edge node.

5.6 Performance Analysis and Monitoring Packages.

There are four levels of monitoring (containers, virtual machines (VM), end-to-end link quality, and application) for self-adaptive applications to edge computing applications [208]. Due to the heterogeneity of the applications used in this implementation context, the need for performance measurement was critical. This research was implemented for container-based performance monitoring purposes. It assisted the process of pulling and migrating container images across computing nodes in fog or cloud in contrast to virtual machines [209].

Docker itself had a built-in mechanism to access and convert container metrics into a statistical format using docker stats methods. Those metrics included runtime metrics and resource usage for a given container. This research proposed to use external built-in components to retrieve detailed sets of metrics by accessing exposed remote API using GET methods.

In monitoring the system behaviors, the following packages were used:

- **Container advisor**, also known as cAdvisor, is a provided open-source platform. It was used to track, measure, aggregate, process, and display performance monitoring data for the running containers.
- **Prometheus** is a monitoring tool relying on an open-source scheme. On top of monitoring activity, Prometheus can also store persistence time-series-based data for further processing and retrieval using PromQL language.
- **Grafana** is a web-based open-source user interface, used to visualize large-scale performance monitoring information. It can be configured to work with the Prometheus database to visualize the time series data of the metrics.

5.7 Discussion of the Case Study - Rwanda

Rwanda is an eastern African country with a fast-growing economy and its cities gradually expanding with city dwellers 881,445 in the largest cities and 8,169,341 in rural areas [111]. This demographic information leads to the fire risks increase. Due to the constructed public houses, with little firefighting equipment or with few people with experience in using them there is a high risk of fire outbreaks. In the last 10 years, the number of fire outbreaks have increased with 327 fires registered across the country since 2010 shown in Figure 5.3. The public buildings include shops and high learning institutions holding a huge cloud of people. The case study taken is from the University of Rwanda, college of science and technology, KALISIMBI Block where the research is conducted and deployed.

A national inquiry set up to investigate the causes of the fire outbreaks found that 61% of the fire incidents were due to short circuits, while 22% were due to unknown causes, 9% were arson, 4% caused by road incidents, 3% by hazardous domestic activities, and 1% by the chemical explosion as shown in Figure 5.3 [113].

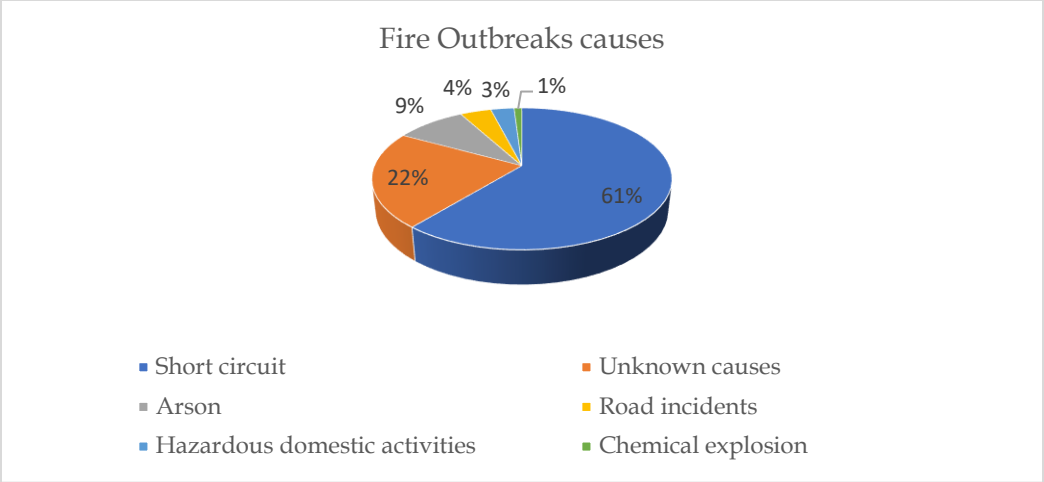


Figure 5. 3: Fire outbreak causes.

Due to the above-mentioned fire outbreaks cause, there is a need to address the issue by incorporating an IoT solution. Some applications of IoT focus on the smart healthcare domain [210], however, this research proposes the IoT frameworks applied in the smart building with the following capabilities: reusability, scalability, accessibility, interoperability, and updating.

Referring to the sensor node deployment shown in Figure 5.4, temperature, humidity, and CO_2 sensors were connected in the middle of the room to maximize capture. A proximity sensor was deployed at the entry point of the room. The reason for choosing this configuration was to be sure that any captured positive value would show the moving being. The last parameter of interest was the counting of entries using two communicating infrared sensors (IR). The presence of a moving object in the office would not necessarily mean that it was a human being. All those sensors in combination would help to improve the questions such as to recognize the presence of humans if there were a moving chair during an earthquake.

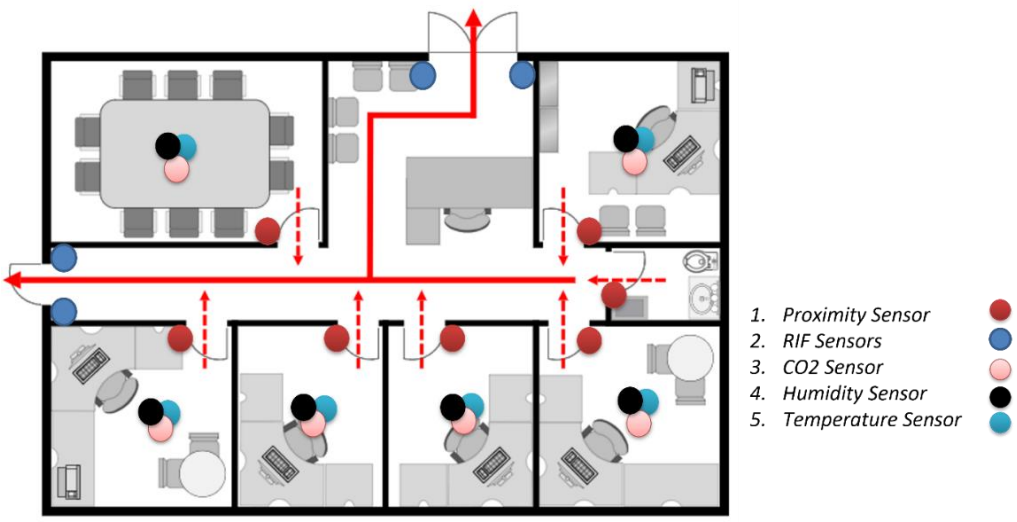


Figure 5. 4: Sensor deployment architecture in UR, Kalisimbi Block offices.

The management of the architecture used a distributed set of identical IoT devices deployed in different locations. Each device was configured to be part of the same group.

Every time a new device was turned on or connected, there were different steps to follow:

- A new IoT device, such as a sensor or sensor node, is connected to the manager.
- The connected device/sensor is identified with its unique identification.
- The manager assigns the device to the device group, meaning the floor coverage of the building.
- The latest device group configurations are pulled.
- The manager checks for any updates from the broker (manager) map table to update the connected groups.

Any time the admin deployed a new device or changed existing device configurations; the following steps were to be followed:

- Push a new container version, with its corresponding tags, into the configured docker registry.
- Update the control API.
- Update the backend database with the new configuration.
- Update all connected IoT devices in their respective groups.

For occupant presence counting, Algorithm 1 was used to synchronize two communicating IR sensors (one counting incoming users, the other one decrementing outgoing users). Its

deployment, shown in Figure 5.5, positioned it at any known entry points of the building. The reason for this was to facilitate the knowledge of the aggregated number of occupants in a specific building, in case of an incident.

The configuration and cooperation of two IR sensors are programmed by following algorithm:

Algorithm 1 An algorithm for entry counting

Require: $peopleIN$, $peopleOUT$, $IRinStatuts$, $IRoutStatuts$, $inPeople, outPeople, stayPeople$

Ensure: $Number\ of\ stayPeople\ in\ a\ Room$

```
peopleIN ← 0
peopleOUT ← 0
while (true) do
    Read IRinStatuts
    Read IRoutStatuts
    if IRinStatuts == 0 then
        RETURN inPeople ← peopleIN++
    end if
    if IRoutStatuts == 0 then
        RETURN outPeople ← peopleOUT++
    end if
    stayPeople = inPeople − outPeople
    if stayPeople <= 0 then
        No people in        ▷ No send process as there is null data to be send
    else
        RETURN stayPeople
        send stayPeople in the database        ▷ Send data to the database
    end if
end while
```

5.7.1 Deployment Architecture of the Internet of Things Sensor Nodes

Figure 5.5 depicts the deployment design architecture of the proposed building monitoring IoT system. The system components are shown concerning the Edge/Gateway/Fog/Cloud architecture.

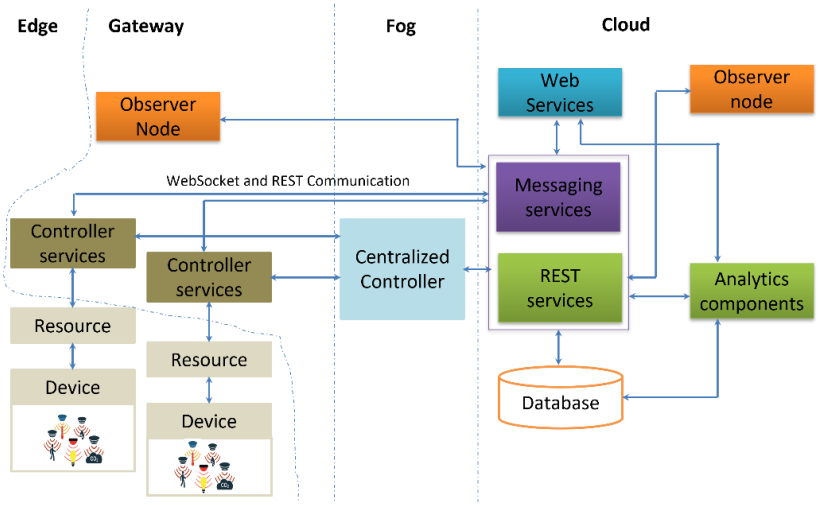


Figure 5. 5: Deployment design of building monitoring IoT system.

The system consists of multiple end nodes deployed in different rooms location to monitor selected environmental parameters including temperature, humidity, CO_2, motion, and occupants along with respective sensor embedded devices. The end nodes are connected to the manager (centralized controller) to ensure connectivity with deactivation in case it is not functioning well.

The controller was configured as a Linux computer with all management capabilities. An orchestrator was installed there to manage all microservices, including the services to connect, update, and disconnect any newly connected sensor node. The evaluation of running, pending swarm of services, was monitored through the configured services docker swarm visualizer. The end nodes sent the data to the cloud database using publish/subscribe services. However, the management of sensor nodes' gateways was managed by the manager through REST services. The data analytics were done in the cloud to aggregate the data and make predictions.

5.7.2 Representation of the Internet of Things Domain Model

This section details the architecture of the proposed connected modules from the device end to the human services visualization.

Figure 5.6 represents the identification of the proposed IoT scenario. The overall figure shows that the user needed to interact with a physical entity in the physical world. To allow/control the interaction of humans with the rest of the system, there was an entity called an active artifact that was running software. In the context of this research, an active digital artifact role was played by the Docker manager to harmonize the sensor nodes' operation with their respective containers. A physical entity represents a discrete identifiable entity in the physical environment: building, rooms, as well as any actuation. A virtual entity represents the physical entity in the digital world; in other words, it could be represented as the logical interface to allow physical object interaction.

The device provided a medium for interaction between physical entities and virtual entities. It could be attached to the physical entities or placed closer to the physical entities. Resources were software components, either on-device or network resources. Services provided an interface for integrating with the physical entity. It accessed the resources hosted on the device (or the network resources) to obtain information about the physical entity or perform the actuation process upon it.

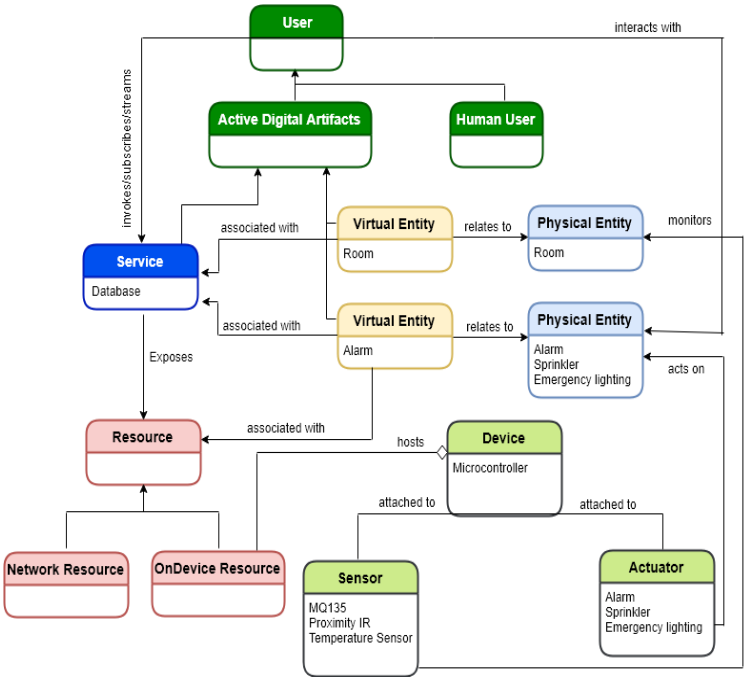


Figure 5. 6: Proposed UML representation of the IoT domain model.

Figure 5.7 shows the process of mounting a new sensor node to the existing system. The configuration of a new sensor node consisted of only a few steps. The first step was the process of mounting the new device with its respective driver. The second was to allow the preparation of device updates on the list of available services. The third was to be able to pull the update from the container manager. The last step was to be able to interact by pushing data to the respective entities.

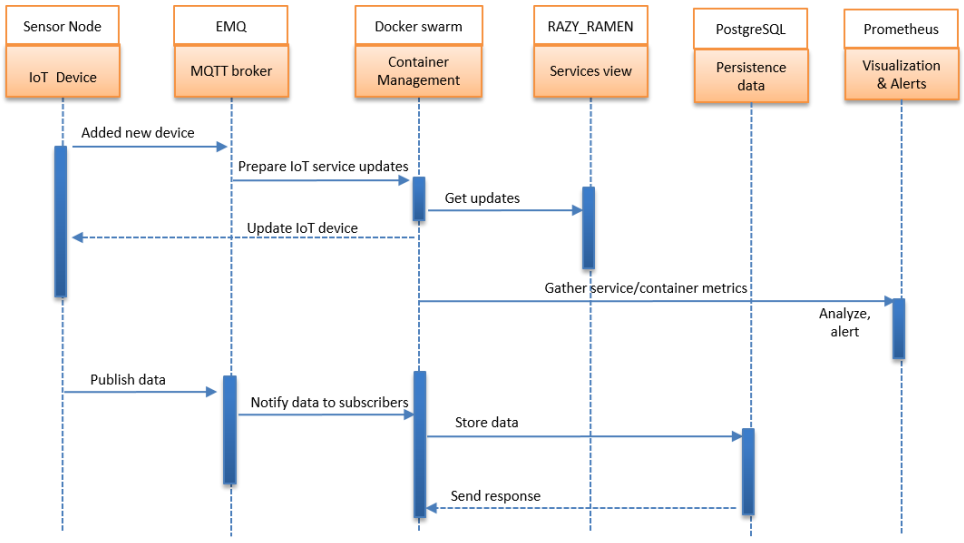


Figure 5. 7: Sequence diagram activity at the sensor node perspective.

