AN ENERGY EFFICIENT HIERARCHICAL APPROACH USING MULTIMEDIA AND SCALAR SENSORS FOR EMERGENCY SERVICES

A THESIS SUBMITTED TO

THE BOARD OF GRADUATE PROGRAMS

OF

MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS

BY

BURAK KIZILKAYA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR

THE

DEGREE OF MASTER OF SCIENCE

IN

THE

SUSTAINABLE ENVIRONMENT AND ENERGY SYSTEMS

JULY 2019

Approval of the Board of Graduate Programs

Prof. Dr. Gürkan Karakaş
Chairperson

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Ceren İnce Derogar
Program Coordinator

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science .

Assoc. Prof. Dr. Enver Ever
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Murat Fahrioğlu EEE / METU NCC

Prof. Dr. Doğu Arifler CMPE / EMU

ETHICAL DECLARATION

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Burak KIZILKAYA

Signature :

ABSTRACT

AN ENERGY EFFICIENT HIERARCHICAL APPROACH USING MULTIMEDIA AND SCALAR SENSORS FOR EMERGENCY SERVICES

KIZILKAYA, Burak
M.S., Department of Sustainable Environment and Energy Systems
Supervisor: Assoc. Prof. Dr. Enver Ever

July 2019, 87 pages

Recently, environment monitoring and detection systems became more accessible with the help of IoT applications. Furthermore, connecting smart devices makes monitoring applications more accurate and reliable. On the other hand, optimizing the energy requirement of smart sensors especially while transmitting data has always been very important, and there are different applications to create energy efficient IoT systems. Detailed analysis of lifetimes of various types of sensors (survival analysis) has therefore become essential. For the environment monitoring scenarios, with the help of smart multimedia sensors, more precise and accurate real-time information can be extracted. Video and audio sensors can be used as complementary mechanisms to have more accurate information. However, transmission of visual data is known to be one of the most costly operations for wireless multimedia sensor networks. To minimize energy consumption, visual data transmission should be minimized. In this thesis, a novel hierarchical approach is presented for emergency applications. Proposed framework makes use of multimedia and scalar sensors hierarchically to minimize the visual data transmission and in turn energy consumption. In addition, edge computing is introduced where lightweight machine learning algorithms are used for edge processing to prevent unnecessary data transmission. The heterogeneous sensor network architecture is applied within the domain of forest fire detection systems. Proposed framework is evaluated in terms of accuracy of detection as well as energy efficiency. The results are quite promising with validation accuracy of 98.20% and 29.94% energy saving compared to multimedia sensor based surveillance systems.

Keywords: WSNs, WMSNs, IoT, energy efficiency, heterogeneous network architecture, edge computing, machine learning

ÖZ

ACİL SERVİSLER İÇİN MULTİMEDYA VE SKALER SENSÖRLERİ KULLANARAK ENERJİ VERİMLİ HİYERARŞİK YAKLAŞIM

KIZILKAYA, Burak
Yüksek Lisans, Sürdürülebilir Çevre ve Enerji Sistemleri Bölümü
Tez Yöneticisi: Doç. Dr. Enver Ever

Temmuz 2019 , 87 sayfa

Son zamanlarda, IoT uygulamalarının yardımıyla çevre izleme ve algılama sistemleri daha erişilebilir hale gelmiştir. Ayrıca, akıllı cihazların bağlanması izleme uygulamalarını daha doğru ve güvenilir kılar. Öte yandan, akıllı sensörlerin enerji ihtiyacını özellikle veri aktarırken optimize etmek araştırmacılar arasında önemli bir konudur ve enerji verimli IoT sistemleri oluşturmak için farklı uygulamalar vardır. Bu nedenle, çeşitli sensör tiplerinin ömrünün ayrıntılı analizi (hayatta kalma analizi) önemli hale gelmiştir. Ortam izleme senaryoları için, akıllı multimedya sensörleri yardımıyla, daha kesin ve doğru gerçek zamanlı bilgiler elde edilebilir. Video ve ses sensörleri daha doğru bilgiye sahip olmak için tamamlayıcı mekanizmalar olarak kullanılabilir. Bununla birlikte, görsel verilerin iletimi, kablosuz multimedya sensör ağları için en pahalı işlemlerden biri olarak bilinir. Enerji tüketimini en aza indirmek için görsel veri aktarımı en aza indirilmelidir. Bu tez çalışmasında acil durum uygulamaları için yeni bir hiyerarşik yaklaşım sunulmaktadır. Öngörülen çerçeve, görsel veri iletimini ve buna bağlı olarak enerji tüketimini azaltmak için hiyerarşik bir yaklaşımla multimedya ve sayıl sensörlerden faydalanır. Ek olarak, gereksiz veri aktarımını önlemek amacıyla hafif (kompleks olmayan) makine öğrenme algoritmaları kullanılarak kenar hesaplama sistemi önerilmiştir. Heterojen sensör ağ mimarisi, orman yangını algılama sistemleri alanında uygulanmıştır. Önerilen çerçeve, saptama doğruluğu ve enerji verimliliği açısından değerlendirilmiştir. Sonuçlar, %98.20 orman yangını tespit doğruluğu ve multimedya sensör tabanlı gözetim sistemlerine göre %29.94 enerji tasarrufu ile oldukça umut vericidir.

