An IoT Prototype to Mitigate Human-Wildlife Conflict

Thuo Ng'ang'a Thuo

094518

Dissertation Submitted in Partial Fulfillment of the Requirements for the Award of the Degree of Master of Science in Information Technology at Strathmore University

Faculty of Information Technology

Strathmore University

Nairobi, Kenya

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the dissertation contains no material previously published or written by another person except where due reference is made in the dissertation itself.

THUO NG'ANG'A THUO

Student Number: 094518

Signature 

Date 12th July 2021

Approval

The proposal of Thuo Ng'ang'a Thuo was reviewed and approved by:

Dr. Vitalis Ozianyi (Ph.D.),
Senior Lecturer,
Faculty of Information Technology ,
Strathmore University.

Dr. Julius Butime (Ph.D.) Dean,
Faculty of Information Technology,
Strathmore University.

Dr. Bernard Shibwabo (Ph.D.),
Director of Graduate Studies,
Strathmore University.

Acknowledgements

First and foremost, I thank God who has enabled me to successfully undertake and complete this dissertation. I would also like to especially thank my supervisor, Dr. Vitalis Ozianyi for his astute guidance throughout the undertaking of this dissertation, my family for their reassuring support without which I would not have completed this research, Arthur Kennedy Otieno for his invaluable insights while building the prototype, my late grandmother, Loise Wanjiru, who always encouraged my efforts, my mother, father and siblings for their sacrifice in facilitating my studies and all those who indirectly supported my academic endeavours whose impact, however small, went a long way.

Abstract

Due to the accelerated population growth in Kenya and around the world, pressure on and competition for resources in game parks, ranches, conservancies and game reserves has grown tremendously. This has led to the reduction of wildlife in terms of population due to poaching and territory invasions by both domestic animals and human beings. This problem calls for a technological solution to mitigate the constant conflict between wild animals and human beings. The eventual fallout from this conflict, if it were allowed to persist, would be the extinction of important flora and fauna that contributes a great deal to natural ecosystems vital for the survival of the human race, the habitability of the planet and the sustenance of economies which gain from the existence of unique wild animals and plants through tourism and scientific research into wildlife. The main objective of this research is to mitigate human-wildlife conflict by developing a prototype that combines web, IoT and SMS technologies. In addition, this research also applies the Rapid Application Development methodology to develop and test a sensor-based system for monitoring animal activity in human-wildlife conflict prone areas through Internet of Things. Its physical architecture consists of an Arduino microcontroller, a Wi-Fi shield a breadboard and a motion sensor. In this architecture, the Arduino microcontroller powers the Wi-Fi shield which then connects to an available access point and in so doing, the internet. The motion sensor detects motion data and sends it to the Wi-Fi shield. The logic for this process is written and compiled into the Arduino microcontroller and Wi-Fi shield via a C++ program. The data is sent via the HTTP protocol to an AWS lambda function written in python processes this data. Once the data is processed, an SMS is triggered and sent to registered phone numbers to warn them of imminent conflict and advise on steps to take. The SMS message also helps the recipients to plan better and deploy resources in a more organised fashion to areas where conflict is rife. This solution is also low cost, accurate and can be implemented at scale along boundary areas. The results in this research show that it is possible to combine web, SMS and IoT technologies to successfully mitigate and reduce human-wildlife conflict.

Table of Contents

List of Figures

List of Tables

List of Acronyms

API – Application Programming Interface

AWS – Amazon Web Services

EC2 – Elastic Compute Cloud

GNU – GNU's Not Unix

GPS – Global Positioning System

HTTPS – Hypertext Transfer Protocol Secure

IOT – Internet of Things

JSON – JavaScript Object Notation

LED – Light Emitting Diode

LTE – Long Term Evolution

PIR – Passive Infrared Sensor

UDP – Unified Datagram Protocol

VHF – Very High Frequency

SMS – Short Message Service

S3 – Simple Storage Service

SSH – Secure Shell

Definition of Terms

Access Point: The main source of Wi-Fi signals in a particular area (Lee & Labinghisa, 2019).

Arduino: This is an open source programmable board that is easy where one can write and compile programs known as sketches onto the board (Kaswan, Singh, & Sagar, 2020).

Internet of Things: The Internet of Things is an emerging paradigm that enables communication between electronic devices and sensors through the internet and facilitates innovative solutions to various challenges in multiple fields and disciplines (Kumar, Tiwari, & Zymbler, 2019).

Application Programming Interface (API): API's provide programmatic access to the features of a framework, operating system or service within predetermined constraints (Ren, Sun, & Xing, 2020)

Secure Shell: This is an encryption protocol based on public/private key pairs for authentication that is used to manage most Linux servers and cloud computing infrastructure in general (Ylonen, 2019)

Chapter One: Introduction

1.1: Background

According to Habib, Nazir, Fazili and Bhat (2015), human-wildlife conflict is the interaction between wild animals and people which has a negative impact on people and their resources and wild animals and their habitat. In Africa, where large mammals such as elephants, rhinos and lions roam in animal reserves, human wildlife conflict is prevalent (Makindi, Mutinda, Olekaikai, Olelebo, & Aboud, 2014). Mishra, Rathore and Pandey (2014) further write that the impact of this conflict is the loss of crops, livestock, property and human lives whereas for wild animals, many of which are endangered, it is retaliatory attacks from humans that they face, where they are often killed.

According to Lewa, Maluki, Vindevov and Farah (2017), human-wildlife conflict is caused by increasing human population encroaching into and near protected areas and wildflife habitats resulting in competition between wildlife and human beings for natural recources.They further add that proximity to forest boundaries, where communities living closest to forests are affected most by human-wildlife conflict and land use transformation, also greatly contributes to the occurrence of human-wildlife conflict

By leveraging the Internet of Things, a solution to this conflict can be found. According to the Internet Society (2015), "The term 'Internet of Things' was first used in 1999 by British technology pioneer Kevin Ashton to describe a system in which objects in the physical world could be connected to the Internet by sensors." According to GSMA (2014), IoT refers to the use of intelligently connected devices and systems to leverage data gathered by embedded sensors and actuators in machines and other physical objects. IoT is expected to spread rapidly over the coming years and this convergence will unleash a new dimension of services that improve the quality of life of consumers and productivity of enterprises, unlocking an opportunity that the GSMA refers to as the 'Connected Life'. According to GSMA (2020), in enterprises, IoT enables new business models that create value by connecting both existing and new devices to create new business processes and subsequently reduce costs, increase business efficiency, enable greater innovation and drive improved visibility in an organisation. This can therefore be leveraged to solve the problem of human-wildlife conflict technologically (WWF, 2018).

Technological attempts to solve human-wildlife have been made with limited success such as the use of drones, GPS tracking, habitat monitoring, L.E.D lights and camera traps. The efficiency of IoT has also been put to use such as in the case of acoustic sensor networks to distinguish animal and human activity through audio signatures (Hodgkinson & Young, 2015). This research intends to fill the gap that exists in communicating action items to rangers about wildlife motion activity near fences and boundaries of conflict prone areas and visualising animal activity in graphical format.

This research focuses on mitigating human-wildlife conflict by combining sensor technologies and web-based technologies to alert rangers of potential conflict via SMS and provide a metrics dashboard for real time visualisation of PIR sensor information. Employees and rangers will receive a message on their phones with information about potential occurrence of conflict within the vicinity of the location of the motion sensors with a call to action giving them instructions on what to do as a result. When motion, in this case, an animal approaching a fence, occurs, a PIR sensor collects that data, sends it over the internet to a database via the UDP protocol and a web application will then scrape the data from the database and visualise it in graphical format. The web application will also have additional functionality to trigger an SMS to send to game rangers whenever unusual spikes in motion data occur.

1.2 Problem statement

The existing land intended for human use is periodically invaded by wildlife causing destruction of property, food, livestock and life. This in turn leads to retaliatory attacks by people on wild animals where people hunt and kill them to ease the frequency of attacks. On the other hand, human beings have encroached on land reserved for wildlife due to the unavailability of land because of the rising human population. Human beings therefore have a need to know when an animal is within their vicinity in order to give them time to prepare and act appropriately and in time to salvage their livestock and property as well as avert potentially deadly conflict.

In this research, a gap was identified along perimeter areas of game parks, reserves and forested areas with a large animal population where rangers have no way of knowing how or when they might encounter an animal close to them or even when an animal leaves its designated area. The existing technological solutions currently in use such as GPS, satellite and radio are expensive and sometimes inaccurate and ineffective due to the limited communications infrastructure which

many areas lack (Liu, Yang, & Yan, 2015). The goal of this research is to reduce this conflict by leveraging the low cost and accuracy of IoT devices and the ubiquitous SMS technologies.

1.3 Objectives

1.3.1 General objective

The general objective of this study is to mitigate human-wildlife conflict by developing a prototype that combines IoT, web and SMS technologies.

1.3.2 Specific objectives

i. To evaluate the existing technological methods that address human-wildlife conflict,
ii. To review the current challenges of IoT based wildlife monitoring,
iii. To develop a sensor-based prototype that remotely detects animal movement, sends the detected metrics via the internet, visualizes the detected metrics on a web application and sends alerts via SMS,
iv. To test the sensor-based IoT prototype.

1.4: Research Questions

i. What are the existing technological methods used to prevent human-wildlife conflict?
ii. Where are the current challenges in existing technological methods of mitigation?
iii. How can the prototype be designed and developed by combining IoT, web and SMS technologies?
iv. How will the prototype be tested and implemented?

1.5: Justification

The sensor-based application is low cost and easy to use. It also leverages well known existing technologies such as SMS to deliver vital information that could help ease run-ins between people and wildlife to authorities who will take more specific action to deploy countermeasures such as rangers to conflict prone areas. The prototype also provides a graphical visualisation of motion sensor data which provides insight into the amount of animal motion along boundary areas.

1.6: Scope Limitation

This research deals with developing a sensor-based prototype to detect animal motion along perimeters by leveraging IoT, web and SMS technologies. The study is limited to a graphical visualisation of motion sensor data and SMS alerts. Due to resource constraints, the final product covered the primary functionality of the system.

Chapter Two: Literature Review

2.1: Introduction

In this chapter, we explore the components of the IoT architecture and the existing applications in human-wildlife conflict mitigation, the protocols that have been used in implementing these solutions and the methods used in interconnecting IoT and web technologies. This chapter ends with a representation of the solution via a conceptual model showing the physical and virtual components.

2.2: Components of an IoT Architecture

The key components of an IoT solution comprise of five layers which are the physical device layer where physical devices are placed remotely in the field. They also have a transmission layer which captures the data from the devices and transmits it to an IoT platform layer which captures the data. A data storage layer then stores and processes metrics stored from the platform layer and the metrics are presented to users via an application layer (Hodgkinson & Young, 2015). The layers are illustrated in figure 2.1.

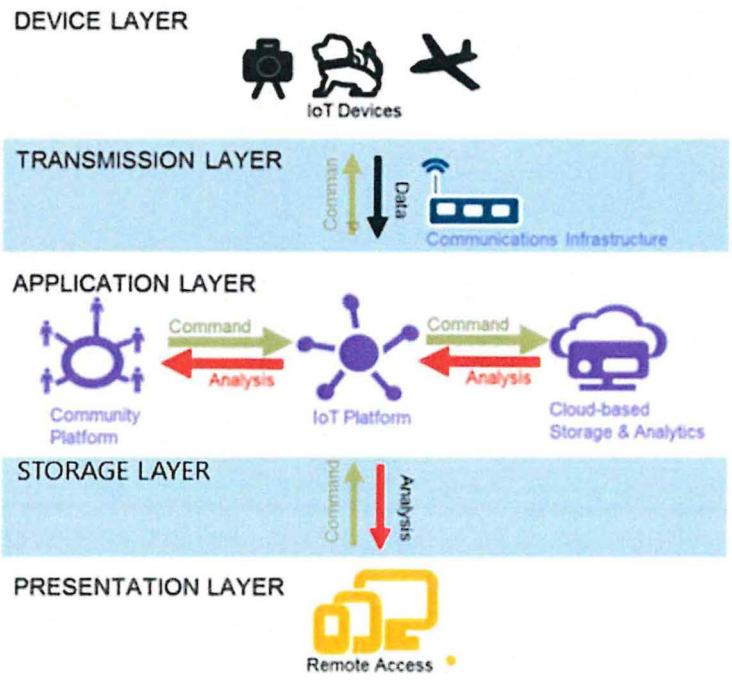


Figure 2.1: IoT architecture (Hodgkinson & Young, 2015)

Hodgkinson and Young (2015), further write that the hardware in the device layer includes sensors, drones, cameras and trackers that capture data from or interact with the environment. In the

transmission layer, a network needs to exist to transmit data from the physical devices; implementations of network infrastructure to transmit this data include Very High Frequency radio, 4G networks or Long-Term Evolution, satellite networks such as Iridium and Automatic Packet Reporting System. The platform layer controls the behaviour of each IoT device and provides automatic collection and processing of data for reporting. The storage layer receives and stores data from the devices in the field and is able to inspect, clean, transform and model with the goal of discovering useful information, suggesting conclusions and supporting decision making. The presentation layer is accessed via a configurable computing resource such as a cloud server. This resource can be rapidly configured for use by potential users and stakeholders from the field of wildlife management and operations such as rangers, ecologists, local community members and NGO's.

2.2.1: IoT Communication models

There exist various methods in which IoT devices communicate with each other over a network. This section aims to describe these methods in brief. According to Kulkarni and Kulkarni (2017), there are four main communication models which are device-to-device communication, device-to-cloud communication, device-to-gateway model and a back-end-data-sharing model.

In device-to-device communications, two or more devices directly connect and communicate between one another, rather than through an intermediary application server. These devices can use many types of networks including IP and bluetooth (Kulkarni & Kulkarni, 2017). Figure 2.3 illustrates how the model works:

Figure 2.2: Device-to-device model (The Internet Society, 2015)

According to Kulkarni and Kulkarni (2017), "In a device-to-cloud communications, the IoT device connects directly to an Internet cloud service like an application service provider to exchange data and control message traffic. This approach frequently takes advantage of existing communications networks like traditional wired Ethernet or Wi-Fi connections to establish a connection between the device and the IP network, which ultimately connects to the cloud service." Figure 2.4 illustrates the device-to-cloud model.