Due to the publish and subscribe services offered by the MQTT protocol for real-time information sharing between connected devices with the central controllers, the sensor node was able to connect with predefined rules to meet the required configuration. The broker deployed in fog edge with docker swarm was able to work in cooperation to ensure that there was no interference. The docker swarm could also make sure that every device with its resources was containerized. Every container was assigned its unique identification to allow the restriction of available resources.

Figure 5.8 shows the system views from the end-user perspective. The user interacted with the system by accessing its services through the web application. In the monitoring process, the end-users, such as the fire brigade, called emergency response controller (ERC) in this context, must be authenticated by the system.

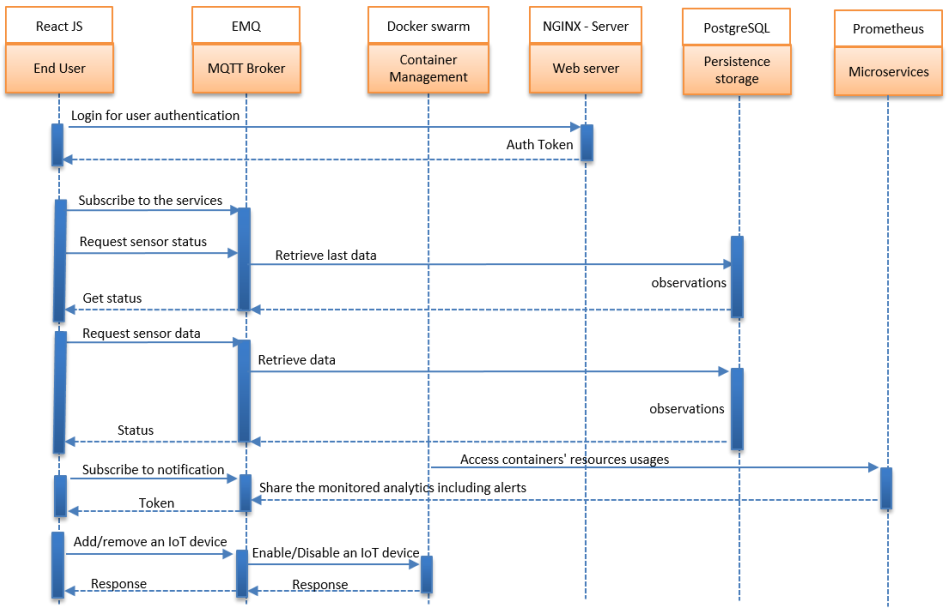


Figure 5. 8: Sequence diagram for end-user perspective.

For the client application (services) to be able to access the data generated by sensors, the services must be subscribed to the broker to allow smooth message queueing and notification sharing using the MQTT protocol. The broker used in the research implementation was EMQ. In the data management process, all generated data from the sensor nodes were saved to another service database subscriber known as the PostgreSQL server. Data could be retrieved from the database server in a different format based on the user context. In this situation, as there was a

need to collect data for research purposes, there was no recommended user interface serving this context. The data were kept for further use and retrieved in JSON format for preprocessing.

For performance monitoring, a combination of different monitoring tools was used—cAdvisor, Prometheus and Grafana. Figure 4.9 shows only Prometheus to simplify the view. All services used in this research (subscription/publish, web application, web server, database server, monitoring servers) were built as microservices under the docker container. This configuration eased dependencies on management and ensured the correct configuration. The services, known as microservices, were wrapped into an efficient orchestration module known as docker swarm. This facilitated the process of activation and deactivation any time there were any malfunctioning services, including device services.

Configuration of the services used in this research was done in layering structures. Prometheus was configured to ensure all the metrics from the cAdvisor were ingested and grabbed. To confirm the time interval of sharing the data, the scrape interval was set, including the endpoint where the cAdvisor was executing. It was forced to expose the data at port 8080. All defined services were wrapped within docker containers to embed all required dependencies. The configurations in the docker-compose file specified which containers were part of the installation and which ports were exposed by each container service. The configuration file focused on data ingestion from cAdvisor, Prometheus, and Grafana for simplicity.

One key point to mention regarding the configuration: by configuring the specific container image, the dependency was able to ensure the direction the data took from container service one to container service two, and so on. There was also another configured time-series database (TSDB) to capture metrics parameters in real-time. In this research context, the configured storage was known as the Redis database.

5.8 Results

This section describes the result outcomes of the implementation. It details the deployment place along with the setup of the testbed including the services usage visualization.

5.8.1 Deployment Location Selection

This research is detailing its application in the fire brigade domain. The site has been selected referring to many fire outbreak cases remarked in the student kitchen in 2018. KALISIMBI Block

site is selected as it is where all attention can be cared about since all laboratory types of equipment for chemistry, physics, electronics, and computer labs. Any time, chemical components can bust which can affect the student's life. The deployment for testing is configured on the third floor, in one selected room to help prevent and forecast incidences.

5.8.2 Hardware Set up

The real implementation of the proposed IoT architecture system uses detailed hardware components. The testbed is configured with all selected sensors based on the case study. They are all mounted in a raspberry Pi microcontroller board running docker services. The reason for deploying all sensor nodes services as microservices in the docker environment was to allow easy scalability with lightweight performance management capability.

The system is powered by a power source and connected to the LAN network of the campus to ensure that it can sync services with docker hub. All data were sent by subscribed sensor nodes through the pipelined microservices up to the hosted database in the UR campus data center. Each RPi sensor was categorized into groups to manage its resources and connection capabilities. The computer was used in the testbed where the docker manager was configured to facilitate microservices management and scalability, as referred to in Figures 5.9 and 5.10.

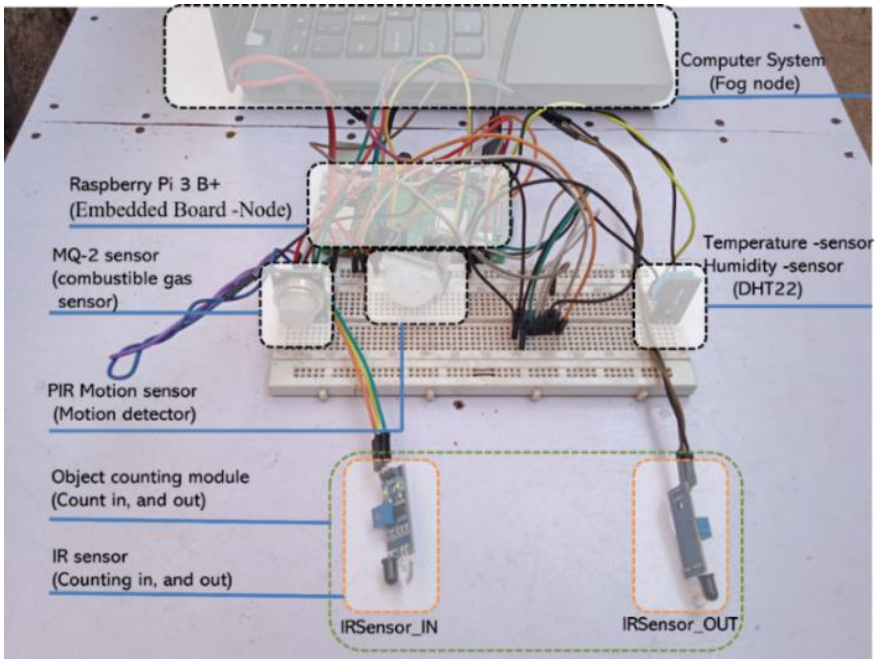


Figure 5. 9: Detailed mounted sensor node with all related components.

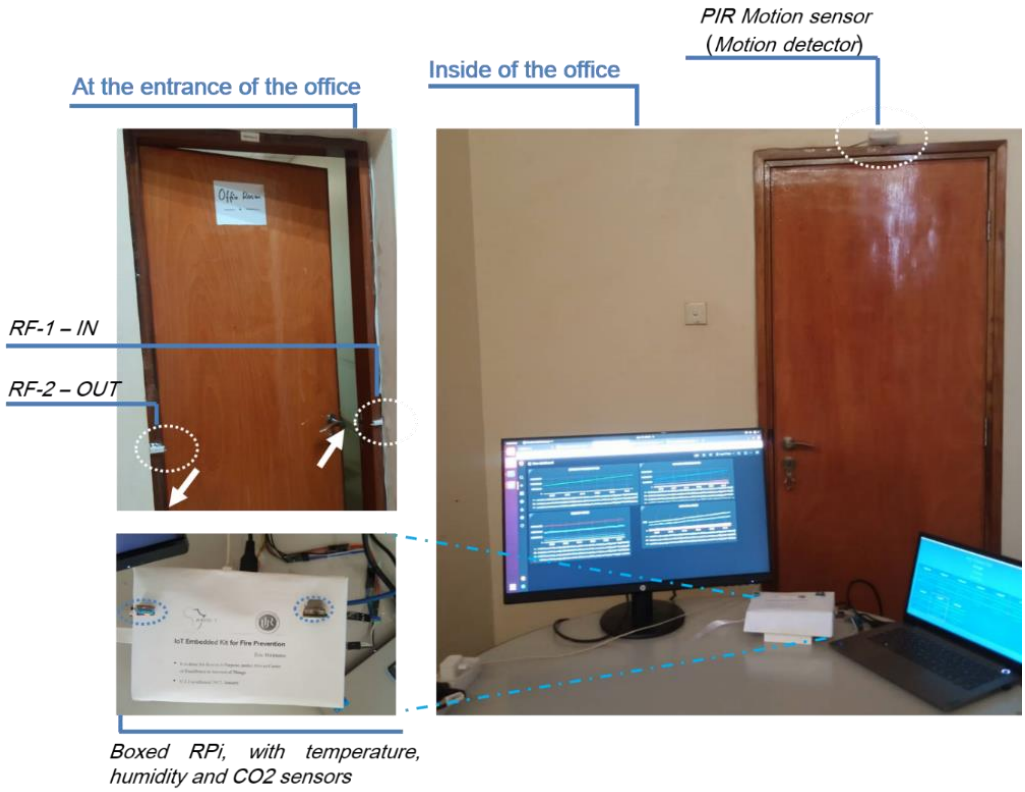


Figure 5. 10: Real-mounted sensor nodes with all related components in the building.

5.8.3 User Application

The user was able to access the platform services through the developed web application under JavaScript, HTML5, Cascading Style Sheets (CSS), and ReactJS technologies. More clearly, the ReactJS framework was used to develop the web application following the responsive paradigm. For monitoring the statuses of the containers through the container manager, docker swarm visualization tools were used, as shown in Figure 5.11.

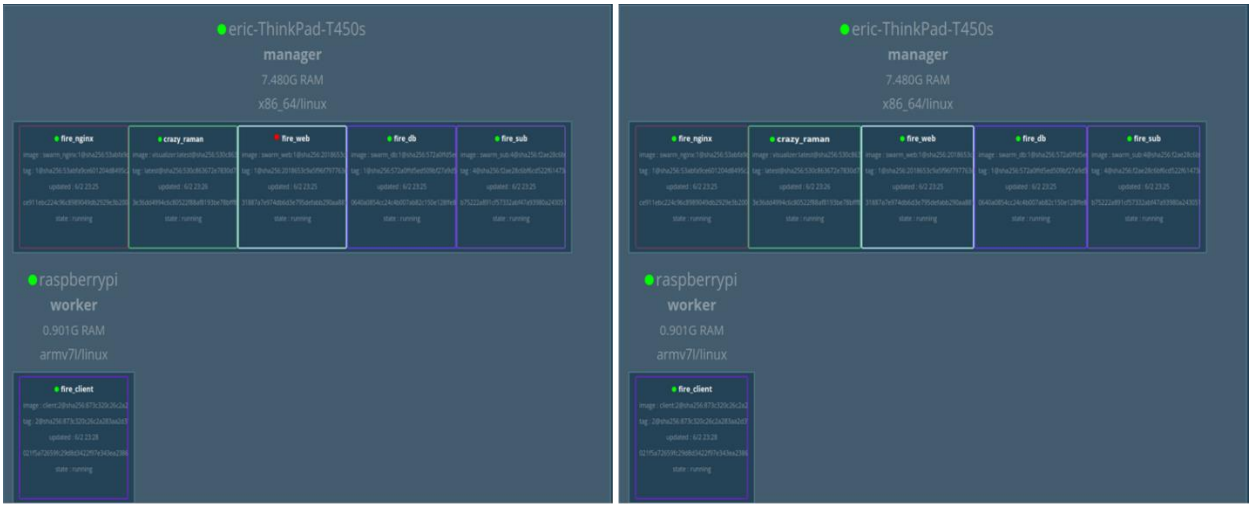


Figure 5. 11: (a) Visualization of container clusters with one non-working container service with the attached RPi worker. (b) Visualization after system container stabilization.

In a test experiment, the initial condition of the visualizer service was executed by the computer system. The docker swarm was configured on the board first. However, due to the limited capability of the storage, the laptop was used as a docker swarm manager. The RPi board was configured as the worker. It was given the IP address to be able to run independently.

In summary, the system consisted of one swarm manager (laptop), and one node worker for testing purposes. All the container services were running in the swarm manager, while the sensor node services were running on the worker side. The swarm manager ran all services including worker services. The reason for this configuration is that it would be able to update the newly connected worker and show the status of non-performing services, as shown in Figure 5.11(a). After the system is stabilized, Figure 5.11(b) details all available services in the performing state.

Figure 5.12 shows the behavior of the docker swarm manager during the process of orchestration at the scale-up process.

Figure 5. 12: Docker swarm with duplicated container services at the scale up purpose.

The other testing process was to measure the system availability level based on the scalability capability of the system by trying to scale up and scale down all the deployed container services running in the docker swarm manager. The performed operation was designed to test the capability of the system availability when it came to serving an instant number of incoming requests that may affect the system overload.

The scaling test was designed to observe how the system would behave in cases where there were hundreds of thousands of connected workers (sensor nodes) interacting with the services. The other concern was to see how the platform behaved when many end-users requested the services (building control units).

5.8.4 Performance Analysis: Reliability and Scalability

This section covers the performance evaluation of the proposed framework by following the approach and metrics. The evaluation strategy used is to make a function to generate a huge amount of data ingesting into the system to observe its behavior. The data generator function's purpose is to simulate IoT devices as publishers. The ingested data allows us to see how the containers will be orchestrated under the governance of the docker swarm manager to compete on the available resources.

The goal of this experiment was to assess the proposed IoT framework from two different points of view:

- **Reliability**: the reliability of the system was measured by utilizing all the available resources to ensure that the system was running at full capacity. Those resources included CPU and active memory usage, with how much data were processed in a specific period. CPU usage metrics indicated average OS core utilization for all available CPU physical and virtual cores of the running system. Active memory showed the number of bytes stored in memory during the experiment at the time of the test.
- **Scalability**: the measure of the prediction for extending the system to support hundreds of thousands of IoT devices publishing data with minimum system overhead to help time-critical applications. The measurement of these characteristics as indicated by the system throughput and response time metrics, showing how the system scalability was maintained.