Anahtar Kelimeler: WSNs, WMSNs, IoT, enerji verimliliği, heterojen ağ mimarisi, kenar hesaplama, makine öğrenmesi

To my mother, the most devoted, and kindest woman I have ever seen,
with endless love

ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to my supervisor Assoc. Prof. Dr. Enver Ever for his continuous support and guidance throughout my MSc studies and research. His warm and friendly attitude, and expert advises always inspired me to make right decisions in the most important moments of my life. I could not have imagined better supervisor and mentor for my MSc studies.

Many thanks to jury members, Prof. Dr. Doğu Arifler, Assoc. Prof. Dr. Murat Fahrioğlu, and Assoc. Prof. Dr. Enver Ever for their assistance and valuable comments.

My sincere thanks also goes to Prof. Dr. Ali Cevat Taşıran for his support and guidance both in my life and my MSc studies. It is an honor to be with a person with experience like him. His advises are always decisive in my life.

My grateful thanks also goes to Prof. Dr. Adnan Yazıcı for his encouragement and help. It is a pleasure for me to be a part of his projects and to have a chance to work with him.

Many thanks to all the professors in Computer Engineering program for all their support and every single thing that they taught me : Assoc. Prof. Dr. Yeliz Yeşilada, Assoc. Prof. Dr. Enver Ever, Prof. Dr. Fadi Alturjman, Assoc. Prof. Dr. Islam Elgedawy, Dr. İdil Candan, Assoc. Prof. Dr. Okan Topçu.

My special thanks goes to my family - my grandfather, Mustafa Tebrekan; my grandmother Radiye Tebrekan; my mother Betül Kızılkaya; my brother Buğra Kızılkaya, and my uncle, Ahmet Tebrekan. They are always supportive and beside me. This achievement was not possible without them. My warmest thanks goes to Duygu Madenci for her patience and love. She is always with me in every single moment of my life.

Many thanks to all friends and teaching assistants at METU NCC. I also want to thank to my colleague Hakan Yekta Yatbaz for his support and help in my thesis writing process. I am happy to have such a good friend.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

WSN	Wireless Sensor Networks
WMSN	Wireless Multimedia Sensor Networks
IoT	Internet of Things
QoS	Quality of Service
ICT	Information and Communication Technologies
UAV	Unnamed Aerial Vehicles
UAS	Unnamed Aerial Systems
MCU	Microcontroller Unit
OS	Operating Systems
PC	Personal Computer
PDA	Personal Digital Assistant
API	Application Program Interface
RTOS	Real-Time Operating System
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
OTcl	Object oriented Tool Command Language
RPL	Routing Protocol for Low Power and Lossy Networks
6LoWPAN	IPv6 over Low Power Wireless Personal Area Networks
IDE	Integrated Development Environment
GUI	Graphical User Interface
CCD	Charge-Coupled Device
IR	Infrared
AVHRR	Advanced Very High Resolution Radiometer
NOAA	National Oceanic and Atmospheric Administration

MODIS	Moderate Resolution Imaging Spectroradiometer
GPIO	General Purpose Input/Output
UART	Universal Asynchronous Receiver/Transmitter
IP	Internet Protocol
LTE-M	Long Term Evolution-Machines
LCD	Liquid Crystal Display
GSM	Global System for Mobile
SMS	Short Message Service
ANN	Artificial Neural Networks
CNN	Convolutional Neural Networks
SVM	Support Vector Machines
KNN	K-Nearest Neighbor

CHAPTER 1

INTRODUCTION

1.1 Wireless Sensor Networks

Wireless sensor networks (WSN) technology is one of the most promising in our era particularly for fully autonomous systems and internet of things applications. Sensor is a device which consists of sensing units, processing units, and communication units. A sensor network is composed of large number of sensor nodes. There are many application areas including military, environment, health, etc. [4]. Generally, wireless sensors have the ability to make basic preprocessing with their processing units. They are not capable of doing too complex processing, yet it is sufficient to use their limited processing power to enable edge computing and decreasing consumed energy by transmission of data. There are various types of sensors to enable wide range of applications. Transmission of data is generally the most energy consuming process compared to sensing, or processing raw data. To decrease transmission rate, it is important to preprocess the raw data and and minimize the amount of transmission while the application requirements are still fulfilled. Energy efficiency in WSNs is one of the main constraints of the system since wireless sensors run on battery. As discussed in [5], there are many strategies proposed to minimize energy consumption and maximize lifetime of a sensor node. For example, by applying energy harvesting techniques to WSNs, it is possible to have infinite lifetime WSNs. Moreover, energy efficiency can be achieved by applying energy efficient routing algorithms like selecting the route whose nodes have maximum amount of energy, clustering algorithms which selects cluster heads in an energy efficient way to maximize lifetime of network. In addition, there are various energy efficient mechanisms which are discussed in [1] and classified in Table 1.1. One of the most important approach to decrease energy consumption is radio optimization since the radio module of the sensor is the most energy consuming module. In WSNs, scalar data are transmitted such as temperature, humidity, light, etc. Since the scalar data transmission is not too costly compared to multimedia data, the sensors in WSNs are deployed densely to the environment to enable multihop sensor networks which increases energy efficiency since sensor nodes are close to each other and transmission power is low. It is easier to achieve more energy efficient systems by making use of scalar sensors. On the other hand, the accuracy of the system may not be sufficient in some critical applications where it is very crucial to have real-time connection over wireless networks for real-time monitoring and control such as medical applications (e.g. wireless patient monitoring), and environment monitoring applications for disaster management (e.g. forest fire detection, flood detection, etc.) [6].