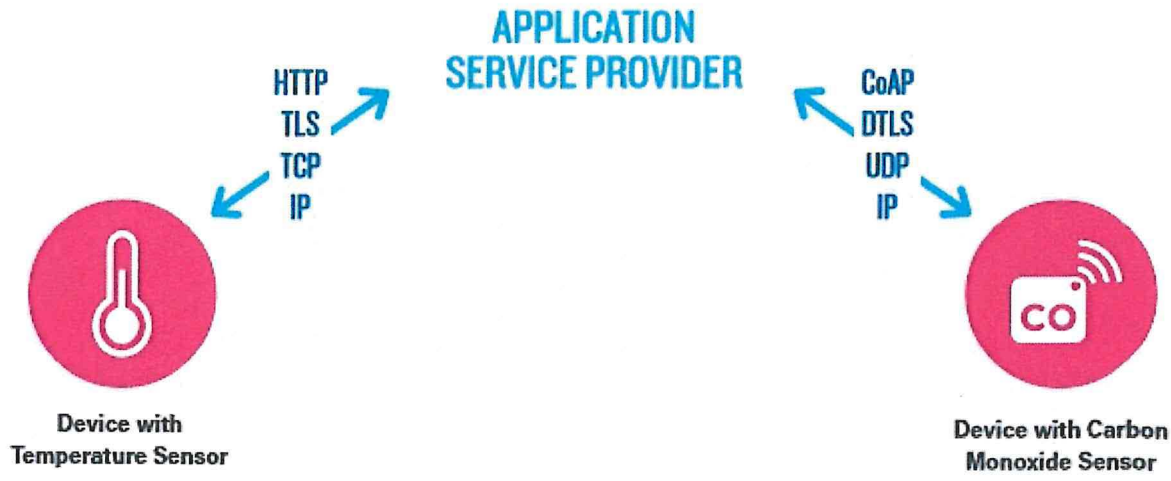


Figure 2.3: Device-to-cloud model (The Internet Society, 2015)

As Kulkarni and Kulkarni (2017) write, in the device-to-gateway model or the device-to-application-layer-gateway (ALG), an IoT device connects using an ALG service as a conduit to reach a cloud service. The functionality is illustrated in figure 2.5.

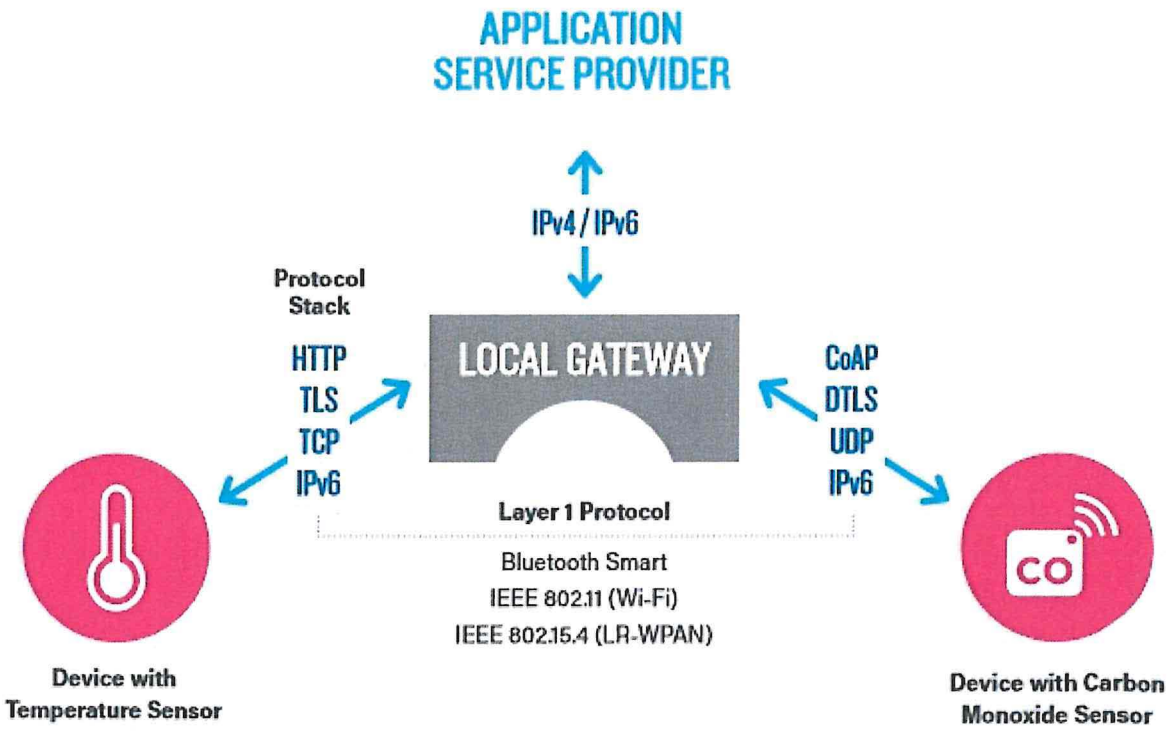


Figure 2.4: Device-to-gateway model (The Internet Society, 2015)

Finally, there is the back-end-data-sharing model that enables users to export and analyse smart object data from a cloud service in combination with data from other sources. This model is an extension of the single device-to-cloud model where data collected from a single IoT device is aggregated and analysed (Kulkarni & Kulkarni, 2017). Figure 2.6 describes the workings of this model:

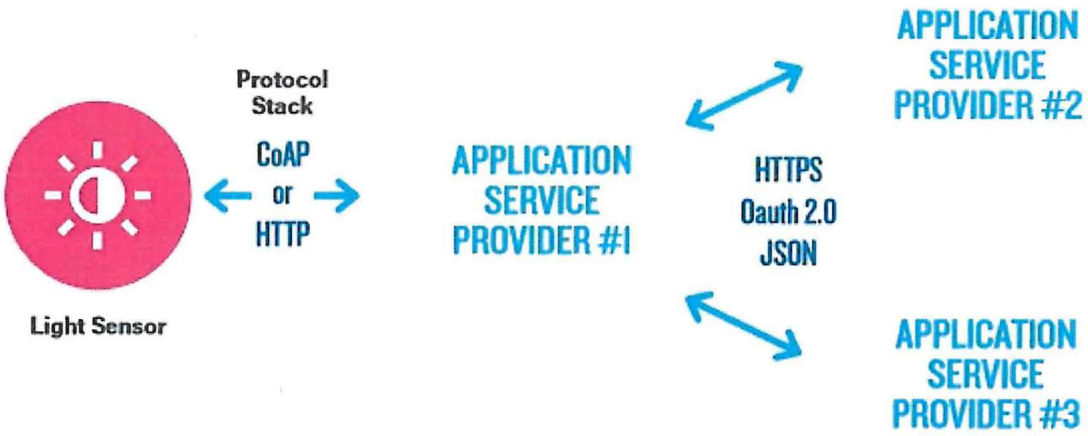


Figure 2.5: Back-end-data-sharing model (The Internet Society, 2015)

2.2.2: General architecture of IoT

In general, most implementations of IoT solutions usually follow a general architecture that consists of a gateway that collects and processes metrics from the physical components, a remote server that contains programs and databases to process and store the metrics from the gateway and an interface that displays analytics generated by the activity from the physical layer. Figure 2.2 illustrates how data is transformed and interacts with the components of the IoT architecture (Kumar, Tiwari, & Zymbler, 2019).

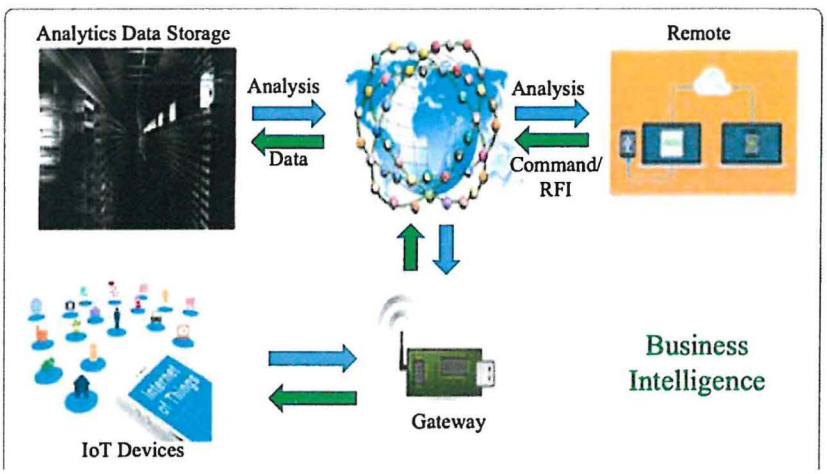


Figure 2.6: General IoT Architecture (Kumar, Tiwari, & Zymbler, 2019)

2.3: IoT Technologies in use to Monitor Wildlife

According to Lopez Research (2013), The Internet of Things is defined as "a system where items in the physical world, and sensors within or attached to these items, are connected to the Internet via wireless and wired Internet connections. These sensors can use various types of local area connections such as RFID, NFC, Wi-Fi, Bluetooth, and Zigbee. Sensors can also have wide area connectivity such as GSM, GPRS, 3G, and LTE." Currently, there are implementations of IOT in various parts of the world which utilize IoT to mitigate human conflict. These implementations use the TCP/IP networking suite as well as radio communication to receive and transmit data which will be discussed in subsequent chapters.

In Sri Lanka, there is a prevalent human elephant conflict and one of the innovations to deal with this conflict is the use of sensing and data processing (Sayakkara, et al., 2015). The University of Colombo's School of Computing attempted to make electric fences smarter and improve elephant warning systems by developing a cost-effective electric fence with small IOT nodes placed along the wire (Sayakkara, et al., 2015).

Sayakkara, et al. (2015), further write that packets are encoded into the high-voltage electric pulses in a way that enables one to identify which node is disconnected from the network. When a node is disconnected from the network, it implies that part of the fence is broken and alerts are sent to maintenance crews with the exact location of the breakage. This solution aims at informing villagers and the researchers on fence breakages and intrusions.

In India, a low-cost alert system has been made to detect the movement of wildlife outside the forest by using motion detectors and a single board computer that relays data to a control centre. In the solution, speakers are used to make loud noises to scare them into staying inside the forests to prevent disruption to normal human life (Shivaram, Chaitra, Kshama, Sneha, & Supriya, 2016)

As Shivaram, Chaitra, Kshama, Sneha and Supriya (2016) write, their solution comprises of PIR sensors and a Raspberry Pi module acting as a computer and sensor tower to process all the incoming signals from the PIR sensors which detect motion. If a PIR sensor is trigerred, an SMS

and a photograph is sent to an official or a loud noise is trigerred to scare wild animals away and prevent them from crossing the perimeter

A solution to detect wildlife crossing roads to prevent collisions has also been developed. As Viani, et al. (2013) write, the system is based on a wireless sensor network architecture deployed along the road sides in order to alert drivers in real time about the presence of deer within a predetermined area. It is a solution for vehicle-wildlife conflict which is low cost, feasible and highly applicable in urban areas where animals may from time to time cross roads. Sensors are installed along road markers on the road sides to detect movement of approaching deer and relay the information through a wireless network to a control unit which processes the data and activates lighting on road signs to warn approaching drivers.

According to Viani, et al. (2013) the architecture to detect wildlife-vehicle collisions consists of gateway nodes for data collection and forwarding to a control unit that implements processing and actuation strategies. It also consists of anchor nodes for wireless network management, actuator nodes for receiving actuation commands to turn on light signals and sensing nodes to detect animal movement.

According to Surya & Selvi (2017), the exisiting methods to solve human-wildfife conflict are the use of solar electric fencing and elephant GPS radio collars and seismic geophone sensors to track elephant movement and provide advance warning to local communities.

2.3.1: Remote sensing using wireless sensor networks

According to Kachhoria, Varma, & Radhakrishna (2019), large area wireless sensor networks are used to monitor unlawful activities in forests by using small, lighweight, battery operated sensor nodes capable of sensing, computing and communicating with each other through radio transmission.

In this model, sensor nodes are placed manually and data are routed through predetermined paths. The data sent are also event based with a predetermined position among deployed sensor nodes on the network (Kachhoria, Varma, & Radhakrishna, 2019).

2.4: Wildlife monitoring implementations

Open-source protocols such as MQTT and HTTP are popularly used to build scalable IoT networks and solutions. Terada, Yoshida and Ishibashi (2019) write that the MQTT based approach to creating reliable sensor networks is the most suitable for modern day IoT networks due to the fact that it subscribes to the publish / subscribe model as illustrated in figure 2.7.

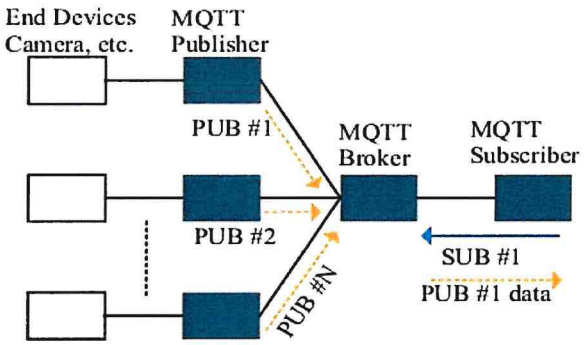


Figure 2.7: MQTT illustration (Terada, Yoshida, & Ishibashi, 2019)

In figure 2.7, a unique, hierarchical address also called a topic is assigned to the data where the publisher sends data to the MQTT broker and the broker in turn sends data to the subscriber who consumes data from the topic. MQTT runs on TCP and can set transmitting topics for each IoT device where each device sends sensor data and receives control messages. In the MQTT network, the publisher receives data from an IoT device and sends it to the broker. Furthermore, the subscriber specifies a specific topic to the broker. When the broker has the target data, it sends the data to the subscriber (Terada, Yoshida, & Ishibashi, 2019).

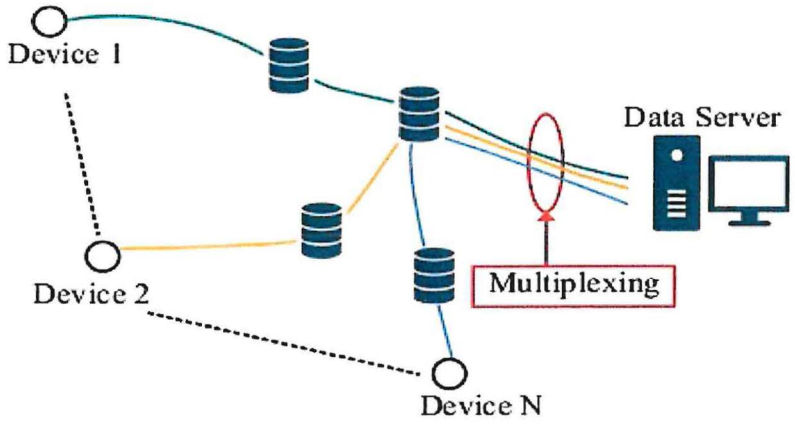


Figure 2.8: HTTP illustration (Terada, Yoshida, & Ishibashi, 2019)

Figure 2.8 illustrates the workings of HTTP for IOT implementation, In the figure, a request/response cycle is implemented by sending data packets across the network using the HTTP protocol. DNS resolution and route searching takes place to specify the server location by IP address. All the devices connected the network use this same request/response cycle to send and receive information to and from the data storage server. Multiplexing and packet loss may occur when a large number of packets is sent to the storage server at the same time which is resolved by optimizing the service for each system (Terada, Yoshida, & Ishibashi, 2019).

A wildlife monitoring system based on MQTT has been developed and has incorporated various components such as cameras to capture visual information of animals passing close to the sensors, an MQTT device to send and receive information, a message gateway hosted on a cloud server to process the information and a notification mechanism to alert residents of any animal activity near them (Terada, Yoshida, & Ishibashi, 2019). This is illustrated in figure 2.9.