The excellent performance was shown by the high throughput rate, as in principle, it should have been increased linearly as the number of devices or requests increased. The calculation of the throughput was given by the number of messages received and processed by the IoT system in one second, the lower the response time, the faster the response and processing of the system.

5.8.5 Evaluation Approach and Tools

The experiment to assess the performance of the proposed framework was measured into two categories: one from the normal connected IoT devices viewpoint. The other one is from the simulated requests to capture the system behaviors. The normally connected sensor node is set up to acquire basic data for measurements. Later, this basic data was simulated for huge data for scaling the experiment. The testing was done to ensure the experimental testbed is feasible. The simulated experiment is done to test the performance while system scaling is conducted.

During the testing phase, the function of sending requests has been executed to simulate virtual IoT devices published in the specified time interval. The simulated IoT devices send requests or messages with an increment of 10^3(1000, 100,000, 1,000,000) to the IoT platform. In this experiment, the platform executes under the MQTT publishing operation with Quality of Service (QoS) level 0 – the client simply publishes the message, and there is no acknowledgment by the broker. The reason for choosing level 0 is to avoid the system overhead created by acknowledgment messages. The data used for the system ingestion has been simulated and represents an observation of a unique and random phenomenon. The estimated average size of the

data generated is 240 bytes. It implies that this amount of data is ingested to the platform's backend services from the sensor nodes, and container controller.

The experiment aiming to collect CPU, throughputs, and memory metrics usages is conducted by using cAdvisor as container resources usage analysis platform and Prometheus in combination with Grafana for monitoring the proposed platform as shown in Table 5.3.

Table 5. 3: Specifications of environments used for performance analysis.

Specifications	Test Runner	Deployment Server
CPU	4.5GHz Intel Core TM (4 cores)	4.5GHz Intel Core TM (4 cores)
Memory	8GB DDR3	8GB DDR3
Storage	500GB HDD	500GB HDD
OS	Linux x86_64	Ubuntu 20.04 LTS
Installed Software	Docker swarm, MQTT Microservices, NGINX, PostgreSQL, Redis, cAdvisor, Prometheus, Grafana	Docker swarm, MQTT, Microservices, NGINX, PostgreSQL, Redis, cAdvisor, Prometheus, Grafana

5.8.6 Experiment Results

This section details the performance analysis conducted given different parameters. The key concerns are about the reliability and scalability of the system based on the changing requests.

5.8.6.1 Performance in Terms of System Reliability and Scalability

Figure 5.13 shows different metrics captured for performance evaluation in case the real IoT sensor node device is sending data. In the section on CPU usage, the average performance results show that the maximum usage is 2.25GHz while the minimum usage is 0.25GHz, this is considered a benchmark to measure if any increase in the load can affect the CPU usage.

The evaluation shows that due to the use of microservices, there is no single core overload, the graph shows that all works are distributed among every available 4 cores used in this testing with an average of around (0.4) 40%. The engagement of all cores implies an increase in system throughput.

Figure 5. 13: Resource usages such as CPU, memory, and throughput with normal sensor node set up.

On the graph of throughput, the dominant operation is the reception as the data are being sent from the sensor node to the platform. The observed maximum value of the reception is 46,000 bytes per second. It shows a minimum of around zero at some specific periods. The reason for some points with zero throughputs may depend on the sensor node device set to send data at an interval of 5 seconds. The maximum memory usage in this configuration is around 4,600 Megabytes while the minimum one is around 4,200 Megabytes. This implies the average usage of 4,400 Megabytes of memory.

Figures 5.14 and 5.15 depict the observations of the system behavior after applying the generated IoT sensor requests. In predicting the platform scalability, 10,000 IoT sensor requests were generated. In these graphs, there was a considerable increase in CPU usage with an average of (0.75) 75%. The memory usage maximum was around 4800 megabytes while the minimum used was around 4400 megabytes. This implied an average of 4600 megabytes of memory usage. In this experiment, the throughput graph showed a maximum of around 50,000 bytes/second with a minimum of around zero, generating an average of 25,000 bytes/second.

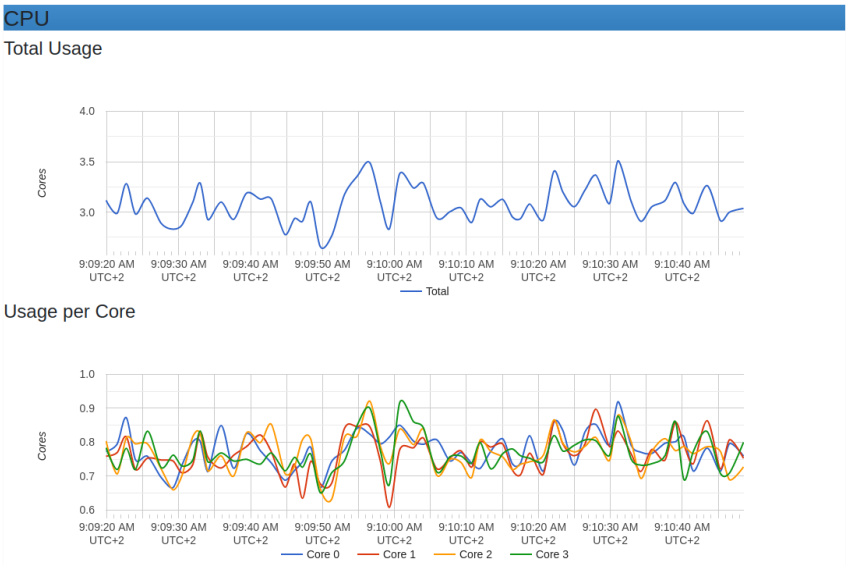


Figure 5. 14: CPU resources with 10,000 generated sensor requests.

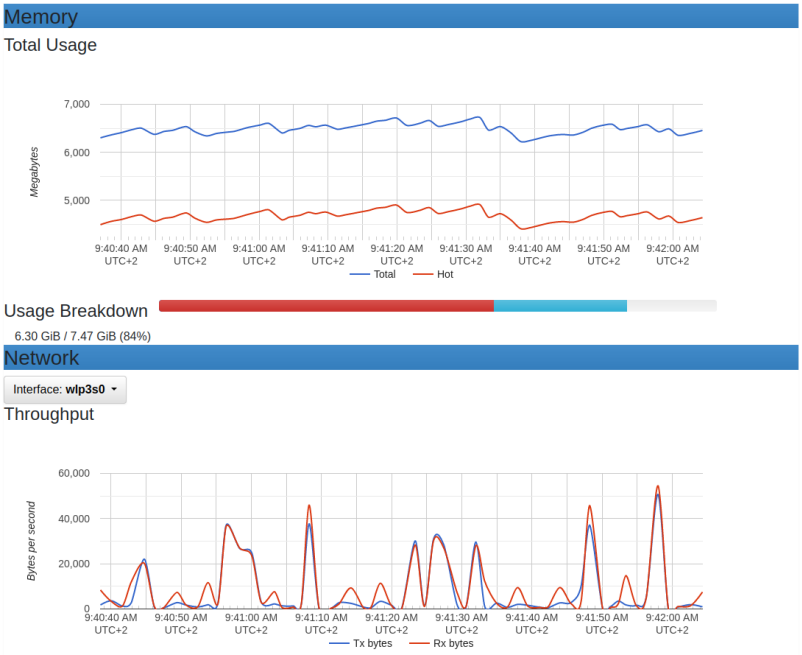


Figure 5. 15: Resource usage for memory and throughput with 10,000 generated sensor requests.

Figures 5.16 and 5.17 show considerable increases, especially in CPU core usage, with the generation of 1,000,000 requests. The average usage of all CPU cores was around (0.8) 80%. The maximum memory of 5000 megabytes and the minimum of 4500 megabytes of RAM was consumed. The average memory usage was around 4750 megabytes. The maximum throughput measured was equal to 60,000 bytes per second, with an average of 30,000 bytes per second.

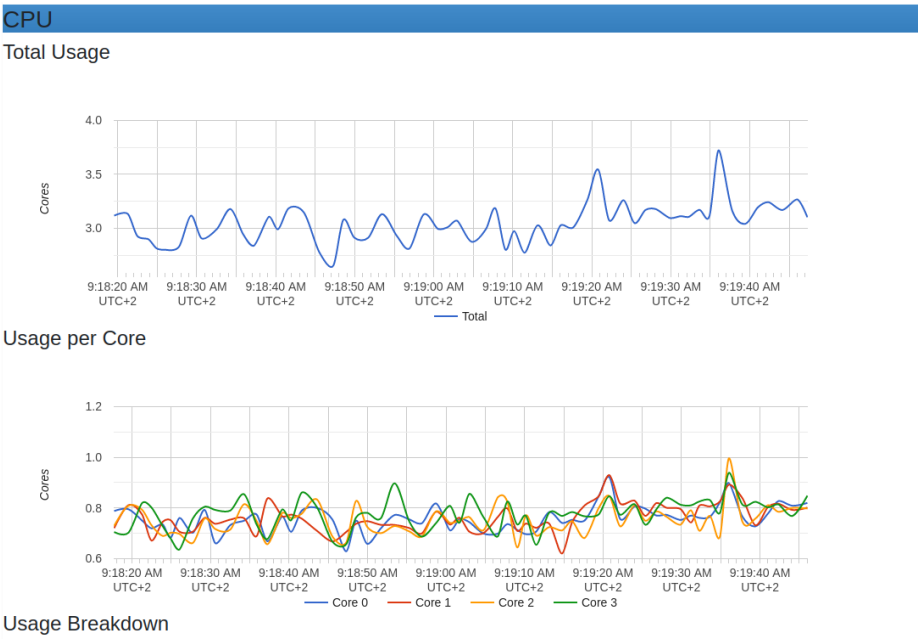


Figure 5. 16: CPU resource with 1,000,000 generated sensor requests.

Figure 5. 17: Resource usage for memory and throughput, with 1,000,000 generated sensor requests.

5.8.6.2 Performance Measurement in Terms of Resources Usages at the Scale-up.

The second experiment is conducted by increasing the number of nodes to track how the system behaves in case it is scaled up. This experiment is conducted by setting the number of nodes into 3, and the observations are measured using the same performance tools discussed (Prometheus and Grafana). In this experimental setup, the assumption is to scale up the containers to observe the system behaviors. The experiment is detailed with two entries: scale the application in terms of container services, and by generating different requests to observe the system. In this experiment, the observations are sticking to CPU, Memory, and Traffic metrics behaviors.

Figure 5.18 visualizes the memory and CPU usages in the scaled 3 nodes, using 1000 generated requests. The graph of memory usage showed that the increases in requests could increase the amount of consumed memory, but for the time being, there was no increase. The CPU usage graph showed that, while at the time, there was a small increase in usage, this increase may have depended on the internal cooperation of all container services involved in the data processing.

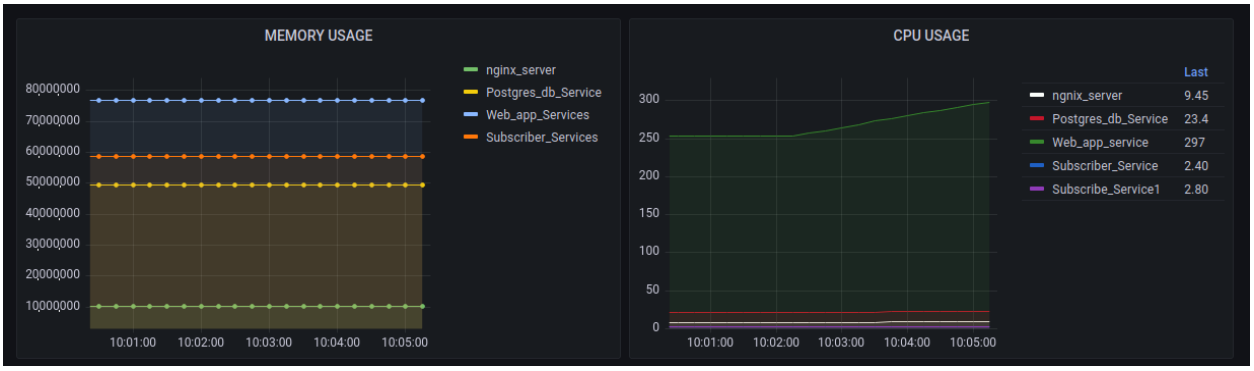


Figure 5. 18: Performance of the platform after scaling up with the requests of 1000—memory and CPU usage.

The performance metrics were concerned with the transmission cost of the network as shown in Figure 5.19. The main point of tracking those parameters was to observe how the internal containers were costly while exchanging message requests. In network transmission, the busy container was cAdvisor, as it was concerned with the huge number of requests being generated.

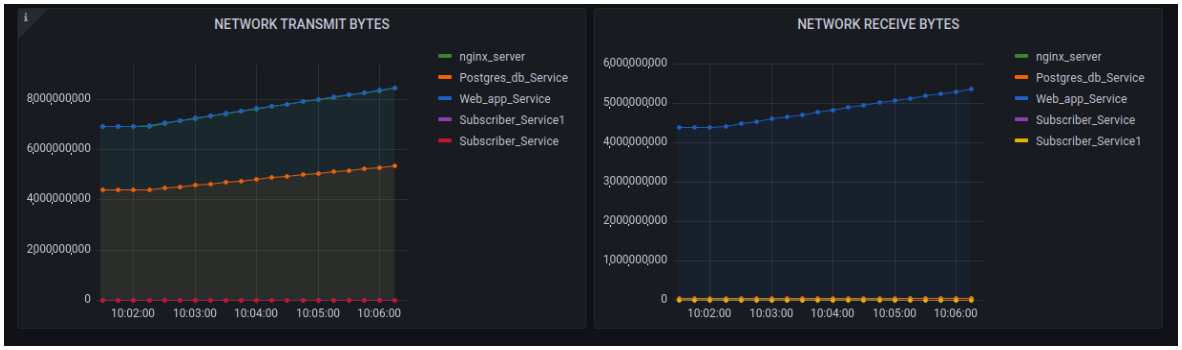


Figure 5. 19: Performance of the platform after scaling up with requests to 1000 network data transmissions.

Figure 5.20 shows the visualization of the system after increasing the number of requests up to 1,000,000. The graph shows that CPU and memory did not change significantly, but the trend increased compared to Figure 5.18.

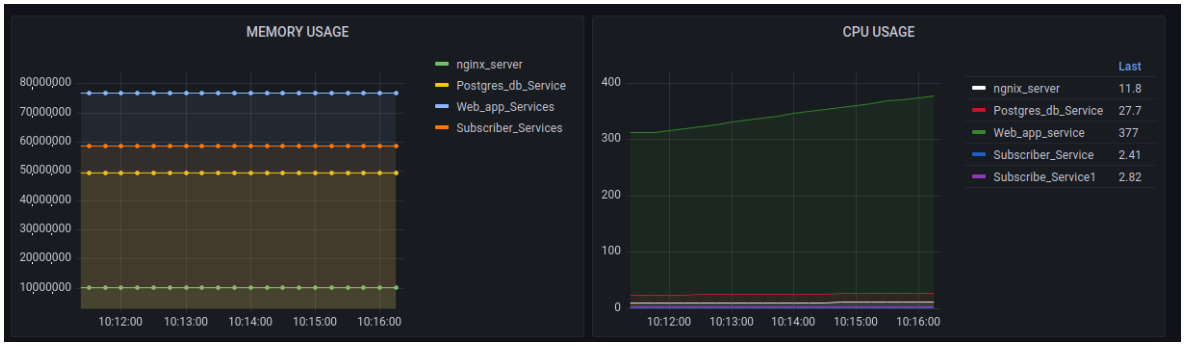


Figure 5. 20: Performance of the platform after scaling up with 1,000,000 requests: Memory and CPU usage.