Radio Optimization	**Data Reduction**	**Sleep/Wakeup Schemes**	**Energy Efficient Routing**	**Battery Repletion**
Transmission Power Control	Aggregation	Duty-cycling	Cluster Architectures	Energy Harvesting
Modulation Optimization	Adaptive Sampling	Passive wakeup Radios	Energy as a Routing Metric	Wireless Charging
Cooperative Communication	Compression	Scheduled Based MAC Protocols	Multipath Routing	-
Directional Antennas	Network Coding	Topology Control	Relay Node Placement	-
Energy Efficient Cognitive Radio	-	-	Sink Mobility	-

Table 1.1: Energy efficient mechanisms [1]

1.2 Wireless Multimedia Sensor Networks

With the development in hardware industry and technology, low-cost multimedia hardware became more and more accessible and commonly used. Availability of multimedia devices in sensor networks enlarged the area of applications in sensor networks and Internet of Things (IoT) application areas by increasing the accuracy of the systems using multimedia sensors. A multimedia sensor is a sensor device which measures and transmits multimedia data such as still images, videos, and audios. Wireless Multimedia Sensor Networks (WMSNs) consist of multimedia or both scalar and multimedia wireless sensors. WMSNs enable more realistic data retrieval and in turn contribute to development of new image processing and machine learning techniques since huge amount of data can be retrieved in various type of applications [7], [8]. There are different types of applications including surveillance systems, traffic and transportation, advanced health care delivery, environmental monitoring, localization systems, etc. [9], [10], [11], [12]. One of the main constraints in WMSNs is energy consumption since the delivery of multimedia data is costly compared to scalar data transmission in WSNs. Sensor devices are constrained not only in terms of battery but also in terms of memory and processing power. Since this is the case, it becomes crucial to use resources of sensors efficiently. To overcome resource constraints of WMSNs, there are many approaches which aim to use resources efficiently to maximize network lifetime as well as to keep the level Quality of Service(QoS) in an acceptable range for end user.

1.3 Sustainability Perspective

Environmental sustainability is a popular research area among researchers and the impact of Information and Communication Technologies (ICT) is discussed to understand whether ICT applications contribute to environmental sustainability or not. According to the study in [2], 2% of global carbon emission is because of ICT devices and detailed shares of different categories are given in Figure 1.1. According to [2], ICT can be a low-carbon enabler when we consider applications such as use of e-mail, e-commerce, e-banking, etc. It is proposed that ICT can help to decrease greenhouse gas (GHG) emission by 16.5% by 2020 which means that $1.9 trillion in gross energy and fuel saving. On the other hand, ICT can be a power drainer since the number of mobile devices, personal devices are increasing dramatically which is expected to increase carbon emission by 72% by 2020 [13]. In conclusion, It is possible to make ICT applications as low-carbon enabler by proposing energy efficient and environment friendly mechanisms and applications.