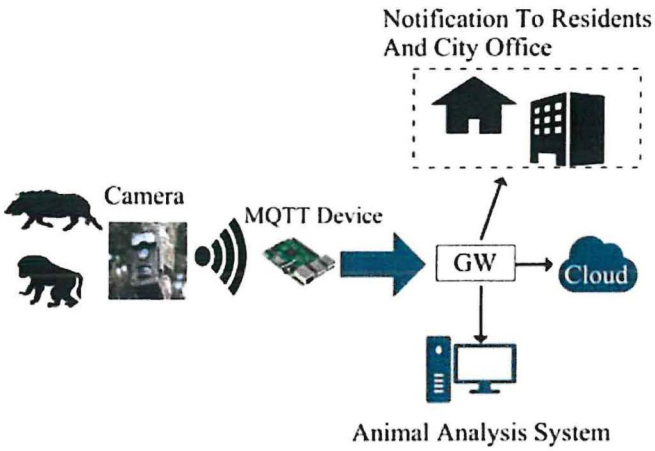


Figure 2.9: Sample monitoring system based on MQTT (Terada, Yoshida, & Ishibashi, 2019)

In the system, residents receive a notification of animal activity in their neighbourhood and they can then evacuate to a safe place. The model also provides for photographs of the animals via embedded cameras which then trigger alerts to residents of animal activity within their area.

2.4.1: Animal harm detection using sensors

There exists a solution to protect human beings from venomous insects and reptiles by using a combination of a sensor tower at a forest boundary to track movement of wildlife and humans near the boundary made up of a Raspberry PI 2 device and a camera (Indushree, Navya, Nandinisridevi, & Nikitha, 2019). The functionality of the solution is illustrated in figure 2.10.

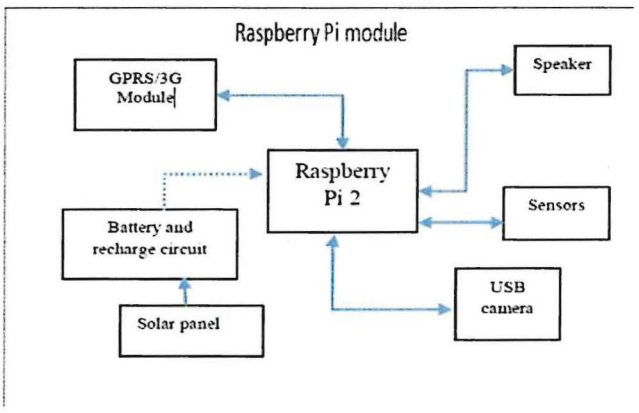


Figure 2.10: Reptile detection prototype (Indushree, Navya, Nandinisridevi, & Nikitha, 2019)

This prototype works by detecting the movement of a reptile or animal along with the body temperature, processing this data and creating an alarm to alert the user of the presence of such animals in their vicinity (Indushree, Navya, Nandinisridevi, & Nikitha, 2019).

2.5: Poacher detection using IoT and machine learning

Edemacu, Kim, Jang and Park (2019) write that "The small unit, low cost and distributed nature of IoT devices coupled with their ability to withstand harsh conditions allows them to be deployed for real-time monitoring in diverse range of areas." They aim to leverage the advantages of IoT devices to set up image capturing nodes with motion detection ability to capture and store images and motion data which will be analysed by a machine learning algorithm to filter out false readings and predict poachers. Their framework architecture had a perception layer that interfaced directly with the physical world, a middleware component which contained the machine learning algorithm to receive data from the sensor nodes through the network layer while also performing analysis, classification and categorisation. It also had an application component that provided the interface to display the results of the algorithm to users. Figure 2.11 illustrates the architecture of the model

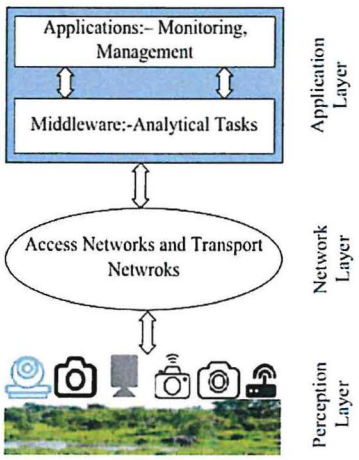


Figure 2.11: Poacher detection Prototype (Edemacu, Kim, Jang, & Park, 2019)

2.6: Conceptual Model

The data-to-cloud model of the IOT prototype will encompass four layers. There will be physical sensors which will be placed remotely in the field. They will use a transmission layer which will capture the data from the devices and transmit it to an application layer which manages the data, stores and processes it. The results will be presented via a web portal (Hodgkinson & Young, 2015).

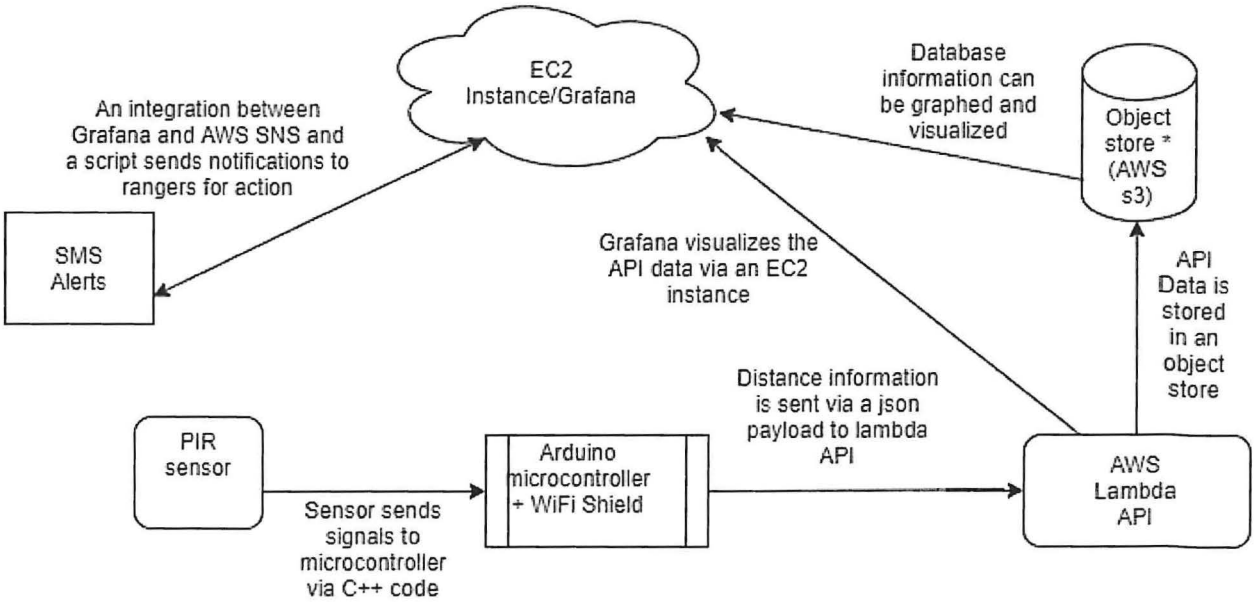


Figure 2.12: Conceptual model

The solution comprises of an Arduino microcontroller connected to a Wi-Fi shield for internet access. A PIR sensor is also connected to it to detect motion. The distance data generated from the sensor is sent to an API endpoint as a JavaScript Object Notation payload. The Grafana web visualisation tool will then fetch the metrics from this API and visualize them in graphical format for use by authorities such as the Kenya Wildlife Service while also having the functionality to trigger alerts to rangers in the field who will then take action and deploy to the affected areas.

Chapter Three: Research Methodology

3.1: Introduction

In this chapter, the approaches that were used to conduct the research are outlined. In this research, the Rapid Application Design also known as RAD was used in developing the prototype.

The Rapid Application Development methodology is suitable for this research because it simplifies and speeds up system development. This methodology is ideal in terms of providing a minimum viable product or solution that can be further improved upon (Susanto & Meiryani, 2019)

According to Susanto and Meiryani, (2019), the general stages of this methodology are system requirements analysis and specification whereby investigating and evaluating the criteria needed for the system to be built is done, system design where functional specifications of the solution are outlined, system testing where the system is built and tested component by component until deemed to be working correctly. The final stage is system implementation whereby the system is finally handed over to the end user for use.

In this research, requirements were gathered by the use of questionnaires administered to employees of Kenya Wildlife Service, and secondary literature from previous works and implementations.

3.2: Research Design

According to Akhtar, (2016), research design is the structure of the research that holds the elements of the project together. The research design used in this dissertation is the experimental research design due to the practical nature of the solution. The experimental research design also aided in identifying the gap present in mitigating human-wildlife conflict and the need to develop a robust IoT based solution to mitigate this conflict.

In this research, empirical data generated by the prototype's sensor, results from the questionnaire survey and graphical visualisations were used to determine the criteria to formulate a hypothesis and determine patterns that formed the criteria to receive SMS messages from the prototype.

3.3: Target population

According to Asamiah, Mensah and Oteng-Abayie (2017), data or information is gathered from participants who belong to the research population which is also the group of individuals having one or more characteristics of interest.

In this research, the target population consisted of employees of the Kenya Wildlife Service. These people played the role of respondents and as a source of primary data by answering a questionnaire illustrated in the appendix section at the end of this document.

The sampling technique that was used to determine the number of participants in the study was convenience sampling which is an implementation of non-probability sampling which was a technique that was low cost and readily available for the researcher. (Ilker, Abubakar, & Sunusi, 2016).

Convenience sampling was also chosen because it was simplistic and less time consuming than other methods taking into account the resource constraints of the reasearcher. (Stratton, 2021)

The sample size was determined to be the number of employees directly working in the organization's main office area who had a first hand experience of human-wildlife conflict reports and they were twenty five employees who participated in the study. They were also chosen due to their proximity to conflict prone areas and context in terms of the conflict reports they have received in the past.

3.4: Data collection methods

Both primary and secondary methods were used to collect data. The primary method that was implemented was the use of an online questionnaire. The online forms will be from an online platform such as google forms. The reason for using questionnaires is that it is an efficient and economical method to collect feedback. It takes a little time to distribute them online and receive responses remotely. The online questionnaire is convenient and gives the respondents adequate time to deliberate, fill and return responses. Secondary data was collected through a comprehensive literature review of existing publications from other scholars. Data from these two sources will be adequate to perform analysis and draw conclusions from the data. The general format of the questionnaire is as illustrated in appendix A at the end of this document.

3.5: Data analysis:

Collected data was quantitative and was analysed using Microsoft excel and google forms which is a spreadsheet application and a data collection tool respectively. Quantitative data was also gathered via the simulations carried out with the prototype along perimeter areas and recorded graphically via the web application running in the cloud as well as prior recorded data in publications and journals. The data from the filled questionnaires was also utilised and was visualized in pie charts.

3.6: System analysis and requirements

This research made use of the object-oriented analysis and design approach. In this approach, the researcher used use case diagrams, context diagrams and sequence diagrams to identify system requirements and visualize system functionality.

3.7: System Implementation

The IoT prototype was built using an Arduino microcontroller with a PIR sensor and Wi-Fi shield connected to it for motion detection and data transmission via the internet as a JSON object which is a standard text-based format for representing structured data. Metrics were visualized by Grafana, which is an open-source visualisation tool that can be accessed via a web browser and SMS alerts were triggered from the web application. The prototype also underwent thorough testing considering that data going into and out of the system had to be secure and confidential in line with the general architecture of IoT. (Kumar, Tiwari, & Zymbler, 2019)

3.8: Ethical issues

This research required participants to give consent on the data they will be providing the researcher. Privacy is key in determining consent of the respondents. Confidentiality was achieved by not collecting data on their names, ages, sex and other personal information. Consent is to be given after properly revealing the purpose of the research to the respondents.

Chapter Four: System Design and Architecture

4.1: Introduction

This chapter presents the results and findings of the data collection and analysis, and also describes the system requirements of the proposed prototype. The system design is achieved through design diagrams drawn using the Unified Modelling Language to thoroughly develop the solution before implementation.

4.2: Questionnaire Analysis

In order to determine the existence of the research gap, responses from the Kenya Wildlife Service were reviewed and analysed in this section. Questionnaires were administered to collect information on the status quo of human-wildlife conflict and how it impacts their day-to-day operations.

4.2.1: Internet Access

Of the respondents sampled, 92% had access to the internet via a smartphone and a computer and only 8 % of respondents had no access to the internet due to lack of a smartphone and computer. The data is illustrated in the figure 4.1.

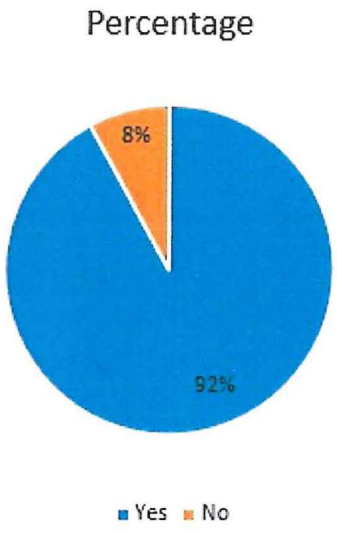


Figure 4.7: Internet Access Statistics

This therefore means that most of the respondents do have sufficient internet access and good network coverage in the areas they are located in. It further implies that sending SMS to the stakeholders is a viable means of communicating events captured by the system.

Receiving SMS does not require an active internet connection but relies on telecommunications infrastructure which is present in most parts of Kenya including the parts where rangers are present. This presents an extra advantage because the telecommunication networks are reliable.

4.2.2: Reporting Statistics

Human-wildlife conflict is reported to the organization at least 98% of the time in every month which means that human-wildlife run-ins are commonplace and actively affect communities living in close proximity to perimeter areas. This is illustrated in figure 4.2.

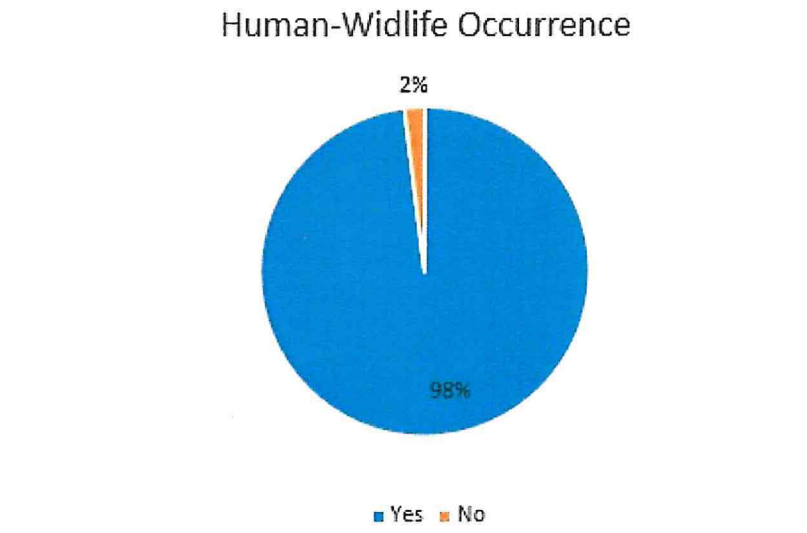


Figure 4.8: Human-Wildlife Conflict Occurrence statistics

Due to active run-ins between people and wild animals, a lack of pre-emptive knowledge about impending is clearly seen. This research aims to mitigate this lack of pre-emptive knowledge by supplying alerts to rangers, nearby residents and staff belonging to KWS.