Figure 5.21 shows the increase in performance for the network transmission, as this experiment did not discuss testing the user access of the requests at the user end side.

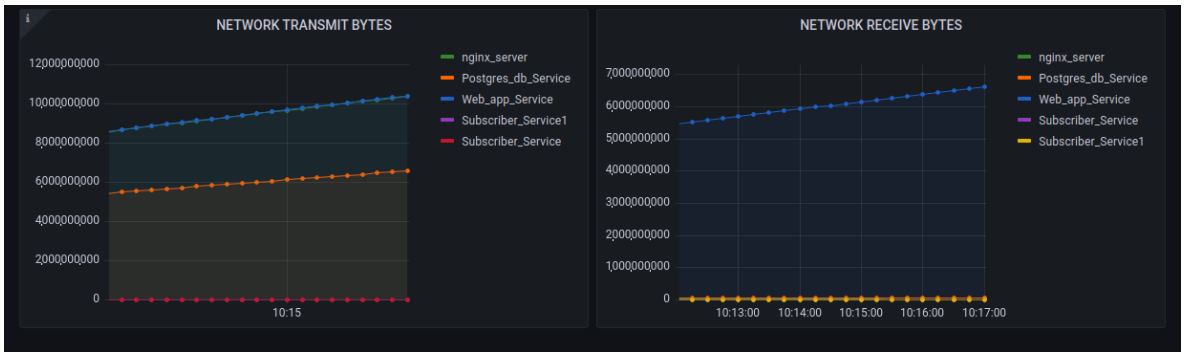


Figure 5. 21: Performance of the platform after scaling up the requests to 1,000,000 network data transmissions.

Figures 5.20 and 5.21 demonstrated the overall behavior of the platform using 3 nodes. Figure 5.22 shows the increase of the CPU usage as the number of requests increased.

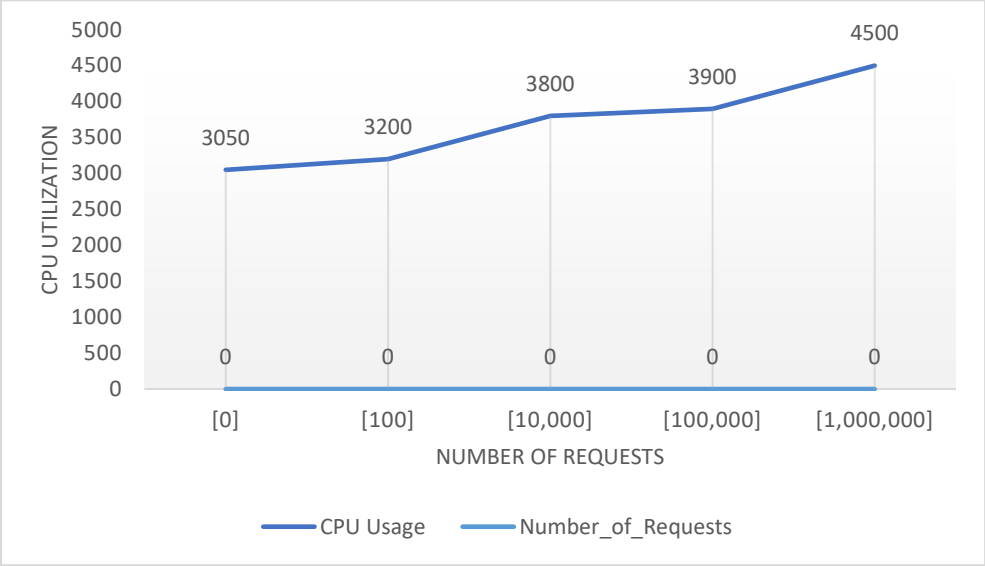


Figure 5. 22: Performance of the scaled system by 3 nodes: - CPU usages.

Figure 5.22 shows the CPU usage performance, while Figure 5.23 visualizes the performance of the system in terms of memory usage and network transmission. It is observed that the memory does not increase in consideration of the increase of the requests as seen in the observations detailed in the above figures. The network transmission (Tx) at the receiver side (Rx) does not increase much as the experiment performed is for generating requests to be sent to the platform. The side of generating requests trying to access the generated data was out of scope in this research.

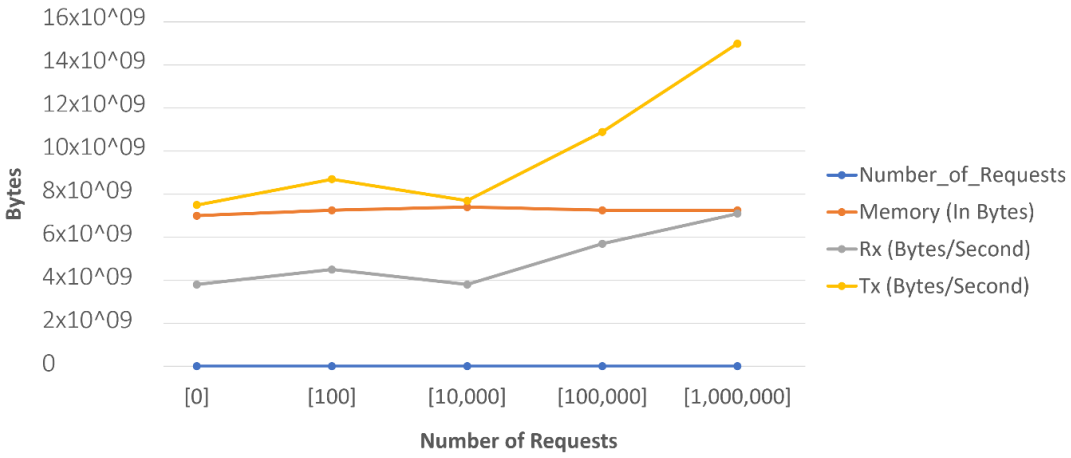


Figure 5. 23: Performance of the scaled system by 3 nodes: - Memory and Transmissions (Rx, and Tx).

The performance was measured experimentally on 3 nodes by varying different requests 0, 100, 10.000, 100.000, and 1.000,000 respectively. Using Grafana and Prometheus, the system resources considered are CPU utilization, memory usage, as well as data transmission (receiving and transmitting bytes). The results showed that due to the containers, the CPU used did not cost high by increasing the number of requests. For memory resources, the increases do not cost any attention. The transmission increases considerably to the bytes received starting from 10.000 requests. The network transmission at the transfer side shows tremendous changes but it is seen that at hundred requests the graph starts to go down up to ten thousand, but then it increases.

The implementation of a comprehensive solution for the whole IoT lifecycle is presented in this research. The lifecycle presented fits into four phases: data capture, communication, analysis, and implementation. A generic framework architecture based on the above layers is defined in the DevOps paradigm. The architecture complies with the main system requirements of all proposed IoT solutions.

The proposed IoT solutions are discussed in the building fire prevention sector, to improve building comfort and security. Different challenges have been discussed concerning the scalability, updating, and performance of IoT devices with low computing capability and power sources. The IoT devices are currently installed in buildings accessing the building's electrical power resources.

The proposed framework for data transmission and management, the nonfunctional requirement was considered to address all critical cases to be solved by IoT solutions. It also showed that some computation was possible at the edge side, minimizing the decision time delay.

Data transmissions were considered by ensuring data transfer in real-time using MQTT-based protocols, along with data packaged in JSON format—which has been considered lightweight—to ensure low network bandwidth consumption. For services, scalability, and reusability, the container-based and microservices solution was proposed, using docker. It satisfied the horizontal scalability of the framework. The sensor nodes, considered docker workers for updating and easy connectivity, were under the control of the docker swarm manager. Data sent from sensor nodes were managed by the Nginx web server, which also acted as a load balancer in case of the huge amount of user requests coming from the front-end.

The IoT framework presented in this research was validated and used in a smart building scenario to contribute to fire prevention situations. The system was deployed in the UR–KALISIMBI block site for three months to experiment with the challenges of the data capture

process. The performance metrics were extracted to visualize if they were suitable for adoption by the whole University.

Finally, in terms of future work, authors are interested in implementing the front-end to perform the user test and conducting standards adoption to improve IoT platform interoperability. Further, integrating the alerting functions to notify the concerned ERC responsible for building control. Addressing the use of solar panels to address the independence of the sensor node in case there is a power shortage will be considered. Another possible extension of the framework is to add features supporting IoT device mobility.

Eric Hitimana, July 2023,

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

This thesis presented the method leading to the development of the carried research on the analysis and performance of IoT framework for fire incidence applications. The integration and mathematical modeling approaches were proposed as solutions to the direction of fire outbreak and occupant detection, counting and prediction. This work demonstrated that continuous integration could go a long way toward improving room occupant detection, counting and prediction. In this research, IoT-based room occupant detection, counting and prediction for fire outbreaks stakeholders' notification through a centralized containerized platform were recommended. A spruce fire detection build based on the designed strategies of IoT methodologies for data acquisition was presented. An IoT-based framework with data analysis using deep learning methods for occupancy prediction were presented and tested. The containerized approaches to improve the proposed architecture are implemented with performance analysis measures to track the accuracy of the framework.

IoT and machine learning have been applied to the fire outbreaks situation to improve the current situation. In this thesis, different indoor parameters to prove the occupant's existence using edge, mist, fog and cloud layer architecture and DevOps were used to achieve a specific purpose. The DevOps methodology for continuous integration of edge sensor nodes is to allow multiple integrations for system scale. The results demonstrate that the containerized and performance measures process is a simple but powerful tool.

In future work, there is a need to look for energy harvesting techniques for the connected devices as they may rely on separate power sources to avoid the cause of fire in a building.

6.2 Recommendations

There is no data available for researchers to be used for prediction or forecasting the indoor fire risk from concerned units. For the sake of research facilitation in Rwanda and Eastern Africa in general, there must be a public dataset for research purposes. Researchers should explore more machine learning models to find out which ones can provide improved prediction accuracy for the fire prediction.

During the development of the data acquisition prototype, it had observed that the available devices (sensors, and actuators) are not accurate and require much calibration, which sometimes does not comply with the needed data accuracy. It is recommended for researchers and industry to avail accurate devices to support the community for emergency solutions. The Eastern African researchers and Rwandans, in general, are recommended to conduct research to provide tangible solutions. The continent and the region, in general, is looking the smart solutions for early warnings and prevention mechanisms to support the community in a disaster such as fire, landslides, and flood.

LIST OF REFERENCES

[1]. Christopoulos, K., Antonopoulos, C., Voros, N., & Orfanoudakis, T. "Building Automation Systems on the Internet of Things World". Components and Services for IoT Platforms **2016**, 355–375. **DoI**:10.1007/978-3-319-42304-3_18.

[2]. Maher Ala'raj, Mohammed Radi, Maysam F. Abbod, Munir Majdalawieh, Marianela Parodi, "Data-driven based HVAC optimization approaches: A Systematic Literature Review", Journal of Building Engineering, Volume 46, **2022**, 103678, ISSN 2352-7102. https://doi.org/10.1016/j.jobe.2021.103678.

[3]. Aguirre, Benigno. (**2014**). "Planning, Warning, Evacuation, and Search and Rescue: A Review of the Social Science Research Literature". 2014. The University of Delaware. https://www.researchgate.net/publication/267382489_Planning_Warning_Evacuation_and_Search_and_Rescue_A_Review_of_the_Social_Science_Research_Literature. (Accessed on 20 May 2021).

[4]. J., Y. Honda, Z.W. Kundzewicz, N. Arnell, G. Benito, J. Hatfield, I.F. Mohamed, P. Peduzzi, S. Wu, B. Sherstyukov, K. Takahashi, and Z. Yan, **2012**: Changes in impacts of climate extremes: Human Systems and Ecosystems. In Managing the Risks of Extreme Events and Disasters to Advance Climate Change Adaptation [Field, C.B., V. Barros, T.F. Stocker, D. Qin, D.J. Dokken, K.L. Ebi, M.D. Mastrandrea, K.J. Mach, G.-K. Plattner, S.K. Allen, M. Tignor, and P.M. Midgley (eds.)]. A Special Report on Working Groups I and II of the Intergovernmental Panel on Climate Change (IPCC). Cambridge University Press, Cambridge, UK, and New York, NY, USA, pp. 231-290. https://www.ipcc.ch/site/assets/uploads/2018/03/SREX-Chap4_FINAL-1.pdf. (Accessed on 21 May 2021).

[5]. Ministry of Natural Resources, "Rwanda National Land Management and Use Authority", Rwanda National Land Use Planning Guidelines, December **2017**. https://www.environment.gov.rw/fileadmin/user_upload/Moe/Publications/Policies/Revised_National_Land_Policy-Final_Version_2019.pdf. (Accessed on 21 March 2021).

[6]. Ministry of Disaster Management and Refugee Affairs, Rwanda, (MIDIMAR), "The Rwanda Disaster Management Policy, Revision of the **2009** National Disaster Management Policy", June 2012. https://www.preventionweb.net/publication/rwanda-national-disaster-management-policy-2012. (Accessed on 21 March 2021).

[7]. International Labour Office, "Infrastructure Development, the Construction Sector and Employment in Rwanda", Joint EU-ILO Government of Rwanda project: Strengthening the impact on employment of sectoral and trade policies Infrastructure Development, the Construction Sector and Employment in Rwanda, September **2018**. https://www.ilo.org/wcmsp5/groups/public/---ed_emp/---ifp_skills/documents/publication/wcms_723290.pdf. (Accessed on 21 March 2021).

[8]. WWF Internationals 2020m "Fires, forests, and the future: A crisis ranging out of control?". https://wwfeu.awsassets.panda.org/downloads/wwf_fires_forests_and_the_future_report.pdf. (Accessed on 21/03/2021).

[9]. Secretariat, E.A.C. "Disaster Risk Reduction and Management Strategy"; Arusha, Tanzania, **2012**. https://www.preventionweb.net/files/EAC_DRRMS(2012-2016)version_1.4%5B1%5D.pdf. (Accessed on 21/03/2021).

[10]. M. In, C. Of, and E. Management. The Republic of Rwanda Ministry in Charge of Emergency Management National Contingency Plan; **2019**; pp. 1–55. https://www.preventionweb.net/files/28709_gendermainstreamingindisastermanage.pdf. (Accessed on 21/03/2021).

[11]. Bettin, G.; Zazzaro, A. "The Impact of Natural Disasters on Remittances to Low- and Middle-income Countries". Dev. Work. Pap. **2016**, 54, 481-500. https://doi.org/10.1080/00220388.2017.1303672.

[12]. Park, H.; Rhee, S.B. "IoT-Based Smart Building Environment Service for Occupants' Thermal Comfort". Available online: https://www.hindawi.com/journals/js/2018/1757409/ (Accessed on 5 January **2020**).

[13]. East African Community. "Disaster Risk Reduction and Management Strategy (2012–**2016**)". Available online: https://www.ifrc.org/Global/ Publications/ IDRL/regional/EAC_DRRMS 2012-2016.pdf. (Accessed on 6 January 2020).