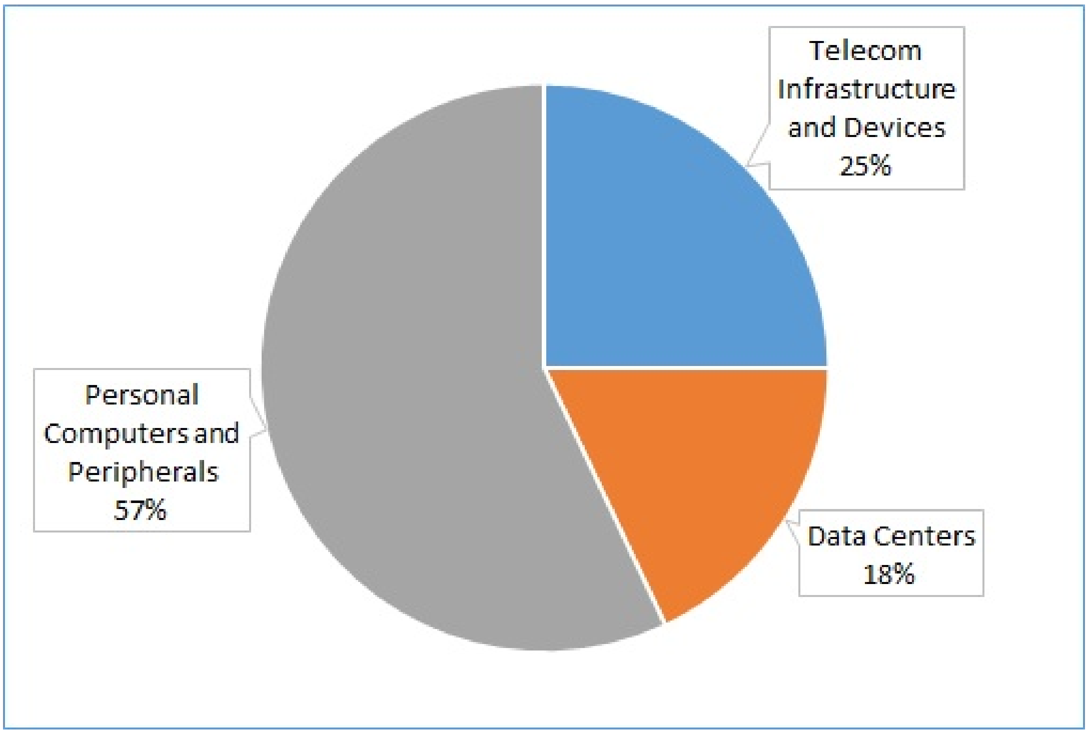


Figure 1.1: ICT carbon emission shares [2]

The effects of wildfire caused disasters can also be very significant in terms of sustainability and the potential damage on natural life itself. Recent studies on environmental disasters, especially wildfires, show the importance of early detection. For example, the wildfire in Athens where at least 90 people died, at least 164 adults and 23 children have been injured [14], was a recent event which showed the potential of similar catastrophes. Similarly, California had another wildfire in summer 2018 which was the largest in California history in which 185,800 hectares of forest, which is very important for the environment, was burned and a firefighter died [15]. Following the California wildfires, the US government and local authorities (California Forest Management) had a long lasting debate on the reasons of late detection and causes of the large scale uncontrollable wildfire. Unfortunately, the situation is similar in Asia as well. The number of fires in Kazakhstan is not negligible and it has increased by 41% according to the Ministry of Emergency Situations [16]. Considering the damage caused by the wildfires on environment and living species, it is quite obvious that a sensory information based early detection system can lend itself as a useful tool. Of course efficiency and accuracy of these surveillance systems should be studied.

1.4 Objectives of the Thesis

The main aim of this study is to present energy efficient and accurate framework for emergency applications by using energy efficient scalar sensors and accurate multimedia sensors. Since forest conditions are not suitable for continuous maintenance, it is necessary to introduce systems which are self configurable, cheap, easily deployable, energy efficient, and accurate. In addition, accuracy is another important metric since emergency applications are considered. In this thesis, an energy aware framework is presented for wildfire detection in forests while the lifetime of multimedia sensor nodes are prolonged. The detection accuracy of the system is considered carefully and evaluated using a data set explicitly established for this particular project. To achieve objectives of this thesis, following tasks are considered :

- Studying existing emergency applications in the area of wireless sensor networks and wireless multimedia sensor networks.
- Studying existing energy efficient approaches applied to wireless sensor networks and wireless multimedia sensor networks.
- Studying hardware platforms used in similar applications.
- Studying specifications, energy consumption characteristics of IoT hardware platforms.
- Studying machine learning approaches applied to similar applications.
- Studying statistical studies in the literature related with low-power sensor nodes.
- Proposing energy efficient and accurate framework for emergency applications.

- Developing simulations as well as real time test beds for evaluation of the proposed system.
- Proposing lightweight Convolutional Neural Networks (CNN) model to enable edge computing and have more accurate system.
- Conducting statistical analysis to investigate factors that affect the lifetime of low-power sensor nodes.

1.5 Thesis Contribution

In this study, an energy efficient hierarchical approach is proposed using multimedia and scalar sensors for emergency applications. Considering existing studies in the literature, most of the studies focus on homogeneous sensor networks [17]. In this work, a novel framework is proposed which consist of heterogeneous sensor nodes. Our proposed framework achieved approximately 29% efficiency in terms of energy.

To achieve energy efficiency objective, hierarchy, heterogeneity, and edge computing paradigms are introduced and applied. By applying hierarchy, scalar sensors are used in the first level of detection. Multimedia sensors are used if and only if triggered by scalar sensor measurements. In addition, heterogeneity (i.e. using both scalar and multimedia sensors) helps to improve both efficiency and accuracy of the system. Edge computing, on the other hand, prevents unnecessary data transmission, in turn, increases efficiency of the system since the most energy consuming activity is data transmission. By decreasing transmission rate, traffic load of wireless channel is also reduced which leads to increase in packet reception rate since the probability of collision is low.

To achieve accuracy objective, a lightweight CNN model is proposed and tested. To train and test the model, a new image dataset is established. According to the numerical test results, approximately 98% accuracy is achieved and validated by validation data set which is used to tune the parameters of a classifier. Proposed lightweight CNN model contributes to energy efficiency as well by enabling edge computing since it prevents unnecessary data transmission. In addition, Quality of Service (QoS) related features are improved by reducing the traffic load through selective transmission. Since the data transmission rate is event triggered and not periodic, packet reception rate is increased and packet loss rate caused by interference is decreased.

1.6 Thesis Outline

In this thesis, energy efficient hierarchical approach is proposed for emergency applications. As a case study, early forest fire detection is implemented.

Background information about WSNs, WMSNs, application areas, hardware platforms, operating systems, and some well known simulation tools are discussed in Chapter 2.