4.2.3: Methods of Reporting

From the respondents, 87% of reports on human-wildlife conflict were manual; that is reports from individuals that live in surrounding areas whose properties get infiltrated by wild animals. The other sources of reports of wild animal movements were from GPS data from endangered animals fitted with GPS collars transmitted via satellite communication which was 9%, triangulation using radio communication which was 3 % and drone technology which was experimental at 1% of all

reports. This data highlights the inefficiency in reporting which is mostly after the conflict has already occurred. Figure 4.3 illustrates the data.

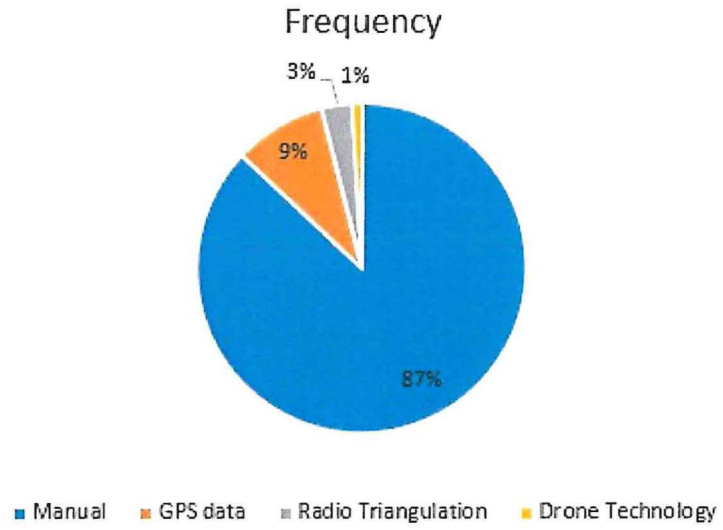


Figure 4.9: Methods of Conflict Reporting

Since the information about real time conflict occurrences are few and far between, using an SMS alerting system will greatly increase the amount of information that can be used to defend against wild animals or take action to hide or move to a safe area in good time.

4.2.5: Data Analysis Summary

The findings of the questionnaire data analysis indicate a gap in the use of technology to mitigate human-wildlife conflict. A low-cost technological improvement in terms of identification and collection of motion metrics to increase context on the likelihood and of occurrence of human-wildlife conflict would be of great benefit to the authorities in charge of preventing it.

4.3: Requirements Specification

This section describes the features and operational behaviour of the IoT prototype. It involves identifying and elaborating functional and non-functional requirements which are elements of the system that will determine how the system will work. In this section, key parameters of the prototype will be defined.

4.3.1: Functional Requirements

Functional requirements describe what has to be accomplished by the prototype by identifying the necessary task, action or activity that must be done. They are:

1. The prototype should read motion data from the environment via the PIR sensor.
2. The prototype should convert sensor signals into digital form and send it to AWS S3, where data from the API will be stored as objects.
3. The IoT cloud platform, in this case an AWS EC2 instance with a running instance of Grafana should visualise API and object storage data.
4. An SMS should be triggered by a call made to the API as it signifies a violation of the proximity threshold.
5. The prototype should collect and store logs.

4.3.2: Non-Functional Requirements

Non-functional requirements are constraints that exist on a system that do not affect the overall functionality of the system. In this IoT prototype, they are:

1. Security – Through role-based access control, the system should implement authorization. Other aspects of security such as confidentiality are to be implemented through secure protocols and encryption.
2. Reliability – The solution should have minimal downtime to ensure that data is not lost in transit and that it can be accessed when needed.
3. Scalability – The design of the system should be adaptable such that new features can easily be added and that current features can be upgraded.
4. Usability – The prototype should be straightforward and simple to use and understand.

4.4: System Architecture

The IoT prototype has been designed in line with the device to cloud model as described in the literature review. The PIR sensor for collecting motion metrics is connected to the Arduino uno microcontroller and constitute the physical layer of the prototype. The cloud server configured with implementations of both the functional and non-functional requirements discussed in sections 4.3.1 and 4.3.2 respectively will also be key components of the architecture.

The Arduino microcontroller converts the analog signals from the sensors into digital values which are subsequently sent through the internet as objects and stored in an object data store. The web dashboard running in the cloud is then configured to display the metrics from the API on a web dashboard as well as trigger and SMS when an object is close enough to the sensor.

4.5: System Analysis

4.5.1: Use Case Diagram

This diagram describes the various interactions between the users of the system and the system itself. In the IoT prototype, the actors in the system comprise of the administrators, game rangers and authorized users of the system as illustrated in figure 4.4.

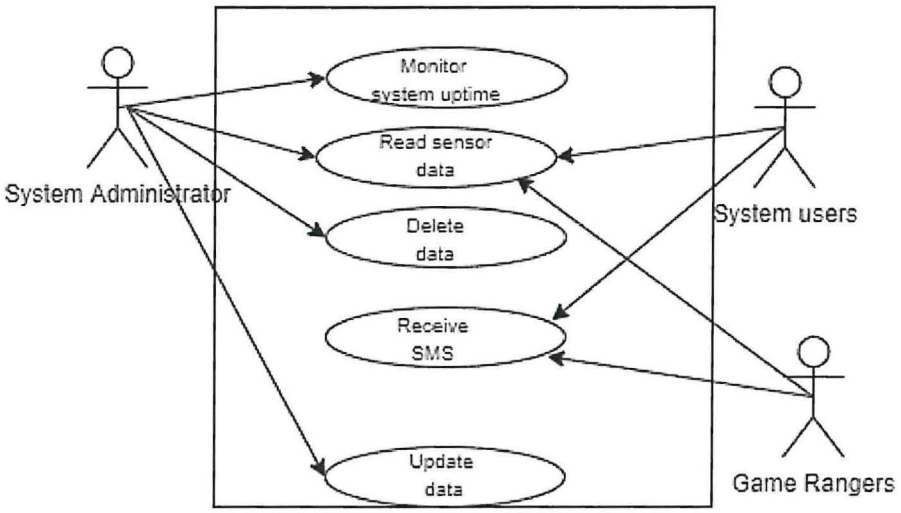


Figure 4.10: Use case diagram

The three main actors and the role they play while interacting with the system is shown in figure 4.4. The system administrator is there to ensure the smooth functioning of all the components of the system while the game rangers will mostly make use of the SMS messages that they received from the system for actioning with an option to sign in and view the metrics. The System users are those who interact with the system without necessarily being in the field such as office-based employees.

4.5.2: Use Case Scenarios

Use case scenarios are used to refer to the detailed step-by-step interactions between actors and the system itself. The description of the interaction is done in a use case narrative which explains a transaction to completion whether it succeeds or fails. For the IoT prototype, the use case narrative is as below:

Use Case Name: Monitor System Uptime

Description: The interconnected components which include the cloud services and the hardware components need to be on always to achieve the desired results

Primary Actor: Admin

Trigger: The PIR sensor not sending data for a prolonged period of time and cloud services being offline.

Pre-condition: The microcontroller needs to be connected to the internet and the PIR sensor needs to send data to the API via the connection and configuration with the microcontroller and Wi-Fi shield.

Post-condition: The hardware components are on.

Use Case Name: Read sensor values

Description: Sensor values are read by a C++ program and sent to the microcontroller.

Primary Actors: System Users

Trigger: Wi-Fi shield sends JSON object to API which populates graph in Grafana

Pre-condition: The PIR sensor sends signals to the microcontroller.

Post-condition: Objects are stored in an S3 bucket.

Use Case Name: Receive SMS

Description: Users of the system and game rangers should be able to receive an SMS containing actionable information.

Primary Actors: Game Rangers

Trigger: When an animal gets too close to the sensor, an SMS is triggered and sent to the phone of the game ranger

Pre-condition: Lambda function receives JSON object from sensor.

Post-condition: SMS is sent via AWS SNS and logged by AWS CloudWatch.

4.5.3: Sequence Diagram

A sequence diagram is used to show interaction between different system classes. It helps to visualize and validate different runtime scenarios that will be implemented in the solution. In this research, it enables logic flow modelling of the IoT prototype visually. The sequence diagram in figure 4.5 shows the interaction between the main system processes.

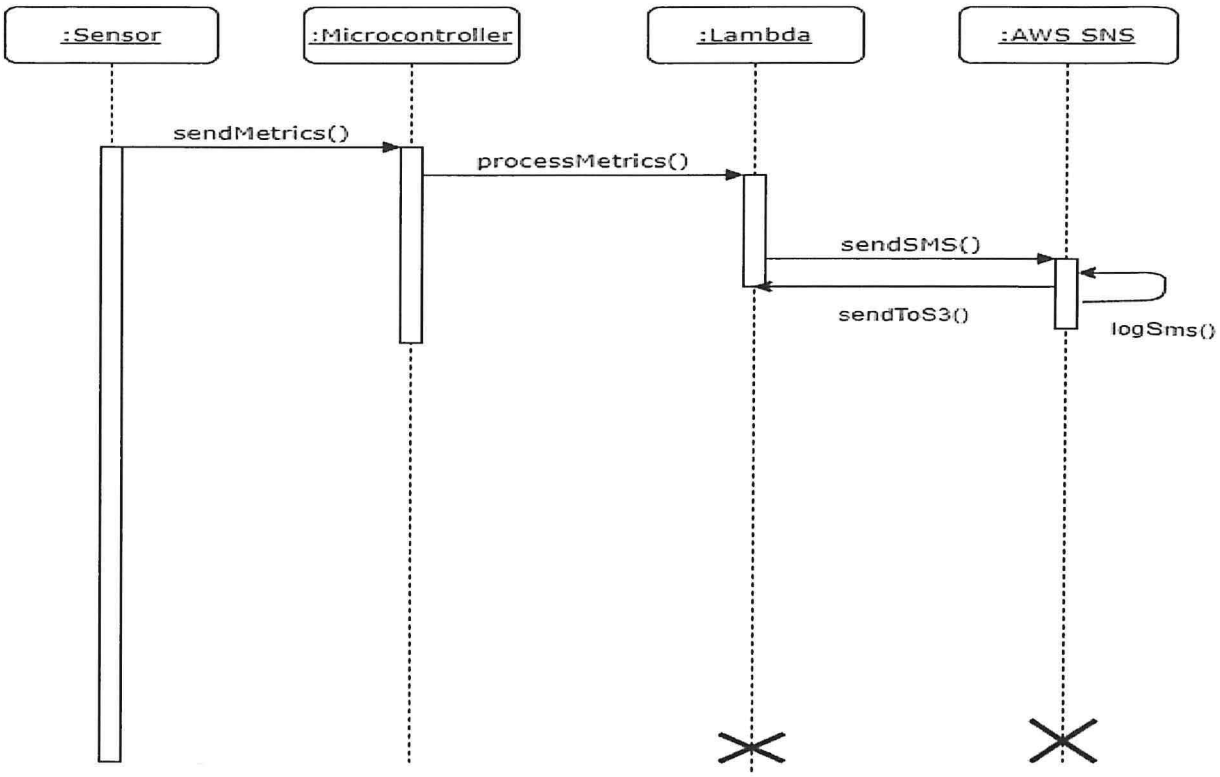


Figure 4.11: Sequence Diagram

4.5.4: AWS S3 object store

Data that will be sent to the API will be stored in an object store which is appropriate for key value pairs and time series data which is the exact type of data that was generated by the PIR sensor. S3 will act like a persistent store for objects generated by the code running the microcontroller and Wi-Fi shield. An illustration of the functionality is shown in figure 4.6.

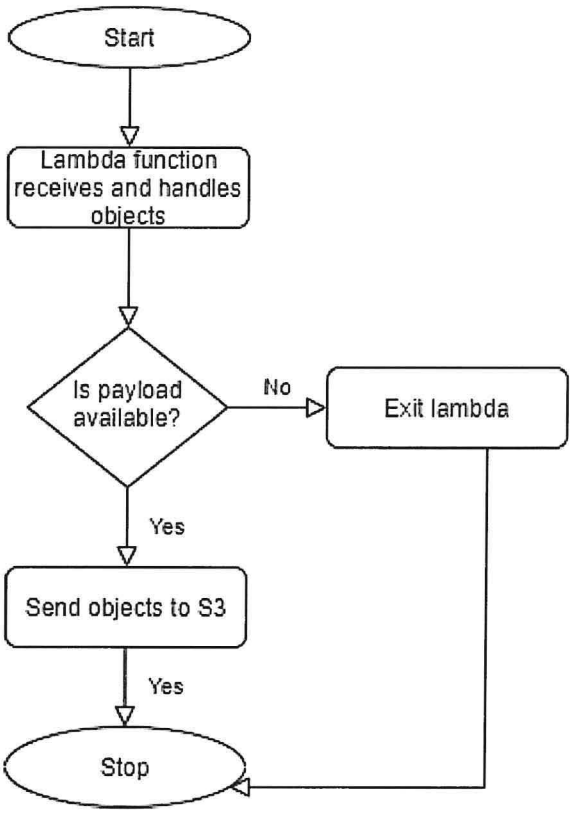


Figure 4.12: S3 flowchart

Chapter Five: System Implementation and Testing

5.1: Introduction

This chapter highlights the implementation of the architecture of the IoT prototype. It shows the physical connections and wiring of the Arduino microcontroller to the Wi-Fi shield, the power supply mechanism to the Arduino board, Wi-Fi shield, breadboard and motion sensor, the data transmission mechanism via HTTP to the API hosted on the cloud server and how the SMS is triggered when the JSON payload meets the API. This section also highlights the visualisation of the API, SNS and CloudWatch metrics on the Grafana interface.

5.2: System Components

The physical components of the prototype comprise of a Passive Infrared Sensor, a transmission layer consisting of an Arduino microcontroller which will be programmed to transmit digital signals to a database. The virtual components of the prototype include a lambda API and function, an AWS SNS integration and a web application running on an AWS EC2 instance.

5.2.1: Hardware components

1. **Passive Infrared Sensor:** The Sensor that was used is a proximity sensor. It was used to detect motions along a perimeter area with a predetermined range.
2. **Arduino Uno Microcontroller:** This component will interface with the AWS EC2 instance to send metrics to an API and lambda function that will store the metrics in an S3 bucket. The microcontroller board is based on the ATmega328P. It has 14 digital input/output pins, 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button.
3. **Wi-Fi shield**: This component interfaces directly with the Arduino microcontroller and adds functionality to the microcontroller that enabled it to connect to the internet.

5.2.2: Application Layer

In this layer, sensor data is organised, stored and visualised. The sensor data is captured by a program that runs the interconnected physical components. This data, in the form of JSON objects is then sent to an API endpoint configured on the AWS cloud environment that runs behind a lambda function which triggered an SMS from AWS SNS when the distance threshold was met.

The SMS event and error messages were logged by AWS CloudWatch for troubleshooting and investigative purposes, and the API objects were stored in S3 for future manipulation.