[14]. Mondal, D.R. "High Risk of Post-Earthquake Fire Hazard in Dhaka, Bangladesh". Fire **2019**, 2, 24. https://doi.org/10.3390/fire2020024.

[15]. Lam, H.C.Y.; Haines, A.; McGregor, G.; Chan, E.Y.Y.; Hajat, S. "Time-Series Study of Associations between Rates of People Affected by Disasters and the El Niño Southern Oscillation (ENSO) Cycle". Int. J. Environ. Res. Public Health **2019**, 16, 3146. **DOI**: 10.3390/ijerph16173146.

[16]. Rwanda Ministry of Infrastructure, "Urbanization, Human Settlement, and Housing Development". https://www.mininfra.gov.rw/digital-transformation-1. (Accessed: 28 December 2022).

[17]. Twigg, J.; Christie, N.; Haworth, J.; Osuteye, E.; Skarlatidou, A. "Improved methods for fire risk assessment in low-income and informal settlements". Int. J. Environ. Res. Public Health **2017**, 14, 139. **DOI**: 10.3390/ijerph14020139.

[18]. Xin, J.; Huang, C. "Fire risk analysis of residential buildings based on scenario clusters and its application in fire risk management". Fire Saf. J. 2013, 62, 72–78. https://doi.org/10.1016/j.firesaf.2013.09.022.

[19]. Wang, Y.; Cai, L.; Chen, Y. "Fuzzy Comprehensive Evaluation Method, and Its Application in Existing Buildings Safety". In Proceedings of the **2015** International Forum

on Energy, Environment Science and Materials, Shenzhen, China, 25–26 September 2015. **DOI**: 10.2991/ifeesm-15.2015.251.

[20]. Vladimir M. Cvetković, Aleksandar Dragašević, Darko Protić, Bojan Janković, Neda Nikolić, Predrag Milošević, "Fire safety behavior model for residential buildings: Implications for disaster risk reduction", International Journal of Disaster Risk Reduction, Volume 76, **2022**,102981, ISSN 2212-4209, https://doi.org/10.1016/j.ijdrr.2022.102981.

[21]. Yi, G.; Li, H.Q. "Fuzzy Comprehensive Evaluation of Fire Risk on High-Rise Buildings". Procedia Eng. 2011, 11, 620–625. https://doi.org/10.1016/j.proeng.2011.04.705.

[22]. Moshashaei, P.; Alizadeh, S.S. "Fire risk assessment: A systematic review of the methodology and functional areas". Iran. J. Health Saf. Environ. 2017, 4, 654–669. Corpus ID: 55263658

[23]. [Ghai, S.K.; Thanayankizil, L.V.; Seetharam, D.P.; Chakraborty, D. "Occupancy detection in commercial buildings using opportunistic context sources". In Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications Workshops, Lugano, Switzerland, 19–23 May 2012. **DOI**:10.1109/PerComW.2012.6197536.

[24]. Zikos, S.; Tsolakis, A.; Meskos, D.; Tryferidis, A.; Tzovaras, D. "Conditional Random Fields—Based approach for real-time building occupancy estimation with multi-sensory networks". Autom. Constr. 2016, 68, 128–145. https://doi.org/10.1016/j.autcon.2016.05.005.

[25]. Jia, Mengda & Komeily, Ali & Wang, Yueren & Srinivasan, Ravi. "Adopting Internet of Things for the Development of Smart Buildings: A review of Enabling Technologies and Applications". Automation in Construction, 2019. DoI:101.111-126. 10.1016/j.autcon.2019.01.023.

[26]. Chang-Su, Ryu, "IoT-based Intelligent for Fire Emergency Response Systems", International Journal of Smart Home 2015, vol. 9, no. 3, pp. 161-168. http://dx.doi.org/10.14257/ijsh.2015.9.3.15.

[27]. The EastAfrican https://www.theeastafrican.co.ke/. (Accessed: 28 December 2022).

[28]. March, A.; Kornakova, M. "Urban Planning for Disaster Recovery"; Elsevier Science: Amsterdam, The Netherlands, 2017. ISBN: 9780128042762. eBook ISBN: 9780128043233.

[29]. Twingg, J. "Disaster Risk Reduction: Mitigation and Preparedness in Development and Emergency Programming"; Humanitarian Practice Network, Overseas Development Institute: London, UK, 2004. https://www.humanitarianlibrary.org/sites/default/files/2014/02/HPN_GoodPracticeReview9DisasterRiskReduction.pdf. (Accessed: 28 December 2022).

[30]. Enas A. A. Abuhamda, Islam Asim Ismail Tahani, R. K. Bsharat. "Understanding Quantitative and Qualitative Research Methods: A Theoretical Perspective for Young Researchers". February **2021** International Journal of Research 8(2):71-87. **DOI**:10.2501/ijmr-201-5-070.

[31]. W. J. W. Botzen, O. Deschenes, and M. Sanders, "The Economic Impacts of Natural Disasters: A Review of Models and Empirical Studies", vol. 13, no. 2, pp. 167–188, Aug. **2019**, **DOI**: 10.1093/REEP/REZ004.

[32]. East African community, "Data collection survey on the draft disaster risk management strategy of the east African community", October **2012**, Japan International Cooperation Agency, East System Science Co. Ltd. https://openjicareport.jica.go.jp/pdf/12083614_01.pdf. (Accessed: 21 June 2022).

[33]. Cavallini, M & Papagni, M.F. & Baruffaldi Preis, Franz. "Fire Disasters in the Twentieth Century". Annals of burns and fire disasters **2007**. 20. 101-3. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3188063/

[34]. UCLA Health, "Classes of Fires & Fire Extinguishers - Safety". https://www.uclahealth.org/safety/ambulatory-safety/ambulatory-fire-and-life-safety-program/classes-fires-fire-extinguishers. (Accessed: 2023-01-28).

[35]. Babrauskas, V. "Ignition: A Century of Research and an Assessment of Our Current Status". J. Fire Prot. Eng. **2007**, 17, 165–183. https://doi.org/10.2307/40323168.

[36]. Troitzsch, J.H. "Fires, statistics, ignition sources, and passive fire protection measures". J. Fire Sci. **2016**, 34, 171–198. https://doi.org/10.1177/0734904116636642.

[37]. Fonollosa, J.; Solórzano, A.; Marco, S. "Chemical Sensor Systems and Associated Algorithms for Fire Detection: A Review". Sensors **2018**, 18, 553. https://doi.org/10.3390/s18020553.

[38]. Chu, M.L.; Law, K.H. "Incorporating Individual Behavior, Knowledge, and Roles in Simulating Evacuation". Fire Technol. **2018**, 55, 437–464. DOI:10.1007/S10694-018-0747-6.

[39]. Tavares, R.M. "Evacuation Processes Versus Evacuation Models: Quo Vadimus"? Fire Technol. **2009**, 45, 419–430. DOI:10.1007/s10694-008-0063-7.

[40]. Shi, J.; Ren, A.; Chen, C. "Agent-based evacuation model of large public buildings under fire conditions". Autom. Constr. **2009**, 18, 338–347. https://doi.org/10.1016/j.autcon.2008.09.009.

[41]. Babrauskas, V. "Estimating room flashover potential". Fire Technol. 1980, 16, 94–103. **DOI**:10.1007/BF02981655.

[42]. Thomas, P.H.; Bullen, M.L.; Quintiere, J.G.; McCaffrey, B.J. "Flashover, and instabilities in fire behavior". Combust. Flame 1980, 38, 159–171. https://doi.org/10.1016/0010-2180(80)90048-6.

[43]. Peacock, R.D.; Reneke, P.A.; Bukowski, R.W.; Babrauskas, V. "Defining flashover for fire hazard calculations". Fire Saf. J. 1999, 32, 331–345. **DOI**:10.1016/S0379-7112(03)00027-4.

[44]. Mohammadreza Shokouhi, Khadijeh Nasiriani et al. "Exploring barriers and challenges in protecting residential fire-related injuries: a qualitative study KEY WORDS Fires Residential buildings Injury and death Safety". J Inj Violence Res, 11, 1, **2019**. **DOI**:10.5249/jivr.v11i1.1059.

[45]. El Dorado County Fire Safe Council, Chapter 4: "Fire Risk Mitigation Strategies", Chapter 4, **2017**, February. https://www.edcfiresafe.org/wp-content/uploads/2017/02/Chapter-4.pdf. (Accessed: 2023-01-28).

[46]. Chao, L.; Zhang, K.; Wang, J.; Feng, J.; Zhang, M. "A comprehensive evaluation of five evapotranspiration datasets based on ground and grace satellite observations: Implications for improvement of evapotranspiration retrieval algorithm". Remote Sens. **2021**, 13, 2414. https://doi.org/10.3390/rs13122414.

[47]. Zhao, X.; Xia, H.; Pan, L.; Song, H.; Niu, W.; Wang, R.; Li, R.; Bian, X.; Guo, Y.; Qin, Y. "Drought monitoring over Yellow River basin from 2003–2019 using reconstructed MODIS land surface temperature in Google Earth Engine". Remote Sens. **2021**, 13, 3748. https://doi.org/10.3390/rs13183748.

[48]. Jaafari, A.; Mafi-Gholami, D.; Thai Pham, B.; Tien Bui, D. "Wildfire probability mapping: Bivariate vs. multivariate statistics". Remote Sens. **2019**, 11, 618. https://doi.org/10.3390/rs11060618.

[49]. Mohajane, M.; Costache, R.; Karimi, F.; Pham, Q.B.; Essahlaoui, A.; Nguyen, H.; Laneve, G.; Oudija, F. "Application of remote sensing and machine learning algorithms for forest fire mapping in a Mediterranean area". Ecol. Indic. **2021**, 129, 107869. https://doi.org/10.1016/j.ecolind.2021.107869.

[50]. Younes Oulad Sayad, Hajar Mousannif, Hassan Al Moatassime, "Predictive modeling of wildfires: A new dataset and machine learning approach", Fire Safety Journal, Volume 104, **2019**, Pages 130-146, ISSN 0379-7112, https://doi.org/10.1016/j.firesaf.2019.01.006.

[51]. Olawoyin, R. "Nanotechnology: The future of fire safety". Saf. Sci. **2018**, 110, 214–221. DOI: 10.1016/J.SSCI.2018.08.016.

[52]. Corbett, G.P.; Brannigan, F.L. "Brannigan's Building Construction for the Fire Service Includes Navigate Advantage Access"; Jones & Bartlett Learning: Burlington, MA, USA,

2019. https://www.psglearning.com/catalog/productdetails/9781284177312. (Accessed on 21st November 2022).

[53]. National Fire Protection Agency, https://www.nfpa.org/Assets/files/AboutTheCodes/1021/Standards%20Guide_1021_1407.pdf. (Accessed on 21st November 2022).

[54]. Rwanda National Police, "Police call for precautionary measures against fire outbreaks", https://www.police.gov.rw/media-archives/news-detail/?tx_news_pi1%5Bnews%5D=15940&cHash=dd9fe61e1b20c2a1aa62786e754fa938. (Accessed: 2023-01-28).

[55]. Gakpe BK, Mahama PY. "Reportage of stories on fire outbreaks in Ghana: an analysis of the Daily Graphic and the Chronicle". New Media Mass Commune **2014**; 24:1e11. DOI: 10.7176/NMMC.VOL241-11.

[56]. Ahrens, M.; Maheshwai, R. "Home Structure Fires; National Fire Protection Agency: Quincy", MA, USA, **2021**. https://injuryfacts.nsc.org/home-and-community/safety-topics/fire-related-fatalities-and-injuries/. (Accessed: 2023-01-28).

[57]. U.S. "Fire Administration. Statistics". Available online: https://www.usfa.fema.gov/statistics. (Accessed on 3 September 2022).

[58]. Brushlinsky, N.; Sokolov, S.; Wagner, P.; Messerschmidt, B. "The CTIF World Fire Statistics Report № 27". CTIF International Association of Fire and Rescue Services Center of Fire Statistics. **2022**. Available online: https://www.ctif.org/news/ctif-world-fire-statistics-report-no-27-now-available-download#:~:text=The%20current%20report%20%E2%84%96%2027,since%20the%20start%20in%202005. (Accessed on 3 September 2022).

[59]. Penuel, K.B.; Statler, M. "Encyclopedia of Disaster Relief"; SAGE: Los Angeles, CA, USA, **2011**; Volume 2. https://doi.org/10.1007/978-3-319-77282-0_1.

[60]. Minnery, J. "Urban Planning for Disaster Recovery"; Taylor & Francis: London, UK, **2018**. **DOI**:10.1080/08111146.2018.1443560.

[61]. Johnston, D.; Becker, J.; Cousins, J. "Lifelines, and urban resilience. In Disaster Resilience: An Integrated Approach"; Paton, D., Johnston, D., Eds.; Charles C Thomas Publisher: Springfield, IL, USA, **2017**. ISBN (Electronic): 9780398091705. ISBN (Print): 9780398091699.

[62]. Hung, H.C.; Yang, C.Y.; Chien, C.Y.; Liu, Y.-C. "Building resilience: Mainstreaming community participation into the integrated assessment of resilience to climatic hazards in metropolitan land use management". Land Use Policy **2016**, 50, 48–58. **DOI**: 10.1016/j.landusepol.2015.08.029.

[63]. Dickson, E.; Baker, J.L.; Hoornweg, D.; Asmita, T. "Urban Risk Assessments: An Approach for Understanding Disaster and Climate Risk in Cities"; The World Bank:

Washington, DC, USA, **2012**. https://openknowledge.worldbank.org/handle/10986/12356.

[64]. Alamdar, F.; Kalantari, M.; Rajabifard, A. "An evaluation of integrating multi-sourced sensors for disaster management". Int. J. Digit. Earth **2015**, 8, 727–749. **DOI**:10.1080/17538947.2014.927537.

[65]. Masoumi, Z.; van Genderen, J.L.; Mesgari, M.S. "Modelling and predicting the spatial dispersion of skin cancer considering environmental and socio-economic factors using a digital earth approach". Int. J. Digit. Earth **2018**, 1–22. **DOI**:10.1080/17538947.2018.1551944.

[66]. Tomaszewski, B. "Geographic Information Systems (GIS) for Disaster Management"; CRC Press: Boca Raton, FL, USA; Taylor & Francis Group: New York, NY, USA, **2014**; p. 304. **DOI**:10.1515/jhsem-2014-0082.

[67]. Li, G.; Zhao, J.; Murray, V.; Song, C.; Zhang, L. "Gap analysis on open data interconnectivity for disaster risk research". Geo-Spat. Inf. Sci. **2019**, 22, 45–58. **DOI**: 10.1080/10095020.2018.1560056.

[68]. Jennings, C.R. "Social, and economic characteristics as determinants of residential fire risk in urban neighborhoods: A review of the literature". Fire Saf. J. **2013**, 62, 13–19. **DOI**: 10.1016/j.firesaf.2013.07.002.

[69]. Pohl, C.; Van Genderen, J. "Remote Sensing Image Fusion: A Practical Guide"; CRC Press: Boca Raton, FL, USA, **2016**. DOI: https://doi.org/10.1201/9781315370101.

[70]. Lorincz, K., Malan, D., R.F, T., Jones, F., Nawoj, A., Clavel, A., Welsh, M. (**2004**). "Sensor Networks for Emergency Response: Challenges and Opportunities". IEEE Pervasive Computing, 16-23. **DOI**: 10.1109/MPRV.2004.18.