In Chapter 3, existing studies are discussed in detail in the area of forest fire detection, energy efficient monitoring, machine learning approaches for forest fire detection, and statistical analysis of sensor lifetime. In addition, research gaps are investigated and comprehensive categorization is presented.

In Chapter 4, statistical analysis is conducted for low-power sensor motes. Results of analysis are discussed and future research directions are given.

Proposed energy efficient framework is discussed and explained in Chapter 5. System design, overall scenario, and used sensor types are also explained.

Evaluation of the system using simulation and test bed implementation is performed in Chapter 6. All simulation and test bed implementation results are discussed and presented comparatively.

In Chapter 7, a lightweight Convolutional Neural Network (CNN) model is proposed for forest fire detection with the newly created dataset. Test results of the proposed model are also discussed in the chapter.

Chapter 8 concludes the study by giving conclusion remarks and possible future research directions.

CHAPTER 2

BACKGROUND

2.1 Applications of WSNs

2.1.1 Military Applications

One of the most important application areas of WSNs is military applications. Communication in military is very crucial and WSNs play important role because of their capability of real time transmission and control over wireless links. It is also advantageous to use WSNs since they offer low-cost deployment, robustness, fault tolerance, reliability since there is always risk of enemy attacks in military applications [18]. Sensors can detect various type of data such as gas, waves, light, pressure, sound etc. where it is crucial to detect explosive chemicals or materials. Also, sensors can be used to detect moving objects and presence of people in the areas where high security is vital [19], [20].

2.1.2 Healthcare Applications

WSNs are commonly used in the area of healthcare since the number of elderly people increased in last few decades [21]. In addition, wireless sensor solutions are important to decrease the cost of care for chronically ill or disabled people. It enables remote monitoring of ill people, remote control of home appliances and even create applications for little children or baby care for working parents. Early detection for medical emergencies, activity recognition for elderly people, control over wireless links, etc. are other application areas of healthcare in WSNs [22]. Healthcare applications are very popular among researchers and there are various novel approaches and mechanisms to enable secure and reliable healthcare by using WSNs and IoT paradigm [23], [24], [25], [26]. On the other hand, it is crucial to consider some challenges in healthcare applications like security, reliability of the system, privacy of patient data, confidentiality, and data integrity [27].

2.1.3 Environmental Applications

With the help of wirelessly connected, low-cost, low-power, and small size sensor nodes, automation and monitoring in environmental applications became more and more popular. WSNs assist our daily life by monitoring environment related phenomena such as temperature, humidity, light etc. They are also used for automation and remote control purposes [28]. Agricultural monitoring is very common and is mostly used on farming areas. It includes monitoring of air conditions, soil conditions, and monitoring of poultry [29], [30], [31]. Habitat monitoring is another popular area of environmental applications. Water quality monitoring, plant and animal monitoring, pollution monitoring in ecological areas, forest monitoring, and behaviour monitoring of some species like seabirds are some application areas of habitat monitoring [32], [33], [34], [35].

2.1.4 Home Applications

Home automation and monitoring is another important area of WSNs. Smart homes, smart cities, smart environments, and many related applications became available with the help of developing technologies in the area of sensors, actuators, and wireless communication. Home applications include light control, valve control, door and window control, remote control over wireless links, smart energy applications like smart grids, automation of electrical devices, remote care, safety, and security [36]. There are many new research directions in home applications. It is possible to create smart homes which share information and data with outside world. However, it creates the concern of security and confidentiality. In addition, smart grid applications are also very popular where an intelligent electricity system create communication between supplier and consumer [37].

2.2 Applications of WMSNs

2.2.1 Surveillance Applications

Surveillance applications need high accuracy, reliability, and more detailed information from the environment to stream and report the phenomenon. In such crucial applications like thefts, car accidents, traffic violations, etc. traditional WSNs and monitoring environment using scalar data become insufficient. Multimedia sensors which are capable of gathering multimedia data such as video, audio, and instant images are used as complimentary of existing surveillance applications to increase the accuracy of the system. Surveillance applications are one of the most popular application areas in WMSNs. Since it is possible to transmit video or audio data in WMSNs, it also makes possible to use more complex techniques like image processing, signal processing to gather more valuable information from transmitted data. These applications can be used for several purposes such as identifying the person, detecting anomalies, detecting fires, etc. [7], [38].

2.2.2 Traffic Monitoring Applications

The increase in number of vehicles and complicated traffic conditions cause huge amount of money and time consumption. Traffic monitoring applications are widely used with WMSNs technology. Now, it is possible to establish intelligent traffic monitoring systems which are capable of collecting information, data fusion, making decision, etc. They improve safety and efficiency of transportation systems, decrease fuel consumption, time consumption, carbon emission, and number of accidents. Current technology of WMSNs is very useful in emergency situations such as fire and accident. Intelligent systems can provide priority to emergency cars in the traffic like ambulance, police car, and firefighters [39]. Unnamed aerial vehicles (UAVs) can also be used to monitor traffic similar to the applications presented in [40]. Also, low-cost sensor motes with attached cameras are popular since energy consumption is one of the main constraints of WMSNs [41], [42].