5.3: System Implementation

This section covers the technical implementation of the physical IoT devices, the web application and data visualisation solution that the prototype will achieve.

5.3.1: Microcontroller and PIR setup

In figure 5.1, the Arduino microcontroller is physically connected to the PIR sensor and Wi-Fi shield via a breadboard as illustrated in figure 5.1. This connection and the underlying program enabled the Wi-Fi shield to connect to the internet and transmit metrics and the sensor to send data to the Wi-Fi shield. The prototype has an extra advantage in case of loss of power in that it can run on batteries which is proof that it only requires limited amounts of electricity making it scalable in the long run. The prototype can also be connected to a solar panel to act as a failover in case there is loss of power or drained batteries especially in areas that are remote.

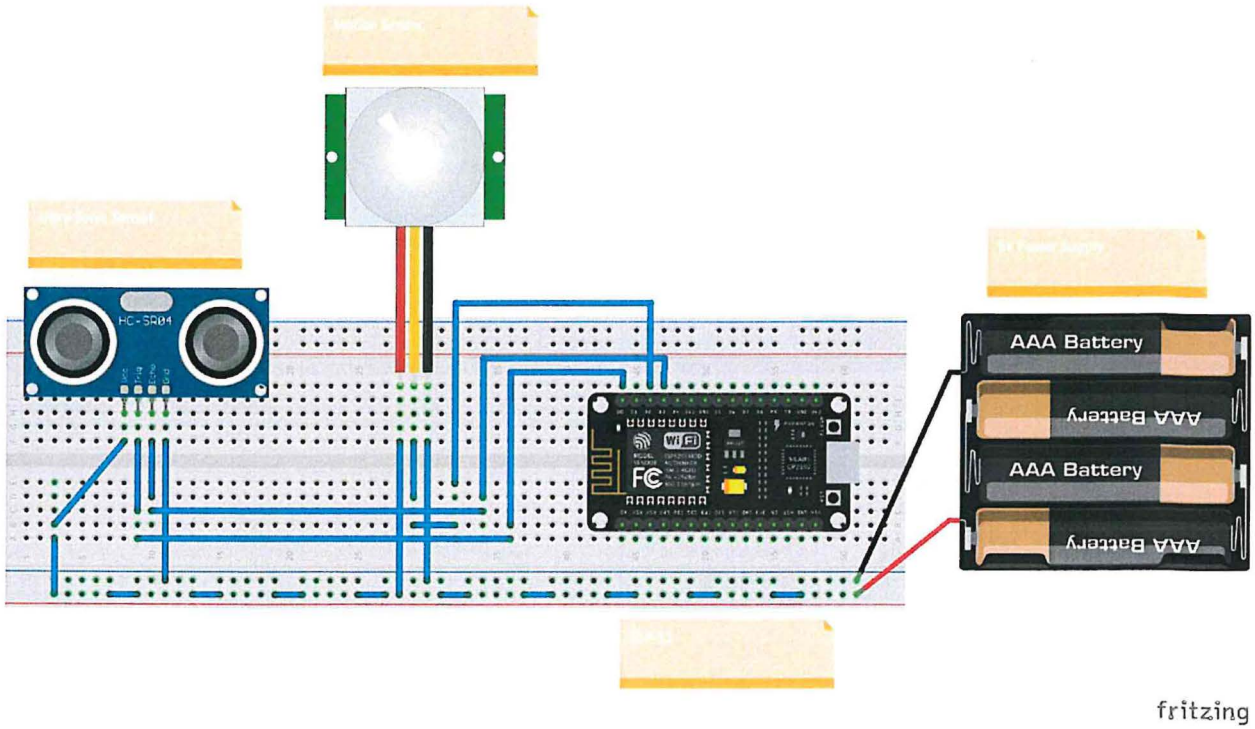


Figure 5.19: Microcontroller and PIR setup

The data was sent to the shield via HTTPS once the distance threshold was met, in this case, due to the limitations of the coverage of my sensor which is four metres, I chose the threshold as two metres before triggering a request to the API.

5.3.2: Setting up the API

The API was created and configured using the AWS API console and contains python code for triggering SMS via AWS SNS. Figure 5.2 below shows this effect.

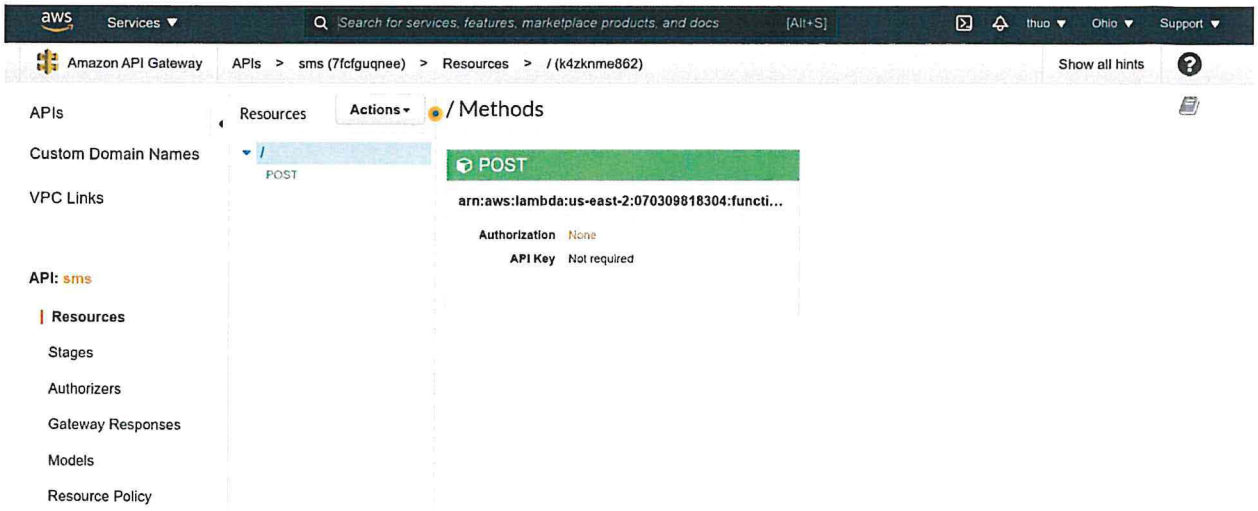


Figure 5.20: Lambda API setup

The API accepts POST requests only for the scope that the prototype covers to handle incoming data from the sensor. The landing page also offers insightful metrics about the API which can be used to troubleshoot and improve the API as shown in figure 5.3.

Figure 5.21: API metrics and Information

In figure 5.3, the API endpoint that will receive the requests used can also be seen. Subsequently, once a call has been made to the API, that event, whether it contains an error or not, is logged by AWS CloudWatch for troubleshooting and future log analysis as shown in figure 5.4.

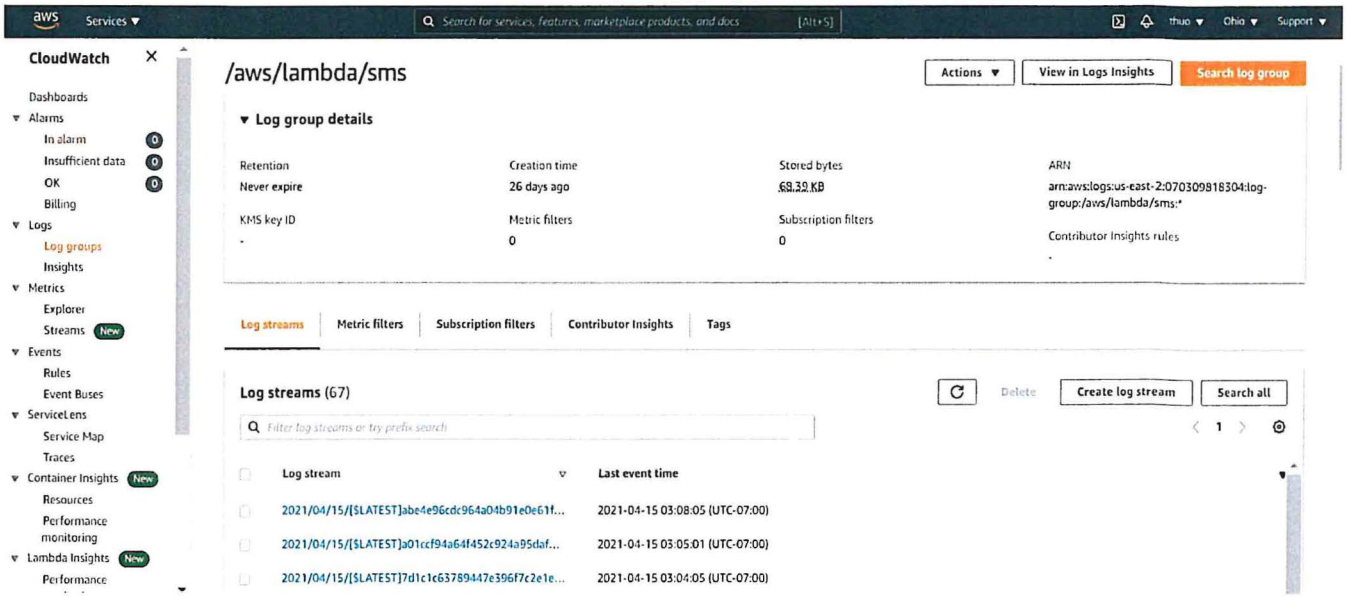


Figure 5.22: CloudWatch logs

In figure 5.5, the code that handled the POST requests coming from the sensor and the AWS SNS trigger coming from the sensor is shown.

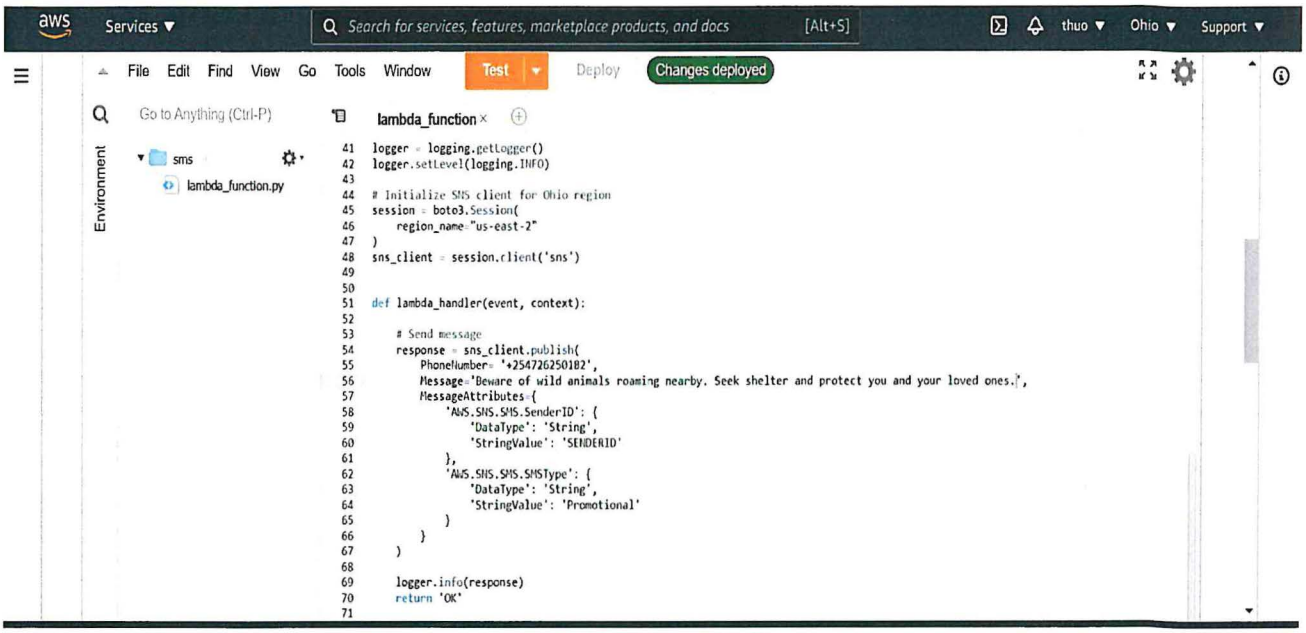


Figure 5.23: Lambda function code

5.3.3: Setting up S3

The objects generated once metrics are sent to the API are stored in an object store, AWS S3 which can store multiple types of objects which gives flexibility in terms of future modification of the prototype. Figure 5.6 shows how the bucket was set up.

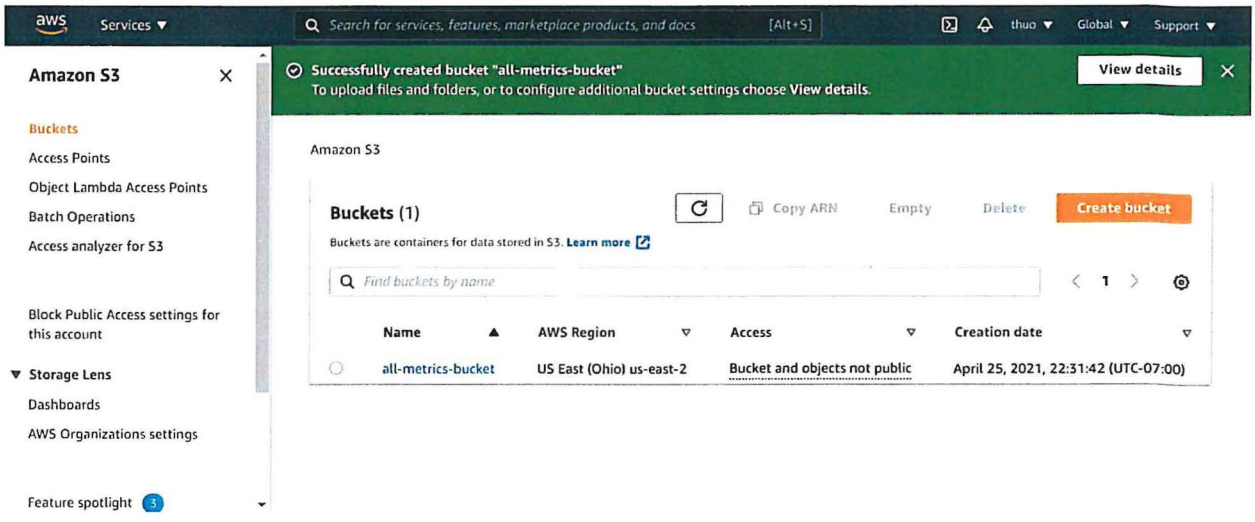


Figure 5.24: S3 bucket

Figure 5.6 shows the contents of the bucket which were the payloads processed by the API.