[71]. Lam, K., & Srivastava, V. (**2005**). "Living in the Intelligent Workplace Structuring and Managing Building Operation Information". Proceedings of the Fifth International Conference for Enhanced Building Operations. ESL. Proceedings of the Fifth International Conference for Enhanced Building Operations, Pittsburgh, Pennsylvania, October 11-13, 2005. https://core.ac.uk/download/pdf/79624179.pdf.

[72]. Alahmad, M., Nader, W., Brumbaugh, A., Cho, Y., Ci, S., Sharif, H., Neal, J. (**2011**). "The "BIM's 4D+" dimension: Real-time energy monitoring". IEEE GCC Conference and Exhibition (GCC), 589-592. January 2019 MATEC Web of Conferences 281:01013. DOI:10.1051/matecconf/201928101013.

[73]. Ochoa, S., & Santos, R. (**2013**). "Human-centric wireless sensor networks to improve information availability during urban search and rescue activities". Information Fusion. Volume 22, March 2015, Pages 71-8. https://doi.org/10.1016/j.inffus.2013.05.009.

[74]. Marzouk, M., & Abdelaty, A. (**2014**). "BIM-based framework for managing the performance of subway stations". Automation in Construction, 70-77. https://doi.org/10.1016/j.autcon.2014.02.004.

[75]. Marzouk, M., Abdelbasset, I., & Al-Gahtani, K. (**2015**). Tracking Indoor Air Quality of Buildings Using BIM. 5th International/11th Construction Specialty Conference. Columbia: CSCE. **DOI**: 10.14288/1.0076312.

[76]. Pagano, S.; Peirani, S.; Valle, M. "Indoor ranging and localization algorithm based on received signal strength indicator using statistic parameters for wireless sensor networks". IET Wirel. Sens. Syst. **2015**, 5, 243–249. https://doi.org/10.1049/iet-wss.2014.0027.

[77]. Rathore, M.M.; Paul, A.; Hong, W.H.; Seo, H.; Awan, I.; Saeed, S. "Exploiting IoT, and big data analytics: Defining smart digital city using real-time urban data. Sustain". Cities Soc. **2017**. https://doi.org/10.1016/j.scs.2017.12.022.

[78]. Li, W.; Zhang, W. "Sensor selection for improving the accuracy of target localization in wireless visual sensor networks". IET Wirel. Sens. Syst. **2012**, 2, 293–301. **DOI**: 10.1049/iet-wss.2012.0033.

[79]. Rathore, M.M.; Ahmad, A.; Paul, A.; Wan, J.; Zhang, D. "Real-time medical emergency response system: Exploiting IoT and big data for public health". J. Med. Syst. **2016**, 40, 283. **DOI**: 10.1007/s10916-016-0647-6.

[80]. Yick, J.; Mukherjee, B.; Ghosal, D. "Wireless sensor network survey". Comput. Netw. **2008**, 52, 2292–2330. https://doi.org/10.1016/j.comnet.2008.04.002.

[81]. Potter, C.H.; Hancke, G.P.; Silva, B.J. "Machine-to-Machine: Possible applications in industrial networks". In Proceedings of the **2013** IEEE International Conference on Industrial Technology (ICIT), Cape Town, South Africa, 25–28 February 2013; pp. 1321–1326. **DOI**: 10.1109/ICIT.2013.6505864.

[82]. Kumar, A.; Hancke, G.P. "An energy-efficient smart comfort sensing system based on the IEEE 1451 standard for green buildings". IEEE Sens. J. **2014**, 14, 4245–4252. **DOI**: 10.1109/JSEN.2014.2356651.

[83]. Silva, B.; Fisher, R.M.; Kumar, A.; Hancke, G.P. "Experimental link quality characterization of wireless sensor networks for underground monitoring". IEEE Trans. Ind. Inform. **2015**, 11, 1099–1110. **DOI**: 10.1109/TII.2015.2471263.

[84]. Chiwewe, T.M.; Mbuya, C.F.; Hancke, G.P. "Using cognitive radio for interference resistant industrial wireless sensor networks: An overview". IEEE Trans. Ind. Inform. 2015, 11,1466–1481. **DOI**: 10.1109/TII.2015.2491267.

[85]. Tan, W.; Wang, Q.; Huang, H.; Guo, Y.; Zhan, G. "Mine Fire Detection System Based on Wireless Sensor Networks". In Proceedings of the Conference on Information Acquisition (ICIA'07), Seogwipo-si, Korea, 8–11 July **2007**. **DOI**: 10.1109/ICIA.2007.4295715.

[86]. Zhang, J.; Li, W.; Han, N.; Kan, J. "Forest fire detection system based on a ZigBee wireless sensor network". Front. For. China **2008**, 3, 369–374. DOI:10.1007/s11461-008-0054-3.

[87]. Cheung, W.-F., & Lin, Y.-C. (**2016**). "Development of a BIM-based Safety Monitoring System Integrated with WSN Technology". 601-608. **DOI**:10.1061/(ASCE)ME.1943-5479.0000760.

[88]. Cheng, M.-Y., Chiu, K.-C., Hsieh, Y.-M., Yang, I.-T., Chou, J.-S., & Wu, Y.-W. (**2017**). "BIM integrated smart monitoring technique for building fire prevention and disaster relief". Automation in Construction, 14-30. https://doi.org/10.1016/j.autcon.2017.08.027.

[89]. T. Zhang, Z. Wang, H.Y. Wong, W.C. Tam, X. Huang, F. Xiao, "Real-time forecast of compartment fire and flashover based on deep learning", Fire Safety Journal. 130 (**2022**) 103579. https://doi.org/10.1016/j.firesaf.2022.103579.

[90]. A. Dexters, R.R. Leisted, R. Van Coile, S. Welch, G. Jomaas, "Testing for knowledge: Application of machine learning techniques for prediction of flashover in a 1/5 scale ISO 13784-1 enclosure", Fire and Materials. 45 (**2021**) 708–719. https://doi.org/10.1002/fam.2876.

[91]. Z. Wang, T. Zhang, X. Wu, X. Huang, "Predicting transient building fire based on external smoke images and deep learning", Journal of Building Engineering. 47 (**2022**) 103823. https://doi.org/10.1016/j.jobe.2021.103823.

[92]. J. Wang, W.C. Tam, Y. Jia, R. Peacock, P. Reneke, E.Y. Fu, T. Cleary, "P-Flash – A machine learning-based model for flashover prediction using recovered temperature data", Fire Safety Journal. 122 (**2021**) 103341. https://doi.org/10.1016/j.firesaf.2021.103341.

[93]. L. chu Su, X. Wu, X. Zhang, X. Huang, "Smart performance-based design for building fire safety: Prediction of smoke motion via AI", Journal of Building Engineering. 43 (**2021**) 102529. https://doi.org/10.1016/j.jobe.2021.102529.

[94]. T. Buffington, J.M. Cabrera, A. Kurzawski, O.A. Ezekoye, "Deep-Learning Emulators of Transient Compartment Fire Simulations for Inverse Problems and Room-Scale Calorimetry", Fire Technology. 57(**2021**) 2859–2885. https://doi.org/10.1007/s10694-020-01037-2.

[95]. E.W.M. Lee, R.K.K. Yuen, S.M. Lo, K.C. Lam, G.H. Yeoh, "A novel artificial neural network fire model for prediction of thermal interface location in a single compartment fire", Fire Safety Journal. 39 (**2004**) 67– 87. https://doi.org/10.1016/S0379-7112(03)00092-4.

[96]. Y. Huang, X. Chen, L. Xu, K. Li, "Single Image Desmoking via Attentive Generative Adversarial Network for Smoke Detection Process", Fire Technology. 57 (**2021**) 3021–3040. https://doi.org/10.1007/s10694-021-01096-z.

[97]. Xayasouk, T.; Lee, H.M.; Lee, G. "Air pollution prediction using long short-term memory (LSTM) and deep autoencoder (DAE) models". Sustainability **2020**, 12, 2570. https://doi.org/10.3390/su12062570.

[98]. Le, X.H.; Ho, H.V.; Lee, G.; Jung, S. "Application of Long Short-Term Memory (LSTM) neural network for flood forecasting". Water **2019**, 11, 1387. https://doi.org/10.3390/w11071387.

[99]. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.V.; Liu, J. "LSTM network: A deep learning approach for short-term traffic forecast". IET Image Process. **2017**, 11, 68–75. https://doi.org/10.1049/iet-its.2016.0208.

[100]. Choi, E.; Cho, S.; Kim, D.K. "Power demand forecasting using long short-term memory (LSTM) deep-learning model for monitoring energy sustainability". Sustainability **2020**, 12, 1109. https://doi.org/10.3390/su12031109.

[101]. Liu, P.; Wang, J.; Sangaiah, A.; Xie, Y.; Yin, X. "Analysis and Prediction of Water Quality Using LSTM Deep Neural Networks in IoT Environment". Sustainability **2019**, 11, 2058. https://doi.org/10.3390/su11072058.

[102]. X. Zhang, X. Wu, Y. Park, T. Zhang, X. Huang, F. Xiao, A. Usmani, "Perspectives of big experimental database and artificial intelligence in tunnel fire research", Tunnelling and Underground Space Technology. 108 (**2021**) 103691. https://doi.org/10.1016/j.tust.2020.103691.

[103]. X. Wu, Y. Park, A. Li, X. Huang, F. Xiao, A. Usmani, "Smart Detection of Fire Source in Tunnel Based on the Numerical Database and Artificial Intelligence", Fire Technology. 57 (2021) 657–682. https://doi.org/10.1007/s10694-020-00985-z.

[104]. S. Hayou, A. Doucet, J. Rousseau, "On the impact of the activation function on deep neural networks training", 36th International Conference on Machine Learning, ICML **2019**. 2019-June (2019) 4746–4754. https://doi.org/10.48550/arXiv.1902.06853.

[105]. V. Dumoulin, F. Visin, "A guide to convolution arithmetic for deep learning", ArXiv Preprint ArXiv:1603.07285. (**2016**). https://doi.org/10.48550/arXiv.1603.07285.

[106]. Jaafari, A.; Zenner, E.K.; Panahi, M.; Shahabi, H. "Hybrid artificial intelligence models based on a neuro-fuzzy system and metaheuristic optimization algorithms for spatial prediction of wildfire probability". Agric. For. Meteorol. **2019**, 266, 198–207. https://doi.org/10.1016/j.agrformet.2018.12.015.

[107]. Moayedi, H.; Mehrabi, M.; Kalantar, B.; Abdullahi Mu'azu, M.; Rashid, A.S.A.; Foong, L.K.; Nguyen, H. "Novel hybrids of adaptive neuro-fuzzy inference system (ANFIS) with several metaheuristic algorithms for spatial susceptibility assessment of seismic-induced landslide". Geomat. Nat. Hazards Risk **2019**, 10, 1879–1911. https://doi.org/10.1080/19475705.2019.1650126.

[108]. Al-Fugara, A.k.; Ahmadlou, M.; Shatnawi, R.; AlAyyash, S.; Al-Adamat, R.; Al-Shabeeb, A.A.-R.; Soni, S. "Novel hybrid models combining meta-heuristic algorithms with support vector regression (SVR) for groundwater potential mapping". Geocarto Int. **2020**, 37, 2627–2646. https://doi.org/10.1080/10106049.2020.1831622.

[109]. Dodangeh, E.; Panahi, M.; Rezaie, F.; Lee, S.; Bui, D.T.; Lee, C.-W.; Pradhan, B. "Novel hybrid intelligence models for flood susceptibility prediction: Meta optimization of the GMDH and SVR models with the genetic algorithm and harmony search". J. Hydrol. **2020**, 590, 125423. DOI: 10.1016/j.jhydrol.2020.125423.

[110]. Haosen Chen, Lei Hou, Guomin (Kevin) Zhang, Sungkon Moon, "Development of BIM, IoT, and AR/VR technologies for fire safety and upskilling", Automation in Construction, Volume 125, **2021**, 103631, ISSN 0926-5805, https://doi.org/10.1016/j.autcon.2021.103631.

[111]. MINEMA, "National Contingency plan for fire incidents". March **2019**. http://minema.gov.rw/fileadmin/user_upload/MATRIX_PLAN_final.pdf.

[112]. Ignatius M., "Fire outbreaks on the increase again?" The New Times, 10 August **2016**.

[113]. [Online]. Available: https://www.newtimes.co.rw/section/read/202470. (Accessed on 25 September **2020**).

[114]. Dutta, J., Roy, S. & Chowdhury, C. "Unified framework for IoT and smartphone-based different smart city-related applications". Microsyst Technol **2019**, 25, 83–96. https://doi.org/10.1007/s00542-018-3936-9.

[115]. Wei-Ling H., Ji-Yun J., Chein-Shiun H., Chiu-Kuo L., and Yan-Chyuan S., "Application of Internet of Things in a Kitchen Fire Prevention System". Appl. Sci. **2019**, 9, 3520; DOI: 10.3390/app9173520.

[116]. Shilpa E., Hymavathi M., Tayyab Unnissa Begum, "Analysis of IoT-based Occupancy Monitoring Technique for Energy Efficient Smart Buildings". International Journal of Innovative Research in Computer, vol. IV, no. 10, pp. 2-3, **2016**. DOI: 10.1109/WCNCW.2015.7122529.

[117]. Anindya Dey; et al. "Namatad: Inferring occupancy from building sensors using machine learning". **2016** IEEE 3rd World Forum on Internet of Things (WF-IoT), 12-14 Dec. **2016**. DOI: 10.1109/WF-IoT.2016.7845462.

[118]. Bekiaris-Liberis N. and Ming J., "Sensing by proxy: Occupancy detection based on indoor CO_2". The 9th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM'15), July **2015**. Corpus ID: 1060467.

[119]. Prasad A., Wisam E., and Ramakrishna T., "Person Detection Techniques for an IoT Based Emergency". MOBIQUITOUS **2016**: Adjunct Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services November 2016 Pages 77–82. https://doi.org/10.1145/3004010.3004019.

[120]. B. S. Çiftler, S. Dikmese, İ. Güvenç, K. Akkaya, and A. Kadri, "Occupancy Counting with Burst and Intermittent Signals in Smart Buildings". In IEEE Internet of Things Journal, vol. 5, no. 2, pp. 724-735, April **2018**, DOI: 10.1109/JIOT.2017.2756689.

[121]. Prabakaran N., Kannan R.J., "Sustainable lifespan of WSN nodes using participatory devices in a pervasive environment". Microsyst. Technol. **2017**, 23, 651–657. https://doi.org/10.1007/s00542-016-3117-7.

[122]. Celik, T.; Demirel, H.; Ozkaramanli, H.; Uyguroglu, M. "Fire detection using the statistical color model in video sequences". J. Vis. Commun. Image Represent **2007**, 18, 176–185. DOI: 10.1016/j.jvcir.2006.12.003.

[123]. Liu, Z.; Kim, A.K. "Review of recent developments in fire detection technologies". J. Fire Prot. Eng. **2003**, 13, 129–151. https://doi.org/10.1177/1042391503013002003.

[124]. Dukuzumuremyi, J.P.; Zou, B.; Hanyurwimfura, D. "A novel algorithm for fire/smoke detection based on computer vision". Int. J. Hybrid Inf. Technol. **2014**, 7, 143–154. http://dx.doi.org/10.14257/ijhit.2014.7.3.15.