2.2.3 Personal and Healthcare Applications

It is possible to implement more realistic and accurate personal and health care applications by using multimedia sensors, therefore WMSN applications in this area are also becoming quite popular [43]. Embedded sensor technology and use of smart phones enlarged the area of assisted living and e-health. There are many applications and monitoring techniques by using embedded sensors on smart phones, smart watches and other wearable devices [44]. Multimedia contents such as audio using smart phones microphone, video streams using cameras of smart phones, etc. are used to monitor, diagnose, and recognition purposes. Sleep apnea monitoring using microphone [45], analyzing skin images to identify cancer using camera of smart phone [46], detection and recognition of melanoma(type of skin cancer) [47], [48] are some examples of e-health applications with WMSNs.

2.2.4 Environmental and Industrial Monitoring Applications

Environmental monitoring has vital role in the areas of climate monitoring, habitat monitoring, agricultural and natural systems monitoring, and natural disaster monitoring and diagnosis. Different from the scalar monitoring like monitoring temperature, humidity, light, gas, etc., with use of cameras, microphones, especially UAVs, novel approaches are proposed. Unnamed Aerial Systems (UAS) make it possible to monitor considerably large scales with high quality and accuracy. They provide more detailed monitoring on wild areas and habitats as well [49].

2.3 Hardware Platforms

WSN based systems are the main source of sensor node activities and an integral part of IoT. They can be considered as the key technology for IoT, since they support the applications through the main infrastructure which consists of numerous sensors working collaboratively to monitor event occurrences in a given habitat [50].

Since there are various applications of WSNs, WMSNs, and IoT, different type of hardware platforms are needed for both industrial and academic purposes. The development in the area of micro-controllers, transceivers, and sensors helps the process of creating low-power, small size and low-cost hardware which can be used in various type of applications. IoT technology mainly makes heterogeneous devices connected via Internet to create interconnected systems with different properties and purposes. Generally, the IoT devices consist of sensing units, processing units, memory, and a battery. However, considering different application areas, different type of hardware devices are developed and used [27]. For example, while it is sufficient to use devices which have limited processing capacity in certain types applications, in some other types, it is better to use high computing power devices or so called sensor motes since implementation of edge computing has potential to significantly contribute to resource optimization. Existing sensor motes can be classified according to their microcontroller unit (MCU), memory, and transceiver types. Some well known and commonly used hardware are listed in Table 2.1. The model of MCU, memory specifications and transceiver versions are provided.

Sensor Mote	MCU	Memory	Transceiver
BTnode[51]	Atmega 128L	4KB RAM, 128KB Flash, 4KB EEPROM	CC1000
Mica2[52]	Atmega 128	4KB RAM, 128KB Flash, 512KB EEPROM	CC1000
MICAZ[53]	Atmega 128	4KB RAM, 128KB Flash, 512KB EEPROM	CC2420
TelosB[53]	MSP 430F 1611	10KB RAM, 48KB Flash, 1MB EEPROM	CC2420
Kmote[54]	MSP 430	10KB RAM, 48KB Flash, 1MB EEPROM	CC2420
XM1000[55]	MSP 430F 2618	8KB RAM, 116KB Flash, 1MB EEPROM	CC2420
TSmoTe[56]	ARM Cortex M3	96KB RAM, 1MB Flash	ZigBee, Wi-Fi, IEEE802.15.4
IRIS[57]	MSP 430F 1611	10KB RAM, 48KB Flash, 1MB EEPROM	CC2420
WisMote[58]	TI MSP430	16KB SRAM, 256KB Flash, 8MB EEPROM	CC2520
WiSense[59]	MSP- 430G-2955	10KB RAM, 48KB Flash, 128KB EEPROM	CC2520
LOTUS[60]	ARM Cortex M3	64KB SRAM, 512KB Flash	802.15.4 Radio
Node+[61]	9-axis motion engine	16MB of onboard storage	Bluetooth 4.0
Infini-Time[62]	MSP 430 FR5969	2KB RAM, 64KB Flash	M24LR-16ER
Mago-Node+[63]	ATmega 256RFR2	32KB RAM, 256KB Flash 8KB EEPROM	CC2530

Table 2.1: Some well known IoT hardware platforms

In this study, AS-XM1000 802.15.4 new generation mote modules which are based on “TelosB” technical specifications are used as scalar sensors [64]. XM1000 sensor motes are easily configurable and they come with a widely used transceiver which has built in libraries for various operating systems as well as simulation tools. For example in Castalia simulation tool there is a built in parameter file for CC2420 transceiver and Contiki OS supports XM1000 hardware platform with built in libraries which can easily run on XM1000. In addition, it is widely used and researched well in the area of WSNs as presented in studies [65], [66], [67], [68], [69]. Considering these advantages of XM1000 in terms of easy configuration, easy deployment, simulation support, and operating system support, it is selected to be used in test bed implementation of the proposed framework.