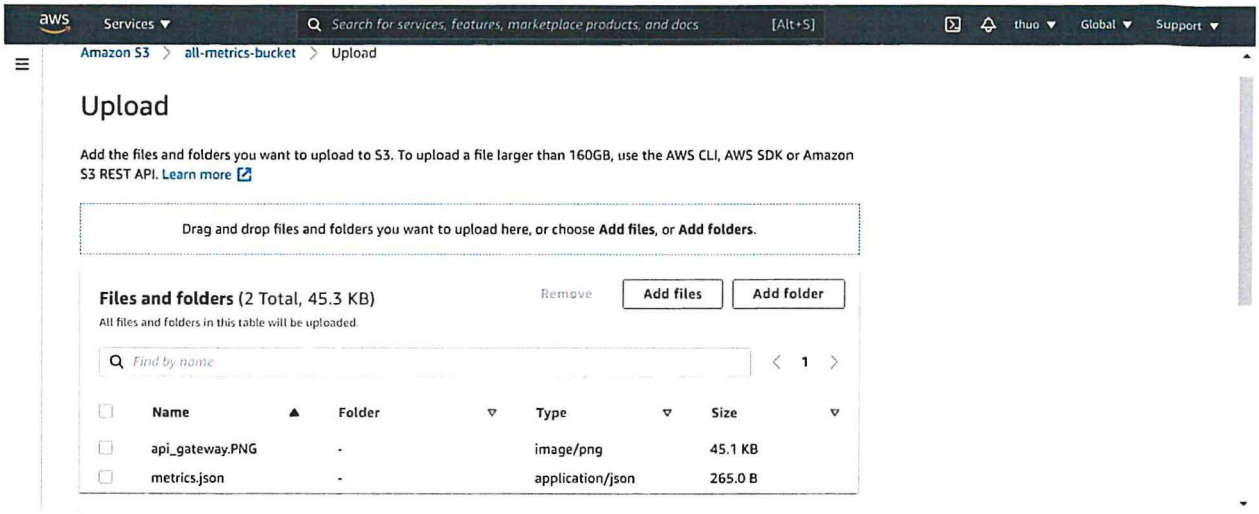


Figure 5.25: S3 bucket contents

5.3.3: Configuring AWS SNS

AWS SNS was used to deliver SMS messages to mobile phones to alert rangers and staff of impending conflict. The message contains actionable information. It could be a warning or an update. An example of these messages is shown in figure 5.8.

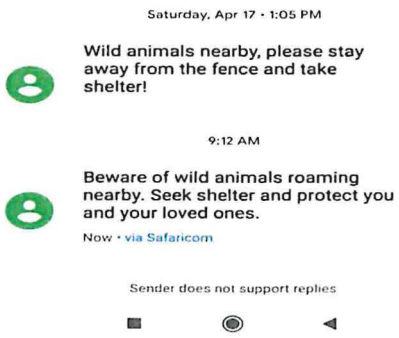


Figure 5.26: Sample Messages

SNS configuration is comprised of setting up the message types which are promotional messages. The success rate of delivery can also be seen on the SNS dashboards as shown in figure 5.9.

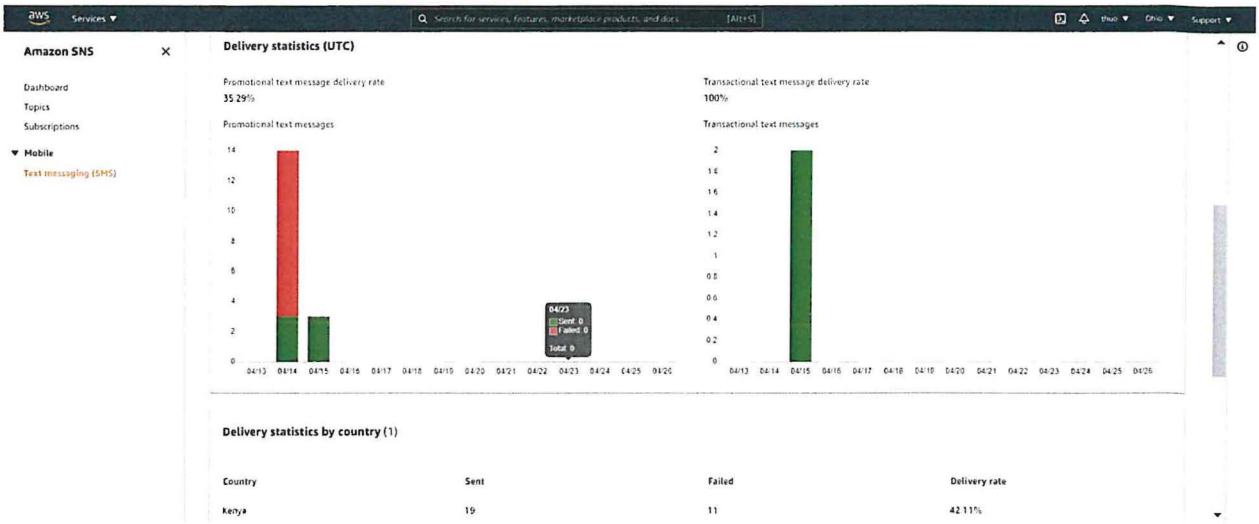


Figure 5.27: SNS dashboard

5.3.4: Setting up Grafana

Grafana was used to visualise motion metrics, lambda metrics and SNS metrics all in one view each with a dashboard of its own. This is to enable authorized staff and rangers to see the data and statistics which will enable them to make decisions about resource allocation in conflict prone areas.

Grafana ran in a Linux environment running the Debian based ubuntu distribution. This operating system was set up as an EC2 instance with network information to access the server remotely via the secure shell protocol.

The Grafana instance contains all the Grafana settings and configurations including the public IP and public/private key pair that I used to access Grafana via SSH. Figure 5.10 shows the instance details of the EC2 instance.

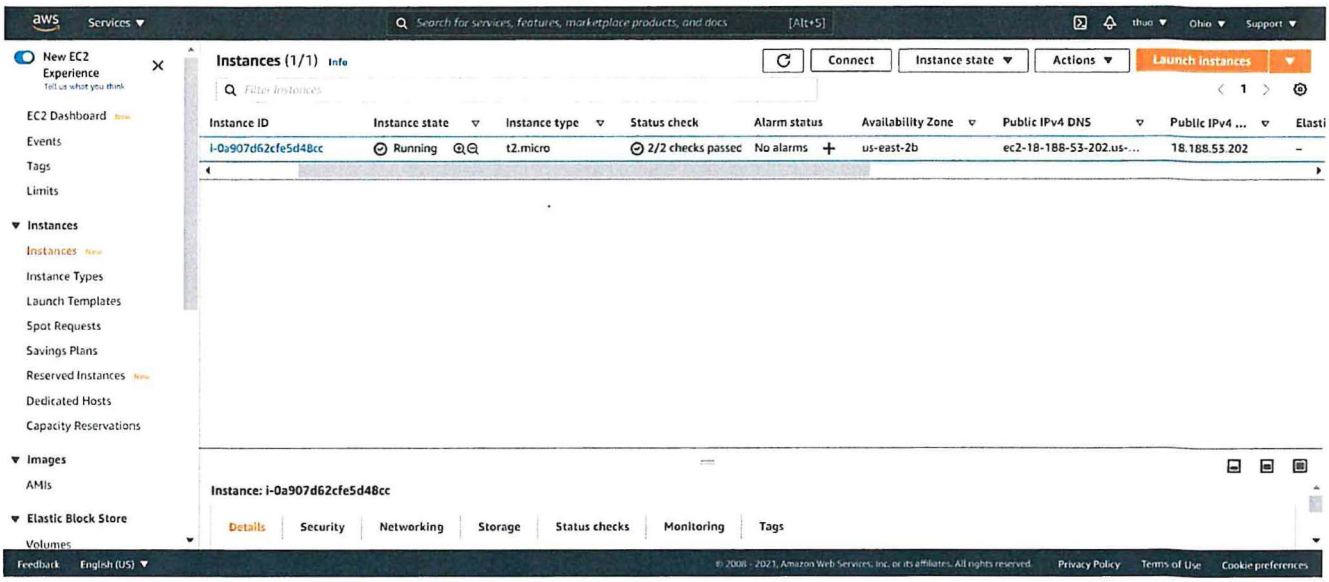


Figure 5.28: EC2 Dashboard

Once access was granted into the instance, I installed and verified that the app was active and accessible to the outside world by adding security groups which are rules that govern what kind of traffic is allowed into my instance as shown in figure 5.11 and 5.12 respectively.

Figure 5.29: Running Grafana service

The Grafana service was also configured to run on startup in case the server is restarted for maintenance purposes.

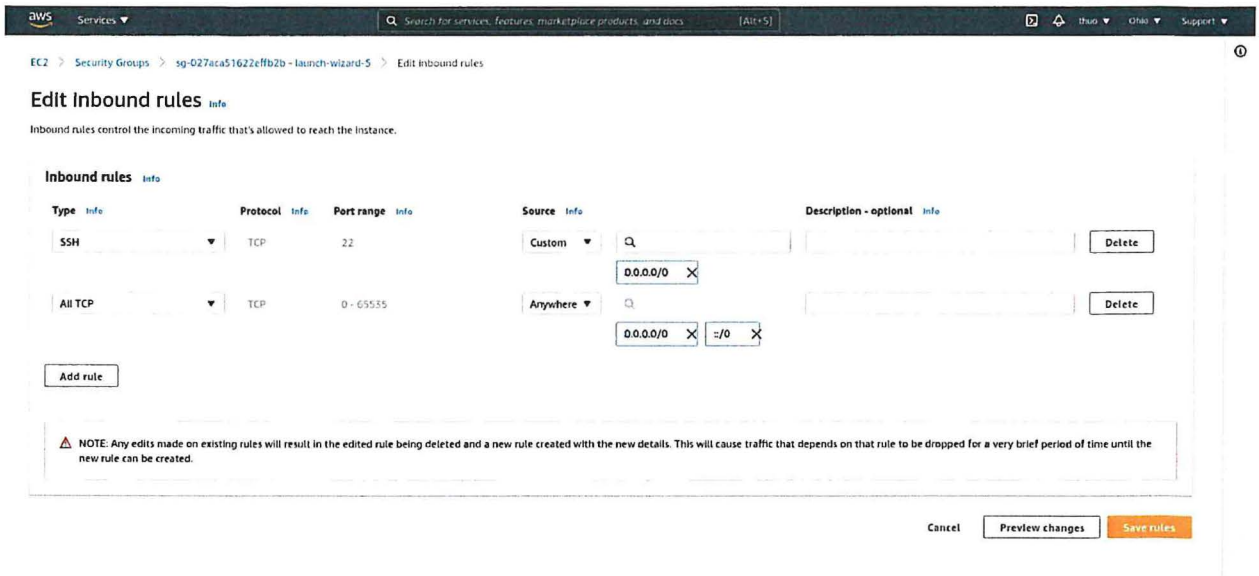


Figure 5.30: Security groups

As illustrated by figure 5.12, security groups were configured to allow SSH and TCP traffic into the instance for external accessibility and SSH access into the server.

5.3.5: User Account creation and Management

For user account creation and management, Grafana has a provision for role-based access control. The initial admin user is created manually by signing in with the default password and provisioning new credentials. Thereafter, administrative and non-administrative users are managed by the admin users and assigned read, write or execute rights based on the need for access. Figure 5.13 illustrates the admin user sign in interface.

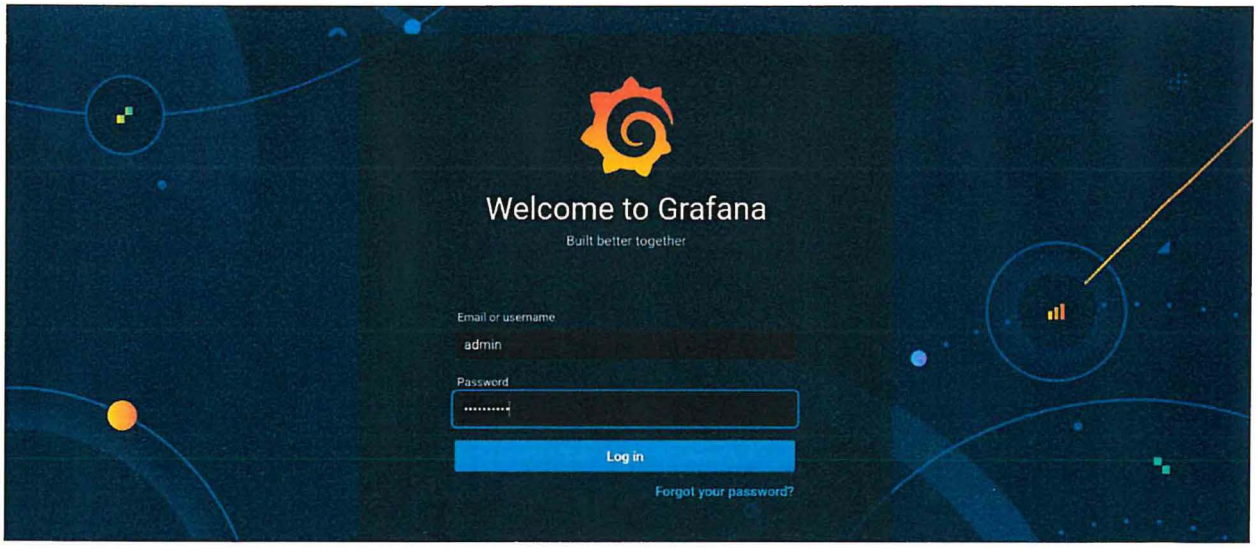


Figure 5.31: Admin user sign in

Figure 5.14 further illustrates the user management dashboard with various options and the permissions that users are allocated according to what type of access their role in the organisation requires.

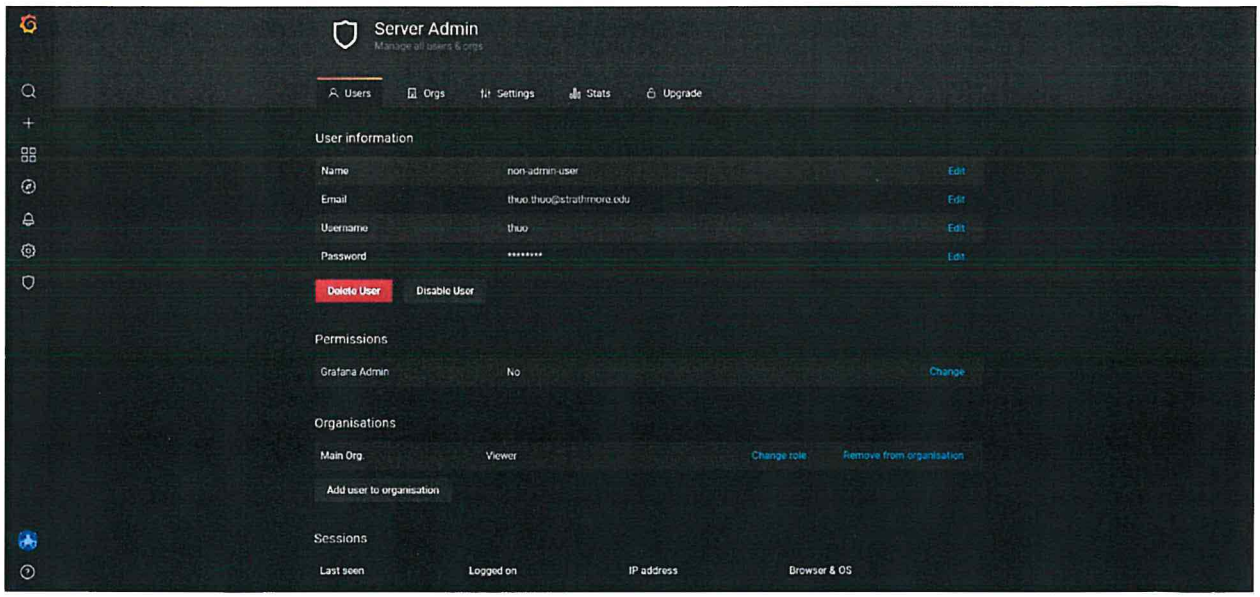


Figure 5.32: User rights and permissions

Figure 5.15 shows a list of two users presently in the system. One administrative and one non-administrative.