[125]. Rossi, L.; Akhloufi, M. "Dynamic Fire 3D Modeling Using a Real-Time Stereovision System". Springer: Dordrecht, the Netherlands, **2010**; pp. 33–38. **DOI**: 10.1007/978-90-481-3656-8_8.

[126]. Celik, T. "Fast and efficient method for fire detection using image processing". ETRI J. **2010**, 32, 881–890. https://doi.org/10.4218/etrij.10.0109.0695.

[127]. Stamford, C. "Analysts to Explore the Value and Impact of IoT on Business at Gartner". **2015**. Available online: https://www.gartner.com/en/newsroom/press-releases/2015-11-10-gartner-says-6-billion-connected-things-will-be-in-use-in-2016-up-30-percent-from-2015. (Accessed on 17 July 2020).

[128]. Ahmad, S.; Kim, D. "Design and implementation of thermal comfort system based on tasks allocation mechanism in smart homes". Sustainability **2019**, 11(20), 5849. https://doi.org/10.3390/su11205849.

[129]. Dhruba Gautam, Jony Mainaly, Regan Sapkota, Bimal Raj Regmi, Dinanath Bhandari, Popular Gentle; "National Disaster Risk Management Plan Climate Change and disaster risk and vulnerability context of Province 5". Report **2019**. https://www.opml.co.uk/files/Publications/a1594-strengthening-the-disaster-risk-response-in-nepal/climate-change-and-diaster-risk-and-vulnerability-context-province-5-final-1.pdf?noredirect=1. (Accessed on 30 May 2023).

[130]. Rwanda, Ministry of Disaster Management and Refugee Affairs, "A Guideline on Gender in Disaster Management"; **2019**; pp. 1–55.

[131]. Rwanda, MIDIMAR "National Disaster Risk Management Plan". **2013**. https://faolex.fao.org/docs/pdf/rwa168359.pdf, (Accessed on 30 May 2023).

[132]. Ghai, S.K.; Thanayankizil, L.V.; Seetharam, D.P.; Chakraborty, D. "Occupancy detection in commercial buildings using opportunistic context sources". In Proceedings of the **2012** IEEE International Conference on Pervasive Computing and Communications Workshops, Lugano, Switzerland, 19–23 May 2012. **DOI:** 10.1109/PerComW.2012.6197536.

[133]. Dey, A.; Ling, X.; Syed, A.; Zheng, Y.; Landowski, B.; Anderson, D.; Stuart, K.; Tolentino, M.E. "Namatad: Inferring occupancy from building sensors using machine learning". In Proceedings of the **2016** IEEE 3rd World Forum Internet Things, Reston, VA, USA, 12–14 December 2016. **DOI**:10.1109/WF-IoT.2016.7845462.

[134]. Sayed, Aya Nabil; Himeur, Yassine; Bensaali, Faycal "Deep and transfer learning for building occupancy detection: A review and comparative analysis"; Engineering Applications of Artificial Intelligence, Year **2022**, Volume 115. https://doi.org/10.1016/j.engappai.2022.105254.

[135]. Allahbakhshi, H.; Conrow, L.; Naimi, B.; Weibel, R. "Using Accelerometer and GPS Data for Real-Life Physical Activity Type Detection". Sensors 2020, 20, 588. https://doi.org/10.3390/s20030588.

[136]. Almanza, E.; Jerrett, M.; Dunton, G.; Seto, E.; Pentz, M.A. "A study of community design, greenness, and physical activity in children using satellite, GPS and accelerometer data". Heal. Place **2012**, 18, 46–54. https://doi.org/10.1016/j.healthplace.2011.09.003.

[137]. Jia, K.; Xiao, J.; Fan, S.; He, G. "An MQTT/MQTT-SN-Based User Energy Management System for Automated Residential Demand Response: Formal Verification and Cyber-Physical Performance Evaluation". Appl. Sci. **2018**, 8, 1035. https://doi.org/10.3390/app8071035.

[138]. Miller, H.J.; Tribby, C.P.; Brown, B.B.; Smith, K.R.; Werner, C.M.; Wolf, J.; Wilson, L.B.; Oliveira, M.G.S. "Public transit generates new physical activity: Evidence from individual GPS and accelerometer data before and after light rail construction in a neighborhood of Salt Lake City", Utah, USA. Heal. Place **2015**, 36, 8–17. https://doi.org/10.1016/j.healthplace.2015.08.005.

[139]. Grgić, K.; Špeh, I.; Heđi, I. "A Web-based IoT solution for monitoring data using MQTT protocol". In Proceedings of 2016 International Conference on Smart Systems and Technologies, Osijek, Croatia, 12–14 October **2016**; pp. 249–253. **DOI**: 10.1109/SST.2016.7765668.

[140]. Tang, K.; Wang, Y.; Liu, H.; Sheng, Y.; Wang, X.; Wei, Z. "Design and Implementation of Push Notification System Based on the MQTT Protocol". In Proceedings of the 2013 International Conference on Information Science and Computer Applications (ISCA 2013); Atlantis Press: Amsterdam, The Netherlands, **2013**, 116–119. **DOI**:10.2991/isca-13.2013.20.

[141]. Luzuriaga, J.E.; Cano, J.C.; Calafate, C.; Manzoni, P.; Perez, M.; Boronat, "Handling mobility in IoT applications using the MQTT protocol". In **2015** Internet Technologies and Applications, ITA 2015—Proceedings of the 6th International Conference; Institute of Electrical and Electronics Engineers (IEEE): New York City, NY, USA, 2015; pp. 245–250. **DOI**: 10.1109/ITechA.2015.7317403.

[142]. Barata, D.; Louzada, G.; Carreiro, A.; Damasceno, A. "System of Acquisition, Transmission, Storage and Visualization of Pulse Oximeter and ECG Data Using Android and MQTT". Procedia Technol. **2013**, 9, 1265–1272. https://doi.org/10.1016/j.protcy.2013.12.141.

[143]. Chooruang, K.; Mangkalakeeree, "Wireless Heart Rate Monitoring System Using MQTT". Procedia Comput. Sci. **2016**, 86, 160–163. https://doi.org/10.1016/j.procs.2016.05.045.

[144]. Lee, S.; Kim, H.; Hong, D.K.; Ju, H. "Correlation analysis of MQTT loss and delay according to QoS level". In International Conference on Information Networking; IEEE: New York City, NY, USA, **2013**; pp. 714–717. **DOI**: 10.1109/ICOIN.2013.6496715.

[145]. OASIS Standard Incorporating Approved Errata 01 | StandICT.eu. Available online: https://www.standict.eu/standards-watch/oasis-standard-incorporating-approved-errata-01. (Accessed on 20 July 2020).

[146]. Kannan Govindan; Amar Prakash Azad. "End-to-end service assurance in IoT MQTT-SN". In Proceedings of the **2015** 12th Annual IEEE Consumer Communications and Networking Conference, CCNC, Las Vegas, NV, USA, 9–12 January 2015; Volume 2015, pp. 290–296. **DOI**: 10.1109/CCNC.2015.7157991.

[147]. Vafeiadis, T.; Zikos, S.; Stavropoulos, G.; Ioannidis, D.; Krinidis, S.; Tzovaras, D.; Moustakas, K. "Machine Learning Based Occupancy Detection Via the Use of Smart Meters". **2017**. Available online: https://www.encompass-project.eu/wp-content/uploads/2017/10/ICESEE17_occupancy.pdf. (Accessed on 31 January 2021).

[148]. Cesana, M.; Redondi, A.; Longo, E.; Giardini, G. "Machine Learning Methods for Indoor Occupancy Detection with CO_2 Multi- Sensor Data". Available online: https://www.politesi.polimi.it/bitstream/10589/147394/3/Tesi_Giorgio_Giardini_874841.pdf. (Accessed on 21 July 2020).

[149]. Dinculeană, D.; Cheng, X. "Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices". Appl. Sci. **2019**, 9, 848. https://doi.org/10.3390/app9050848.

[150]. Sánchez, P.; Álvarez, B.; Antolinos, E.; Fernández, D.; Iborra, A. "A Teleo-reactive node for implementing internet of things systems". Sensors **2018**, 18, 1059. https://doi.org/10.3390/s18041059.

[151]. EMQ—Erlang MQTT Broker—EMQ 2.2—Erlang MQTT Broker 2.2-beta.1 Documentation. Available online: https://docs.emqx.io/broker/v2/en/index.html (Accessed on 20 July 2020).

[152]. Kaur, K.; Rani, R. "Modeling and querying data in NoSQL databases". In Proceedings—**2013** IEEE International Conference on Big Data, Big Data; IEEE: New York City, NY, USA, 2013, 2013, 1–7. **DOI**: 10.1109/BigData.2013.6691765.

[153]. Milan, A.; Rezatofighi, S.H.; Dick, A.; Reid, I.; Schindler, K. "Online Multi-Target Tracking Using Recurrent Neural Networks". In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA. 4–9 February **2017**. **DOI**: https://doi.org/10.1609/aaai.v31i1.11194.

[154]. Wang, Q.; Lin, J.; Yuan, Y. "Salient Band Selection for Hyperspectral Image Classification via Manifold Ranking". IEEE Trans. Neural Netw. Learn. Syst. **2016**, 27, 1279–1289. **DOI**: 10.1109/TNNLS.2015.2477537.

[155]. Graves, A.; Mohamed, A.; Hinton, G. "Speech Recognition with Deep Recurrent Neural Networks". Speech and Signal Processing; IEEE: New York City, NY, USA, **2013**; pp. 6645–6649. https://doi.org/10.48550/arXiv.1303.5778.

[156]. Mahata, S.K.; Das, D.; Bandyopadhyay, S. "MTIL2017: Machine Translation Using Recurrent Neural Network on Statistical Machine Translation". J. Intell. Syst. **2019**, 28, 447–453. https://doi.org/10.1515/jisys-2018-0016.

[157]. Vedantu Learn LIVE Online, NCERT-National Council of Educational Research and Training. Available online: https://www.vedantu. com/ncert-solutions/ncert-solutions-class-11-biology-chapter-17-breathing-and-exchange-of-gases. (Accessed on 8 June 2020).

[158]. How to Diagnose Overfitting and Underfitting of LSTM Models. Available online: https://machinelearningmastery.com/diagnose-overfitting-underfitting-lstm-models/ (Accessed on 30 September 2020).

[159]. Georgakopoulos, S.V.; Tasoulis, S.K.; Mallis, G.I.; Vrahatis, A.G.; Plagianakos, V.P.; Maglogiannis, I.G. "Change Detection and Convolution Neural Networks for fall Recognition". Neural Comput. Appl. **2020**, 32, 17245–17258. **DOI**:10.1007/s00521-020-05208-8.

[160]. Kalajdjieski, J.; Zdravevski, E.; Corizzo, R.; Lameski, P.; Kalajdziski, S.; Pires, I.M.; Garcia, N.M.; Trajkovik, V. "Air Pollution Prediction with Multi-modal Data and Deep Neural Networks". Remote Sens. **2020**, 12, 1–19. **DOI**:10.3390/rs12244142.

[161]. Dufour, J. Coefficients of Determination; McGill University: Montreal, QC, Canada, 2011. https://jeanmariedufour.github.io/ResE/Dufour_1983_R2_W.pdf. (Accessed on 26 July 2020).

[162]. Chai, T.; Draxler, R.R. "Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature". Geosci. Model Dev. **2014**, 7, 1247–1250. https://doi.org/10.5194/gmd-7-1247-2014.

[163]. RMSE: Root Mean Square Error—Statistics How To. Available online: https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error. (Accessed on 18 October 2020).

[164]. Bessa, R.J.; Trindade, A.; Silva, C.S.P.; Miranda, V. "Probabilistic solar power forecasting in smart grids using distributed information". Int. J. Electr. Power Energy Syst. **2015**, 72, 16–23. https://doi.org/10.1016/j.ijepes.2015.02.006.

[165]. Chen, Y.; Abraham, A.; Yang, J.; Yang, B. "Hybrid Methods for Stock Index Modeling". Available online: https://link.springer.com/ chapter/10.1007/11540007_137. (Accessed on 16 June 2020).

[166]. Lei, L. "Wavelet Neural Network Prediction Method of Stock Price Trend Based on Rough Set Attribute Reduction". Appl. Soft Comput. J. **2018**, 62, 923–932. https://doi.org/10.1016/j.asoc.2017.09.029.

[167]. Enke, D.; Mehdiyev, N. "Stock Market Prediction Using a Combination of Stepwise Regression Analysis, Differential Evolution-based Fuzzy Clustering, and a Fuzzy Inference Neural Network". Intell. Autom. Soft Comput. **2013**, 19, 636–648. **DOI**:10.1080/10798587.2013.839287.

[168]. Akay, B.; Ragni, D.; Ferreira, C.S.; van Bussel, G.J.W. "Investigation of the root flow in a Horizontal Axis". Wind Energy **2013**, 2016, 1–20. https://doi.org/10.1002/we.1620.

[169]. Corizzo, R.; Ceci, M.; Fanaee-T, H.; Gama, J. "Multi-aspect renewable energy forecasting". Inf. Sci. **2021**, 546, 701–722. DOI: 10.1016/j.ins.2020.08.003.

[170]. Ceci, M.; Corizzo, R.; Japkowicz, N.; Mignone, P.; Pio, G. "ECHAD: Embedding-Based Change Detection from Multivariate Time Series in Smart Grids. IEEE Access **2020**, 8, 156053–156066. **DOI**: 10.1109/ACCESS.2020.3019095.

[171]. Benazzouz, Y.; Munilla, C.; Gunalp, O.; Gallissot, M.; Gurgen, L. "Sharing user IoT Devices in the Cloud". In Proceedings of the IEEE World Forum on Internet of Things, WF-IoT 2014, Seoul, Korea, 6–8 March **2014**; pp. 373–374. https://doi.org/10.1109/wf-iot.2014.6803193.

[172]. Macías, A.; Navarro, E.; González, P. "A Microservice-Based Framework for Developing Internet of Things and People Applications". Multidiscip. Digit. Publ. Inst. Proc. **2019**, 31, 85. https://doi.org/10.3390/proceedings2019031085.

[173]. Reason, J.; Hobbs, A. "Managing Maintenance Error: A Practical Guide"; CRC Press: Boca Raton, FL, USA, 2003; ISBN 9780754615910.

[174]. Caballero, A. "Information Security Essentials for Information Technology Managers: Protecting Mission-Critical Systems". In Computer and Information Security Handbook; Morgan Kaufmann: Burlington, MA, USA, **2013**; pp. 1–45. https://doi.org/10.1016/b978-0-12-416688-2.00001-5.

[175]. Judvaitis, J.; Nesenbergs, K.; Balass, R.; Greitans, M. "Challenges of DevOps ready IoT Testbed". In Proceedings of the ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems, Munich, Germany, 15–20 September **2019**. http://ceur-ws.org/Vol-2442/paper1.pdf. (Accessed on 18 October 2020).

[176]. "DevOps—The Web's Largest Collection of DevOps Content". Available online: https://devops.com/ (Accessed on 16 July 2021).

[177]. Muller, G. "Systems Engineering Research Methods". Procedia Comput. Sci. **2013**, 16, 1092–1101. https://doi.org/10.1016/j.procs.2013.01.115.