2.4 Operating Systems(OS)

IoT devices are very limited in terms of memory, processing power and energy as discussed earlier. In addition, there are various number of hardware and heterogeneous system to support. In this sense, traditional operating systems like Windows and Linux are not suitable because of hardware constraints of IoT devices. IoT OS should consider some requirements. One of them is small memory footprint since IoT devices typically provides kilobytes of memory. In addition, OS should support heterogeneous hardware since there are different types of hardware and related platforms as summarized in Table 2.1. Another important requirement of IoT OS is energy efficiency considering that most of the IoT devices run on battery and have limited lifetime. OS should be energy aware to maximize the lifetime of devices. Moreover, security, real-time capabilities, and network connectivity are also important requirements of IoT OS [70].

2.4.1 Contiki OS

Contiki is an OS which is developed for very low memory devices like 8-bit MCUs. Later on, it was further improved to run on 16-bit and 32-bit IoT devices. It is based on event-driven programming which also supports multithreading. Contiki OS is written using C, yet it uses macro-based abstractions like protothreads [71]. It is an open source platform where various number of versions exist since it is widely used for research and industrial purposes. It is lightweight and portable [72]. Contiki supports many different communication protocols such as IPv4, IPv6, uIP (TCP/IP protocol stack for 8-bit MCUs), Rime (another lightweight layered protocol stack) which provides single hop unicast, single hop broadcast, and multihop communication. It also provides implementation of IPv6 routing protocol RPL for lossy networks [73]. Contiki has a file system which is called Coffee file system. It provides lightweight and efficient storage abstractions since traditional storage systems are too complex and not widely used in WSNs. Coffee is very useful for network layer components such as routing tables and packet queues [74].

2.4.2 Tiny OS

Tiny OS is the most used operating system in IoT hardware(limited memory) with Contiki OS [73]. It is written with one of the C dialects which is nesC. It runs on 8-bit and 16-bit MCU. It supports very complex programs with very low memory requirements. It is also important that it supports low power applications. It has BLIP (Berkeley Low-power IP stack) network stack which implements 6LoWPAN (IPv6 Low-power Wireless Personal Area Networks) network stack. There are three main abstractions in Tiny OS which are commands, events, and tasks. Command is the request to any component such as sensor to receive any type of service. Events on the other hand can be defined as the completion signal of the service which is initialized by command. Lastly, tasks are the functions which are executed by Tiny OS.[75].

2.4.3 MANTIS OS

MultimodAl system for NeTworks of In-situ wireless Sensors (MANTIS) is a cross-platform embedded operating system for WSNs. It is an energy efficient OS which fits within less than 500 bytes of RAM memory which includes kernel, scheduler, and network stack. It also uses power-efficient scheduler to achieve energy efficiency by decreasing consumption of current to the μA levels. MANTIS OS is very flexible which supports cross-platform and testing on PDAs, PCs, and different MCUs. It supports a simple C API which actually enables cross-platform support [76].

2.4.4 Nano-RK OS

Nano-RK is a real-time operating system (RTOS) which works reservation-based. Each task is created with priority and priority based preemption is applied. For communication purposes, it supports ad-hoc multihop wireless networking by providing port based socket abstraction. It provides energy efficiency by enforcing limits on memory and energy usage of individual applications. It is also possible to use CPU and network bandwidth reservation in Nano-RK OS which are guaranteed by operating system itself. It provides some tools to estimate energy consumption of each individual application and in turn the lifetime of the system [77].

In this study, Contiki OS is used to implement real life experiment of the proposed framework. It is a lightweight operating system which fully fits to hardware that is used in this study. Contiki OS specifically supports the XM1000 mote with built in applications and libraries. In addition, it is an open source operating system with a huge community support. There are various sources for Contiki where developers and researchers can easily refer to and benefit from. The main advantage of Contiki is that it is specifically created for low-power wireless sensor networks with support of lightweight communication protocols such as uIP, Rime, and RPL. Memory management is also considered for IoT hardware where the main constraint is limited resources. It has its own file system Coffee which provides lightweight and efficient storage abstraction compared to traditional file systems. Considering various advantages of Contiki OS, it is used to implement real life experiment of the proposed framework with real hardware.

2.5 Simulation Tools

IoT and sensor networks areas are very dynamic. New algorithms, protocols, and new techniques are proposed to provide more energy efficient systems and frameworks. Proposed novel approaches require testing to be able to understand the dynamics included and evaluate the efficiency and QoS. However, implementing real life large scale systems for these purposes is costly and time consuming. Also, it is not flexible to run again and again same scenarios with different scales. Researchers use simulations and emulations instead of costly and time consuming real life implementations to test their new algorithms, approaches, or mechanisms [78]. In this section, some well known and commonly used simulation tools for IoT and sensor network applications are discussed.

2.5.1 Network Simulation Version 2 (NS-2)

Network Simulation Version 2 or NS-2 is created in 1989 and it grew with the help of research community and their contributions throughout years. It is one of the most popular network(both wired and wireless) simulation tool. NS-2 is an event-driven simulation which provides wide range of network protocols including routing algorithms, TCP, UDP, etc. It is very useful to study and research the dynamic nature of WSNs [79]. It is also known as object oriented discrete event simulator since it is totally based on object oriented programming. It mainly consist of C++ and OTcl (Object oriented Tool Command Language) as programming languages [80].