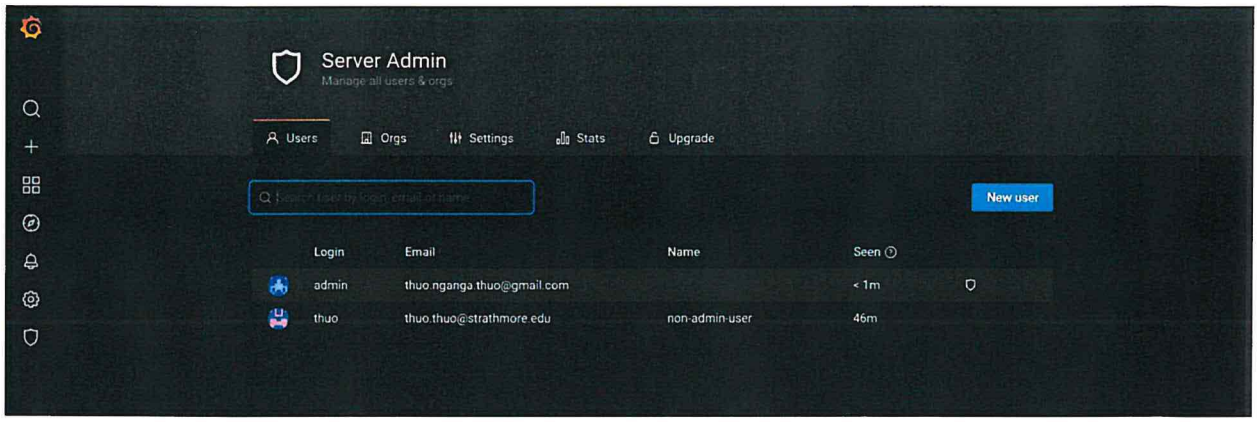


Figure 5.33: List of users

The administrative user has super user permissions in the system meaning that administrators can perform create, read, update, and delete operations whereas the non-administrative users are limited to read permissions only to reduce human error and enforce security.

5.3.6: Configuring graphs

In order to visualise metrics going through the system. Creating graphs is the best way to see data in graphical or tabular format within the system and accelerates decision making because of the presence of statistics that can inform decision making. Figure 5.16 illustrates how authentication into services within the AWS ecosystem were configured.

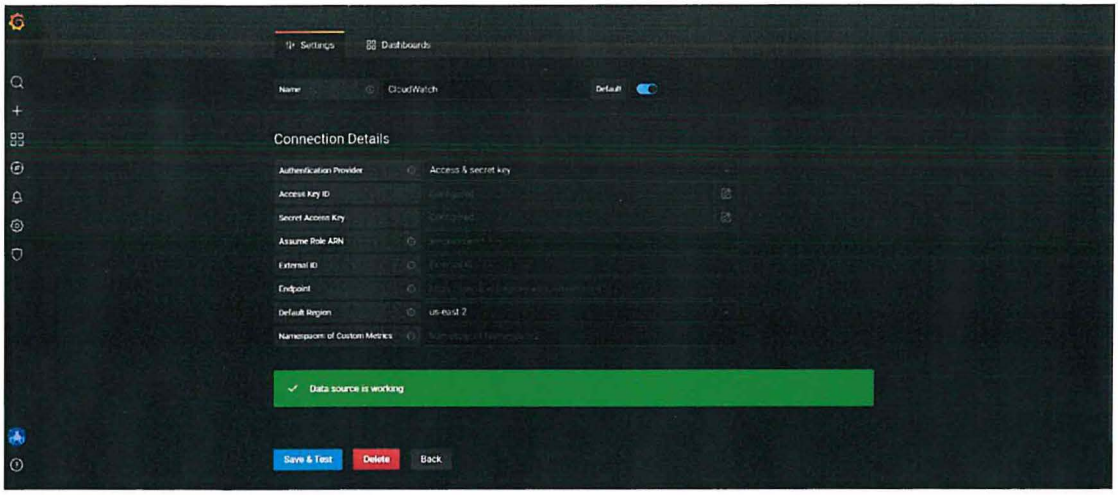


Figure 5.34: Visualisation configuration

Connection information is filled on a web form by an administrative user which contains an access key and a secret access key to authenticate into the CloudWatch environment and check for historical and real time metrics in a predetermined interval. This can also be done via SSH on a command line interface in case connectivity is slow or the dashboard is inaccessible Figure 5.17 shows how the graphical visualisation is implemented.

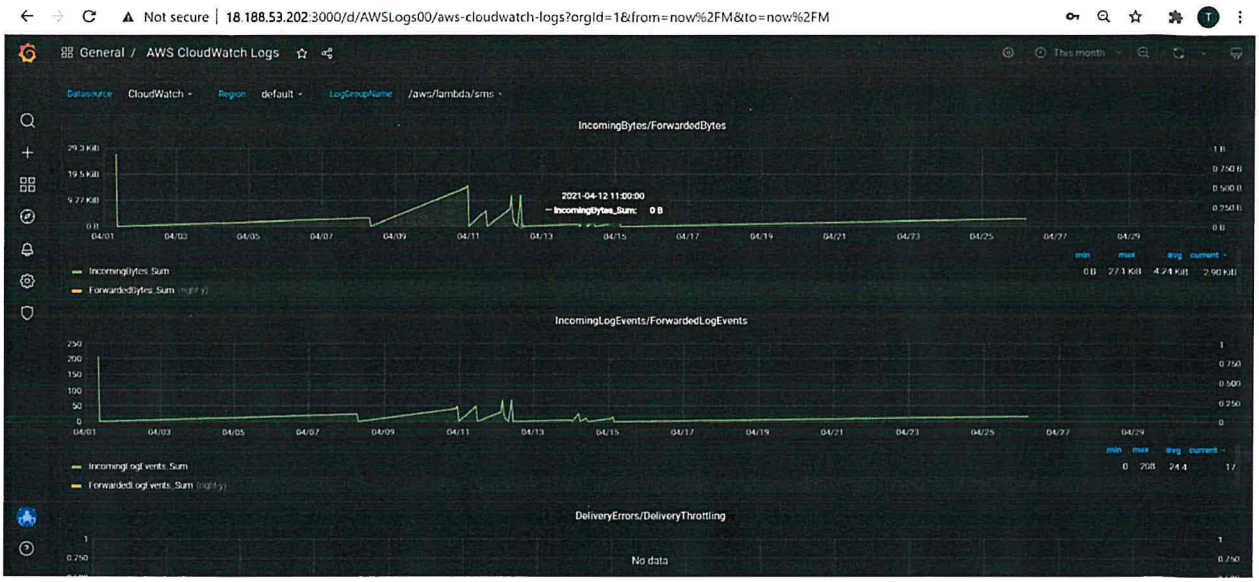


Figure 5.35: Graphical CloudWatch visualization

5.3.6: Configuring Alerts

Alerts are configured on Grafana as webhooks and integrated into Grafana dashboards to trigger POST requests when a query or rule has been violated on the dashboard. Figure 5.18 shows the configuration of alert notification channels.

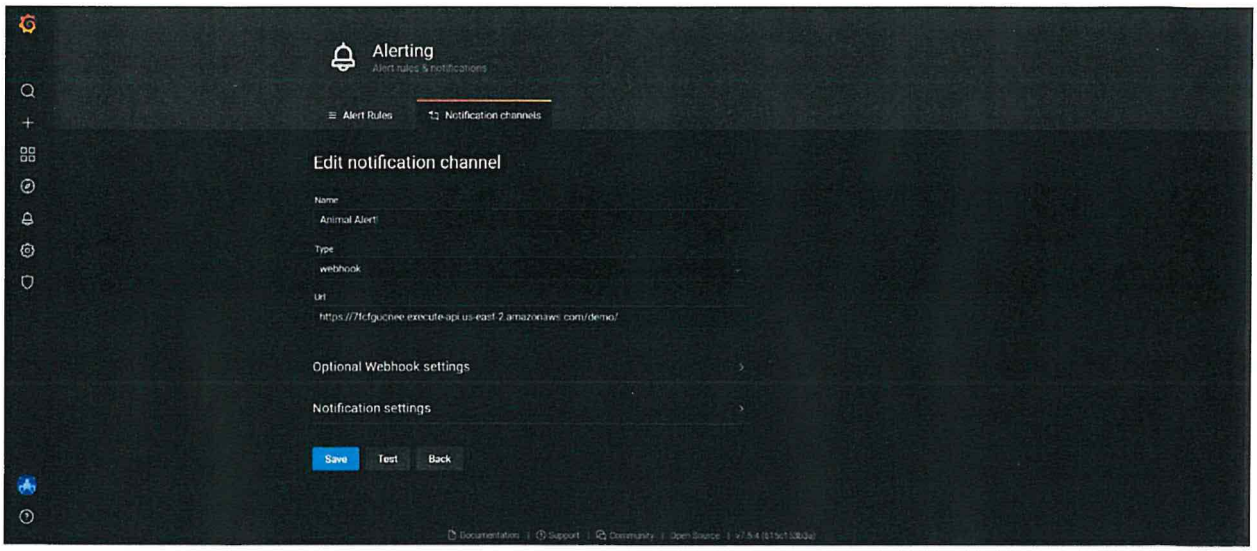


Figure 5.36: Notification channel

This notification channel scrapes information via the webhook configuration from the API endpoint to detect POST requests. These requests then trigger an SMS from SNS if the rules used to configure the query on the dashboard are violated.

5.4: System Testing

The solution was tested to ensure compliance with the functional requirements as defined in chapter 4 and to ensure the first iteration of the solution had minimal bugs.

5.4.1: Functionality Testing

The functionality tests were performed on the Arduino microcontroller in an iterative fashion to determine whether user requirements were met. Table 5.1 showcases the test cases that were fulfilled. For the solution to run as expected the defined test cases in the table must have been met in order to ensure completeness.

Table 5.1: Functionality testing

Test Case	Description	Priority	Results (0-5 five is highly effective while zero is non-functional)
Functionality	Confirm that the PIR sensor sent signals to Wi-Fi shield	High	4
Functionality	Confirm that API endpoint received and processed json payload	High	4
Functionality	Confirm that requests are logged by CloudWatch	High	5
Functionality	Confirm that Grafana can authenticate and read metrics from AWS services	High	4
Functionality	Confirm that user account creation and storage is successful	Medium	5
Functionality	Confirm that sending SMS is functional and successful	High	4
Functionality	Confirm that objects are stored in s3	Low	2.5

5.4.2: Compatibility Testing

The prototype was hosted on an Ubuntu Linux operating system which is a GNU based Linux distribution on an AWS EC2 instance. Tests were later carried out on the web application to establish that the metrics were visible on all the different browsers with no bugs. The outcome of the tests is illustrated in table 5.2.

Table 5. 2: Browser Compatibility Tests

Browser Type	Compatibility
Microsoft Edge	Yes
Google Chrome	Yes
Internet Explorer	Yes
Mozilla Firefox	Yes
Chromium Browser	Yes

5.4.3: Acceptance Testing

To gather feedback on the effectiveness of the application in terms of user experience, a short survey was done on the respondents, in this case, KWS to determine the satisfaction levels of the users of the application. The results are displayed in chart 5.19.

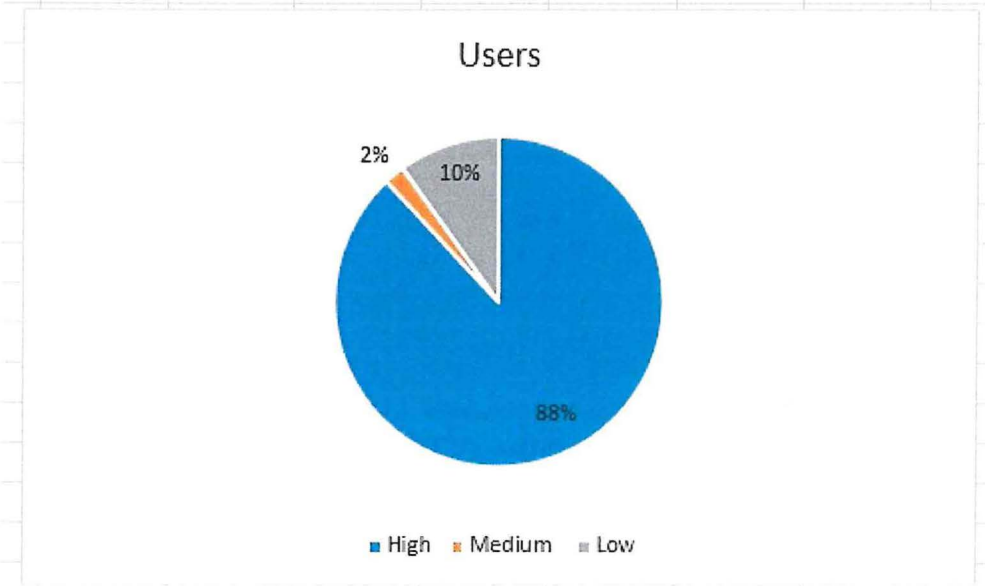


Figure 5.37 : Acceptance Test Results

A majority of the users at 88% felt that the application was good enough to use as is and accepted the solution. A further 10 per cent thought that the application was too complex for them to understand and recommended simplifying the dashboards. The remaining 2% of the users accepted the application with moderate feedback about it.

Chapter 6: Discussions

6.1: Introduction

The IoT prototype solution was based on the research findings achieved in the previous chapters. The prototype's functionalities were tested to abide by the research. In this chapter, an analysis was done to determine the correlation between the findings and the research objectives as well as the literature review. In this section, the relationship between the research objectives findings and literature review are outlined.

6.2: Existing Methods That Address Human-Wildlife-Conflict

In this research, the first objective that was met was a review of existing methods that deal with human-wildlife conflict. Though there are non-technological solutions to this conflict, this paper focused on the technological solutions that have currently been implemented.

This section was addressed in the literature review where it was established that the use of IoT sensors in various implementations exist. This includes the use of electric fences, GPS collars, drones, radio and satellite communication to mitigate human-wildlife conflict and track the motion of wildlife. There are also implementations of IoT focused on poacher detection using IoT and machine learning as well as reptile detection using a camera trap and temperature sensors. This research showed that it is viable to use IoT technologies in combination with other technologies such as machine learning, SMS and cameras to build a solution to mitigate human-wildlife conflict.