[178]. Farroha, B.S.; Farroha, D.L. "A framework for managing mission needs, compliance, and trust in the DevOps environment". In Proceedings of the **2014** IEEE Military Communications Conference, Washington, DC, USA, 6–8 October 2014; pp. 288–293. https://doi.org/10.1109/MILCOM.2014.54.

[179]. Casale, G.; Chesta, C.; Deussen, P.; Di Nitto, E.; Gouvas, P.; Koussouris, S.; Stankovski, V.; Symeonidis, A.; Vlassiou, V.; Zafeiropoulos, A.; et al. "Current and Future Challenges of Software Engineering for Services and Applications". Procedia Comput. Sci. 2016, 97, 34–42. https://doi.org/10.1016/j.procs.2016.08.278.

[180]. Lwakatare, L.E.; Kuvaja, P.; Oivo, M. "Relationship of DevOps to Agile, Lean and Continuous Deployment". In Proceedings of the International Conference on Product-Focused Software Process Improvement, Trondheim, Norway, 22–24 November **2016**; pp. 399–415. https://doi.org/10.1007/978-3-319-49094-6_27.

[181]. Lwakatare, L.E.; Kuvaja, P.; Oivo, M. "An Exploratory Study of DevOps: Extending the Dimensions of DevOps with Practices". ICSEA **2016**, Rome, 2016. Available online: http://n4s.dimecc.com/publication/an-exploratory-study-of-devops-extending-the-dimensions-of-devops-with-practices/ (Accessed on 17 July 2021).

[182]. Dyck, A.; Penners, R.; Lichter, H. "Towards Definitions for Release Engineering and DevOps". In Proceedings of the **2015** IEEE/ACM 3rd International Workshop on Release Engineering, Florence, Italy, 19 May 2015; p. 3. https://doi.org/10.1109/releng.2015.10.

[183]. Ali, S.; Jarwar, M.A.; Chong, I. "Design Methodology of Microservices to Support Predictive Analytics for IoT Applications". Sensors **2018**, 18, 4226. https://doi.org/10.3390/s18124226.

[184]. Moore, J.; Kortuem, G.; Smith, A.; Chowdhury, N.; Cavero, J.; Gooch, D. "DevOps for the Urban IoT". In Proceedings of the Second International Conference on IoT in Urban

Space, Tokyo, Japan, 24–25 May **2016**; pp. 78–81. https://doi.org/10.1145/2962735.2962747.

[185]. Gokceli, S.; Zhmurov, N.; Kurt, G.K.; Ors, B. "IoT in Action: Design and Implementation of a Building Evacuation Service". J. Comput. Networks Commun. **2017**, 2017, 8595404. https://doi.org/10.1155/2017/8595404.

[186]. Truong, H.-L.; Klein, P. "DevOps Contract for Assuring Execution of IoT Microservices in the Edge". Internet Things **2019**, 9, 100150. https://doi.org/10.1016/j.iot.2019.100150.

[187]. Abbott, M.; Fisher, M. "The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise"; Addison-Wesley Prof.: San Francisco, CA, USA, **2009**; ISBN-10: 0134032802; ISBN-13: 978-0134032801.

[188]. Vresk, T.; Cavrak, I. "Architecture of an Interoperable IoT Platform based on Microservices". In Proceedings of the 2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 30 May–3 June **2016**; pp. 1196–1201. https://doi.org/10.1109/mipro.2016.7522321.

[189]. Villamizar, M.; Garces, O.; Castro, H.; Verano, M.; Salamanca, L.; Casallas, R.; Gil, S. "Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud". In Proceedings of the **2015** 10th Computing Colombian Conference (10CCC), Bogota, Colombia, 21–25 September **2015**; pp. 583–590. https://doi.org/10.1109/columbiancc.2015.7333476.

[190]. Ali, S.; Kibria, M.G.; Jarwar, M.A.; Lee, H.K.; Chong, I. "A Model of Socially Connected Web Objects for IoT Applications". Wirel. Commun. Mob. Comput. **2018**, 2018, 6309509. https://doi.org/10.1155/2018/6309509.

[191]. Akasiadis, C.; Pitsilis, V.; Spyropoulos, C.D. "A Multi-Protocol IoT Platform Based on Open-Source Frameworks". Sensors **2019**, 19, 4217. https://doi.org/10.3390/s19194217.

[192]. Salikhov, D.; Khanda, K.; Gusmanov, K.; Mazzara, M.; Mavridis, N. "Microservice-based IoT for smart buildings". In Proceedings of the **2017** 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), Taipei, Taiwan, 27–29 March 2017; pp. 302–308. https://doi.org/10.1109/WAINA.2017.77.

[193]. Morabito, R.; Farris, I.; Iera, A.; Taleb, T. "Evaluating Performance of Containerized IoT Services for Clustered Devices at the Network Edge". IEEE Internet Things J. **2017**, 4, 1019–1030. https://doi.org/10.1109/jiot.2017.2714638.

[194]. De Prado, R.P.; García-Galán, S.; Muñoz-Expósito, J.E.; Marchewka, A.; Ruiz-Reyes, N. "Smart Containers Schedulers for Microservices Provision in Cloud-Fog-IoT Networks. Challenges and Opportunities". Sensors **2020**, 20, 1714. https://doi.org/10.3390/s20061714.

[195]. Rodriguez, M.A.; Buyya, R. "Container-based cluster orchestration systems: A taxonomy and future directions". Softw. Pract. Exp. **2018**, 49, 698–719. https://doi.org/10.1002/spe.2660.

[196]. Babar, M.A. "Understanding Container Isolation Mechanisms for Building Security-Sensitive Private Cloud"; Technical Report; CREST—Centre for Research on Engineering Software Technologies, University of Adelaide: Australia. Available online: https://www.researchgate.net/publication/316602321_Understanding_Container_Isolation_Mechanisms_for_Building_Security-Sensitive_Private_Cloud (Accessed on 28 November 2021).

[197]. Docker. Swarm Mode Overview|Docker Documentation. Available online: https://docs.docker.com/engine/swarm/ (Accessed on 5 December 2021).

[198]. Smith, B. ARM and Intel Battle over the Mobile Chip's Future. Computer **2008**, 41, 15–18. https://doi.org/10.1109/mc.2008.142.

[199]. Petrolo, R.; Morabito, R.; Loscrì, V.; Mitton, N. "The design of the gateway for the Cloud of Things". Ann. Telecommun. **2016**, 72, 31–40. https://doi.org/10.1007/s12243-016-0521-z.

[200]. ARM, A.L. Arch Linux ARM. **2018**. Available online: https://archlinuxarm.org/ (Accessed on 22 October 2021).

[201]. Karl, H.; Willig, A. "Protocols and Architectures for Wireless Sensor Networks"; John Wiley & Sons: Hoboken, NJ, USA, **2005**. https://doi.org/10.1002/0470095121.

[202]. Singh, S.K.; Singh, M.; Singh, D.K. "Routing Protocols in Wireless Sensor Networks—A Survey". Int. J. Comput. Sci. Eng. Surv. **2010**, 1, 63–83. https://doi.org/10.5121/ijcses.2010.1206.

[203]. Device Authority, Docker + Device Authority's KeyScaler Platform. **2019**. Available online: https://www.deviceauthority.com/wp-content/uploads/2021/09/solutionbrief_docker_keyscaler.pdf (accessed on 27 April 2022).

[204]. Felter, W.; Ferreira, A.; Rajamony, R.; Rubio, J. "An Updated Performance Comparison of Virtual Machines and Linux Containers". In Proceedings of the 2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Philadelphia, PA, USA, 29–31 March **2015**; pp. 171–172. https://doi.org/10.1109/ispass.2015.7095802.

[205]. Liu, D.; Zhao, L. "The Research and Implementation of Cloud Computing Platform Based on Docker". In Proceedings of the 2014 11th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 19–21 December **2014**; pp. 475–478. https://doi.org/10.1109/iccwamtip.2014.7073453.

[206]. Ismail, B.I.; Goortani, E.M.; Ab Karim, M.B.; Tat, W.M.; Setapa, S.; Luke, J.Y.; Hoe, O.H. "Evaluation of Docker as Edge Computing Platform". In Proceedings of the 2015 IEEE Conference on Open Systems (ICOS), Melaka, Malaysia, 24–26 August **2015**; pp. 130–135. https://doi.org/10.1109/icos.2015.7377291.

[207]. Morabito, R.; Petrolo, R.; Loscri, V.; Mitton, N. "Enabling a Lightweight Edge Gateway-as-a-Service for the Internet of Things". In Proceedings of the **2016** 7th International Conference on the Network of the Future (NOF), Búzios, Brazil, 16–18 November 2016; pp. 1–5. https://doi.org/10.1109/nof.2016.7810110.

[208]. Taherizadeh, S.; Jones, A.C.; Taylor, I.; Zhao, Z.; Stankovski, V. "Monitoring self-adaptive applications within edge computing frameworks: A state-of-the-art review". J. Syst. Softw. **2018**, 136, 19–38. https://doi.org/10.1016/j.jss.2017.10.033.

[209]. Seo, K.-T.; Hwang, H.-S.; Moon, I.-Y.; Kwon, O.-Y.; Kim, B.-J. "Performance Comparison Analysis of Linux Container and Virtual Machine for Building Cloud". Adv. Sci. Technol. Lett. **2014**, 66, 2. https://doi.org/10.14257/astl.2014.66.25.

[210]. Atzori, L.; Iera, A.; Morabito, G. "The Internet of Things: A survey". Comput. Netw. **2010**, 54, 2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010.

[211]. S. K. Mekni, "Design and Implementation of a Smart fire detection and monitoring system based on IoT," 29–31 March **2022** 4th International Conference on Applied Automation and Industrial Diagnostics (ICAAID), Hail, Saudi Arabia, 2022, pp. 1-5, **DOI**: 10.1109/ICAAID51067.2022.9799505.

[212]. S. Kabir, P. Gope and S. P. Mohanty, "A Security-Enabled Safety Assurance Framework for IoT-Based Smart Homes," in IEEE Transactions on Industry Applications, vol. 59, no. 1, pp. 6-14, Jan.-Feb. **2023**, **DOI**: 10.1109/TIA.2022.3176257.

LIST OF AUTHOR'S PUBLICATIONS

Papers submitted to international journals.

1. **Hitimana, E**.; Bajpai, G.; Musabe, R.; Sibomana, L.; Kayalvizhi, J. "Implementation of IoT Framework with Data Analysis Using Deep Learning Methods for Occupancy Prediction in a Building". *Future Internet* 2021, *13*, 67. https://doi.org/10.3390/fi13030067.

2. **Hitimana, E**.; Bajpai, G.; Musabe, R.; Sibomana, L.; Kayalvizhi, J. "Containerized Architecture Performance Analysis for IoT Framework Based on Enhanced Fire Prevention Case Study: Rwanda". Sensors 2022, 22(17), 6462 https://doi.org/10.3390/s22176462.

3. **Hitimana, E**., Bajpai, G., Musabe, R., Sibomana, L., Kayalvizhi, J. "IoT Application for Spruce Fire Detection in Rwanda". In: Fong, S., Dey, N., Joshi, A. (eds) ICT Analysis and Applications. Lecture Notes in Networks and Systems 2022, vol 314. Springer, Singapore. https://doi.org/10.1007/978-981-16-5655-2_79.

APPENDIX

Figure 5.11: (a) Visualization of container clusters with one non-working container service with the attached RPi worker. (b) Visualization after system container stabilization.

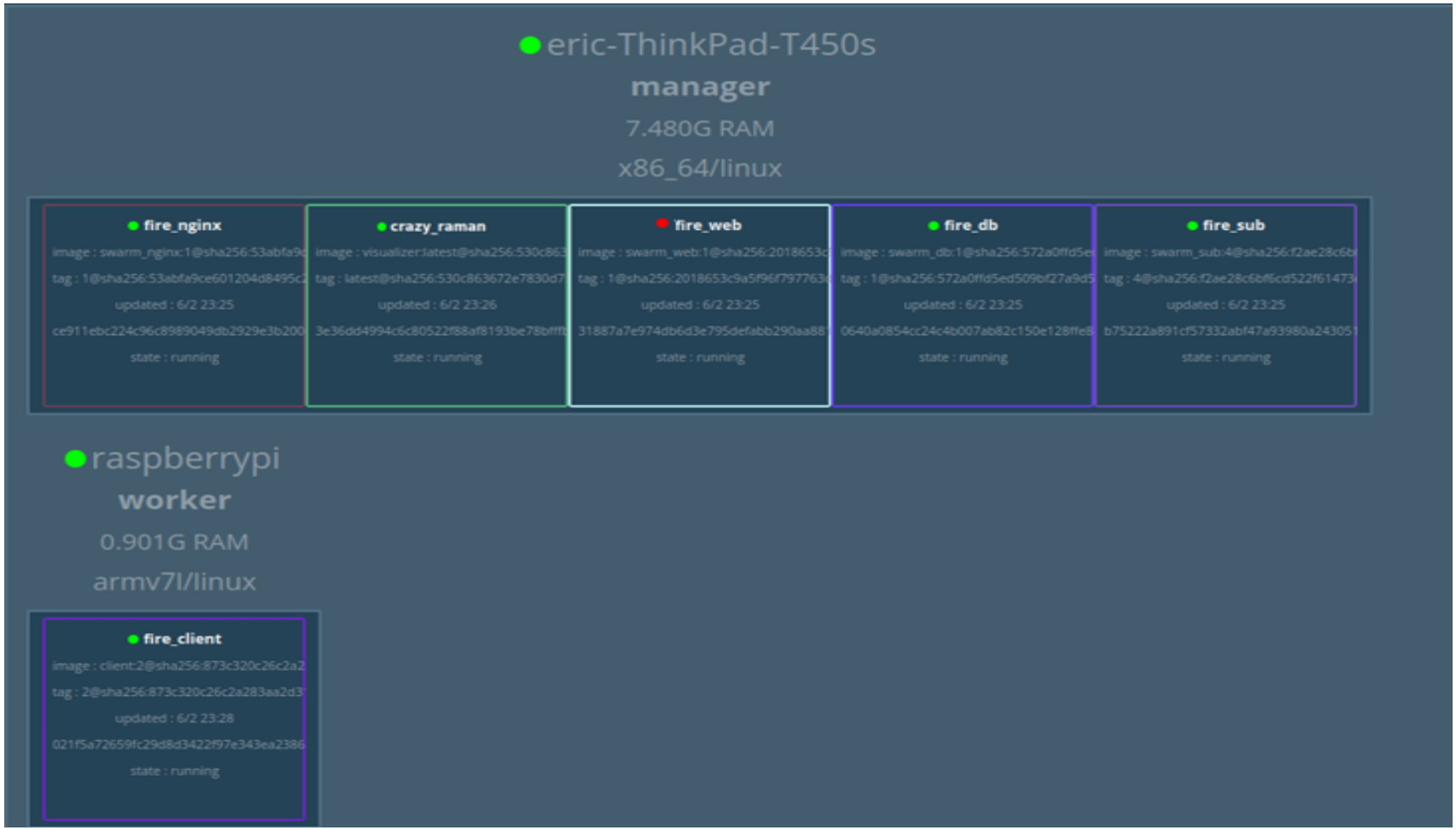


Figure 5.12: Docker Swarm with duplicated container services at the scale up purpose