6.3: Current Challenges of IoT based Monitoring

In Chapter two, it was established that the current methods of addressing this conflict are expensive, inaccurate and difficult to scale. This research revealed a gap that needed a low-cost, accurate and easily scalable solution that combined ubiquitous technologies such as SMS to be implemented to tackle human-wildlife conflict in combination with on demand cloud infrastructure to solve the problem of scaling over time as more devices are connected to the system.

The current methods of IoT based monitoring are also not well adopted due to their complexity and lack of accessibility for example radio and GPS technologies which require expert knowledge, training and heavy capital investment to implement. This reduces the overall effectiveness and penetration of these implementations.

6.4: Development of the IoT Solution

The IoT solution was implemented using low power and affordable IoT hardware devices and cloud resources. The data that was generated and visualised by this solution was accurate and yielded tangible results that could be used by rangers, staff and the system administrator to aid in their decision-making process. The development was done using Rapid Application Development approach that focused on iteration and improvement until the minimum viable product was ready for shipping.

6.5: Testing the IoT Prototype

The functionality of the prototype was thoroughly tested to ensure it met the defined functional requirements. During testing, bugs were identified and rapidly fixed to adhere to the requirements that were pointed out in Chapter 5. The prototype was tested for compatibility with web browsers which was successful as well as user acceptance in which most responses were affirmative.

During functionality testing, the PIR sensor would detect motion, the microcontroller could process the metrics it received from the sensor and send the API endpoint as a JSON payload. The web application would display the metrics accurately in graphical format after querying successfully from AWS as the primary data source.

During acceptance testing, a majority of feedback from the users was positive feedback in terms of the fact that they felt that the prototype was useful and relevant in improving their overall operations and added value to the technological efforts against human-wildlife conflict in the conservation industry at large.

Chapter 7: Conclusions and Recommendations

7.1: Conclusion

This research was aimed at developing an IoT solution that mitigated human-wildlife conflict by combining cloud technologies with IoT technologies to visualise motion data along perimeter areas. The solution generated metrics triggered by motion which were translated into signals and displayed in an organized fashion on a web application to be referred to, analysed and used for operational decision making.

From the findings in this research, human-wildlife conflict is a prevalent problem that has existed for a long time with minimal and insufficient technological intervention. Using IoT, an end-to-end solution was developed to enrich the process of motion detection along perimeter areas which would yield metadata that would be used by authorities to allocate resources to affected areas strategically and efficiently.

The solution was also low-cost indicating the economic incentive to implement it at scale. The prototype developed was also tested and found to be accurate in detecting motion and capturing distance metrics from the motion sensor and was also found to be real time in sending the distance metrics to the Grafana interface for graphical visualisation. This prototype was effective at detecting and mitigating human-wildlife conflict and would lead to a reduction in cases in the long run if implemented at scale.

7.2: Recommendations

This IoT prototype is a sophisticated low-cost and accurate solution that will address human-wildlife conflict in Kenya and other parts of the world where human beings and wild animals coexist. It will aid conservation efforts and boost the tourism economy indirectly since wildlife numbers will increase due to reduced cases of revenge killings of animals by people. In order for this solution to have a maximum impact and benefit, it is recommended that this solution be deployed in at scale in parks and conservation areas under the jurisdiction of the government and Kenya Wildlife Service. From this research, it is recommended that the government, private sector and other entities contribute financially and by developing more technical expertise in IoT to implementing this technology and deploying this solution widely.

7.3: Future Work

The implemented prototype generated useful data stored as logs and S3 objects. This data at scale can be mined and analysed further to find deeper insights that can inform additional features that can be inculcated into the system or identify areas of improvement within the system. By making use of additional tools and techniques such as using longer range sensors and integrating the prototype with existing large scale animal monitoring systems, the prototype can be enriched and encompass more use cases. This prototype can also be connected to existing implementations such as GPS to increase the scope of the data points that are aggregated by it for analysis. It can also be used in combination with other technologies such as camera traps and machine learning algorithms to encompass larger geographical areas and send stakeholders information that is better segmented and customised to different environments

References

Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017). Scrum Model for Agile Methodology. *International Conference on Computing, Communication and Automation* , 864-869.

Kenya National Bureau of Statistics. (2019). *2019 KENYA POPULATION AND HOUSING CENSUS* . Nairobi: KNBS.

Sarhan, A. (2019). *Cloud-based IoT Platform: Challenges and Applied Solutions* .

Jayson, E. A. (2016). *Assessment of human-wildlife conflict and mitigation measures in Northern Kerala.* Kerala Forest Research Institute.

Terada, K., Yoshida, E., & Ishibashi, K. (2019). Implementation of IoT Networks Based on MQTT for Wildlife Monitoring System. *2019 IEEE International Conference on Internet of Things and Intelligence System* (pp. 161-166). IEEE.

The Internet Society. (2015). *The Internet of Things: An Overview.*

Makindi, S. M., Mutinda, M. N., Olekaikai, N. K., Olelebo, W. L., & Aboud, A. A. (2014). Human-Wildlife Conflicts: Causes and Mitigation Measures in Tsavo Conservation Area, Kenya. *International Journal of Science and Research* , 1025-1031.

Singh, A. S., & Masuku, M. B. (2014). Sampling Techniques & Determination of Sample Size in Applied Statistics Research: An Overview. *International Journal of Economics, Commerce and Management* .

Schwaber, K., & Sutherland, J. (2013). *The Scrum Guide.*

Mauvais, G., Hodgkinson, S., & Young, D. (2015). *The Internet of Things for Protected Areas.* Smart Earth Network.

F. Viani, A. P., Giarola, E., Robol, F., Benedetti, G., & Zanetti, S. (2016). Performance assessment of a smart road management system for the wireless detection of wildlife road-crossing. *2016 IEEE International Smart Cities Conference (ISC2)*, (pp. 1-6). Trento.

Shivaram, S., Chaitra, Kshama, Sneha, & Supriya. (2016). Low Cost Alert System for Monitoring the Wildlife from Entering the Human Populated Areas Using IOT Devices.

International Journal of Innovative Research in Science, Engineering and Technology, 128-132.

Habib, A., Nazir, I., Fazili, M. F., & Bhat, B. A. (2015). Human-wildlife conflict-causes, consequences and mitigation measures with special reference to Kashmir. *Journal of Zoology Studies*, 26-30.

GSMA. (2014). *Understanding the Internet of Things.* GSM Association.

GSMA. (2020). *The Mobile Economy, 2020.* GSMA.

WWF. (2018). *WWF Tanzania Report.* Dar Es Salaam: WWF Tanzania.

Lopez Research . (2013). *An Introduction to the Internet of Things (IoT).*

Sayakkara, A., Dabare, P., Suduwella, C., Sandaruwa, D., Voigt, T., Hewage, K., & Chamath Keppitiyagama Zoysa, K. D. (2015). Listening to the Giants: Using Elephant Infra-Sound to Solve the Human-Elephant Conflict. *APNIC 42.*

Viani, F., Robol, F., Salucci, M., Giarolla, E., Vigili, S. D., Rocca, M., . . . Massa, A. (2013). WSN-Based Early Alert System for Preventing Wildlife-Vehicle Collisions in Alps Regions: From the Laboratory Test to Real World Implementation. *7th European Conference on Antennas and Propagation*, (pp. 1913-1916).

Hodgkinson, S., & Young, D. (2015). *The Internet of Things for Protected Areas: The Application of Innovative Technologies to Improve Management Effectiveness* . Smart Earth Network.

Mishra, D. K., Rathore, & Pandey, D. (2014). HUMAN-WILDLIFE CONFLICT AND WILDLIFE WATERSHED MANAGEMENT. *International Journal of Bioassays*, 3210-3213.

Liu, X., Yang, T., & Yan, B. (2015). Research on the Architecture of Wildlife Observation and Communication System. *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, 415-418.

Kulkarni, S., & Kulkarni, P. S. (2017). Communication Models in Internet of Things: A Survey. *IJSTE - International Journal of Science Technology & Engineering*, 87-90.

Edemacu, K., Kim, J. W., Jang, B., & Park, H. K. (2019). Poacher Detection in African Game Parks and Reserves with IoT: Machine Learning Approac. *2019 International Conference on Green and Human Information Technology (ICGHIT)*, (pp. 12-17).

Asamiah, N., Mensah, K., & Oteng-Abayie. (2017). *General, Target and Accessible Population: Demystifying the Concepts for Effective Sampling.* . The Qualitative Report.

Akhtar, I. (2016). Research Design. *Research in Social Science: Interdisciplinary Perspectives* .

Sharma, S., Sarkar, D., & Gupta, D. (2012). Agile Processes and Methodologies: A Conceptual Study. *International Journal on Computer Science and Engineering* , 892-898.

Lewa, S. K., Maluki, D. P., Vindevov, P. V., & Farah, D. I. (2017). ROOT CAUSES OF HUMAN-WILDLIFE CONFLICT AND ALTERNATIVE DISPUTE RESOLUTION METHODS: THE CASE OF ARABUKOSOKOKE FOREST, KENYA. *International Academic Journal of Arts and Humanities*, 25-36.

Surya, T., & Selvi, S. C. (2017). A Literature Review on Analysis of Cause and Impact of Human Wildlife Conflict. *2017 IEEE International Conference on Smart Technologies and Management or Computing, Communication, Controls, Energy and Materials (ICSTM)*, (pp. 455-459). Chennai.

Kachhoria, R., Varma, S., & Radhakrishna, M. (2019). A Case Study of a Remote Sensing Using WSN. *2019 IEEE TENGARSS*, 41-53.

Indushree, Navya, Nandinisridevi, & Nikitha. (2019). IoT Based Animal Harm Detection using Sensors by Creating an Alert. *International Journal of Engineering Research & Technology (IJERT)*, 1-3.

Susanto, A., & Meiryani. (2019). System Development Method with The Prototype Method. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH* , 141-144.

Kumar, S., Tiwari, P., & Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: a review. *Journal of Big Data*.

Lee, D. M., & Labinghisa, B. (2019). Indoor localization system based on virtual access points with filtering schemes. *International Journal of Distributed Sensor Networks*.

Kaswan, K. S., Singh, S. P., & Sagar, S. (2020). Role Of Arduino In Real World Applications. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, 1113-1116.

Ren, X., Sun, J., & Xing, Z. (2020). International Conference on Software Engineering . *Demystify Official API Usage Directives with Crowdsourced API Misuse Scenarios, Erroneous Code Examples and Patches*, (pp. 925-936). Seoul, Republic of Korea.

Ylonen, T. (2019). SSH Key Management Challenges and Requirements. *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)* (pp. 1-5). New Technologies, Mobility and Security .

Ilker, E., Abubakar, M. S., & Sunusi, A. R. (2016). Comparison of Convenience Sampling and Purposive Sampling. *American Journal of Theoretical and Applied Statistics*, 1-4.

Stratton, S. J. (2021). Population Research: Convenience Sampling Strategies . *Prehospital and Disaster Medicine* , 373-374.

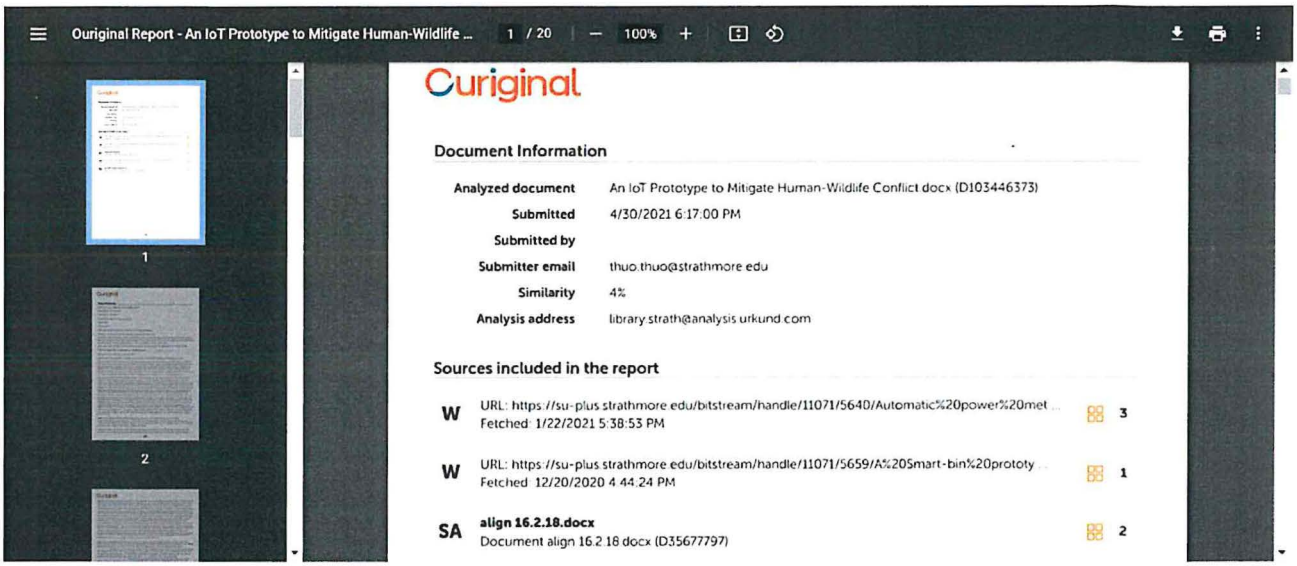
APPENDICES

Appendix A: Questionnaire

Please complete this questionnaire. The information requested is needed for academic purposes only and will be treated in strict confidence.

1. Do you have internet access?
    a) Yes ☐
    b) No ☐
2. Have you received reports of human-wildlife conflict in the last month?
    a) Yes ☐
    b) No ☐
3. If yes, how was it reported?
    a) Manual ☐
    b) Technological ☐
4. If manual, would you be interested in an automated solution?
    a) Yes ☐
    b) No ☐
5. If technological, which one among the list of technologies do you use? (tick all that apply)
    a) GPS data ☐
    b) Radio Triangulation ☐
    c) Drones ☐

Appendix B: Turnitin Report



Appendix C: Ethical Approval

Strathmore UNIVERSITY

18th October 2021

Mr Thuo Ng'ang'a Thuo,
thuo.thuo@strathmore.edu

Dear Mr Thuo,

RE: An IoT Prototype to Mitigate Human-Wildlife Conflict

This is to inform you that SU-IERC has reviewed and approved your above SU- master's research proposal. Your application reference number is **SU-IERC1187/21**. The approval period is **18th October 2021 to 17th October 2022.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-IERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-IERC within 48 hours of notification
iv. Any changes, anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-IERC within 48 hours
v. Clearance for export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days upon completion of the study to SU-IERC.

Prior to commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology and Innovation (NACOSTI) https://research portal.nacosti.go.ke/ and also obtain other clearances needed.

Yours sincerely,

for: Prof Fred Were,
Chairperson; SU-IERC

18 Oct 2021

Ole Sangale Rd, Madaraka Estate, PO Box 59857-00200, Nairobi, Kenya. Tel +254 (0)703 034000
Email admissions@strathmore.edu www.strathmore.edu