2.5.2 Network Simulation Version 3 (NS-3)

Similar to NS-2, NS-3 is a discrete event simulator. The main development objective of NS-3 is contribution to communication research. It is an open source simulation tool. On the other hand, NS-3 is not a new version of NS-2 where it does not support any API of NS-2. In contrary to NS-2, it is written in pure C++. Optional python bindings are also supported. There are various WSNs modules in NS-3 such as 802.15.4, RPL, 6LoWPAN, etc. It is suitable with almost all operating systems like Linux and Windows [81].

2.5.3 OMNeT++

Objective Modular Network Tested in C++ (OMNeT++) is an extensible, component-based, flexible C++ based simulation tool and framework. Primary objective is building network simulations. It provides support for sensor networks, wireless ad-hoc networks, internet protocols, performance modelling, etc. OMNeT++ mainly consists of simulation kernel library of C++, the NED topology description language, simulation IDE based on Eclipse, simulation run-time GUI (Qtenv), command line interface, documentations, and sample simulations. Simulation kernel runs on all platforms which have a modern C++ compiler. However, simulation IDE requires Windows, Linux, or macOS. It is distributed under academic public licence [82].

In this study, OMNeT++ based Castalia simulator is used to simulate the proposed framework. Castalia is generally used for low power sensor devices and body area networks. It is quite popular in the area of WSNs and it is widely used in various recent WSNs studies [83], [84], [85], [86], [87], [88]. Radio models are fully implemented in Castalia which are based on real low-power radios with different levels of TX power and different power consumption levels according to simulated hardware. Various MAC and routing protocols are available as well. It also supports the mobility of sensor node. It is quite flexible with parameter file (.ini file) where many features can be set such as simulation time, deployment area, power consumption of nodes, TX output power, MAC protocol, routing protocol, radio module, packet transmission rate, and data payload. It gives flexibility to simulate various sensor motes by setting these parameters. Considering discussed advantages of Castalia, it is used to simulate the proposed framework by setting parameters which are based on real hardware used in test bed implementation.

CHAPTER 3

RELATED WORKS

3.1 Forest Fire Detection

Forest fires are most hazardous disasters in dry areas considering damage caused by wildfires. The forest fire in Athens in summer 2018 [14] is one of the most recent examples. Unfortunately in this disaster, 90 people died, 187 people were injured and more than 1,000 buildings were destroyed. California had another wildfire in summer 2018 which was the largest in California history where 185,800 hectares were burned and a firefighter died [15]. These recent examples show that forest fires are very dangerous both for the living species and environment. Since 30% of carbon dioxide comes from forest fires, the local weather patterns are also affected in long term. This in turn cause extinction of rare species as well [89]. Because of its consequences, various monitoring and detection techniques are proposed, tried, and used over years. Finding solutions and countermeasures to forest fires are also very popular among researchers where serious projects are conducted on this area. In the past, forest fires are detected with human observations. People built lookout towers at high attitude areas or points to detect forest fires. These towers had lookout personnel who were responsible for observing the area. However, lookout tower technique had very bad working conditions for lookout personnel and its accuracy is open for discussion since it relies on human observation [90]. Insufficiency of human observation led to development of video surveillance systems. CCD (Charge-Coupled Device) cameras and IR (Infrared) detectors are used in most of the video surveillance systems. These detection devices are installed on towers to enable monitoring. In case of emergency, automatic surveillance system alerts the fire department. However, the main concern of these systems is accuracy since they are affected by weather conditions such as fog, clouds, smoke from other activities. In addition it is almost impossible to deploy automatic video surveillance systems to large scale forests because of wild nature of forests and cost of deployment [91].

Satellite based forest fire detection is another advanced detection method. Satellite based detection starts with two main satellites launched for forest fire detection. AVHRR (Advanced Very High Resolution Radiometer) was launched in 1998 by NOAA (National Oceanic and Atmospheric Administration). The main purpose of AVHRR is monitoring clouds and thermal emission of the Earth [92]. In 1999, MODIS (Moderate Resolution Imaging Spectroradiometer) was launched by NASA for the purpose of capturing cloud dynamics and surface radiation [93]. Current forest fire detection systems which are based on satellite images, use AVHRR and MODIS to gather Earth images and monitor forests to detect forest fires. The main problem with satellite based surveillance systems is accuracy. Using satellite images, surveillance system can detect minimum 0.1 hectares size of fire with the accuracy of 1 km [94]. In addition, AVHRR and MODIS provide complete Earth images every one or two days. Considering speed of wildfires and importance of accuracy in detection, satellite based surveillance systems are not sufficient enough for early detection of forest fires. They cannot be used as a surveillance tool in such applications where time is one of the most important metric. In addition, satellite images are vulnerable to weather conditions such as clouds, rain, and fog. Since accuracy and real-time detection are very crucial for forest fire detection and surveillance, satellite based surveillance systems are not suitable. Because, to minimize the scale and damage of the disaster, it is vital to have a system with an immediate response [95].

Considering constraints of traditional methods, video surveillance systems, and satellite based surveillance for forest fire detection, one of the most promising technologies is wireless sensor networks with IoT enabled applications. As discussed in previous chapters, WSNs technology can be applied to various areas such as health, military, environment monitoring, transport system, smart homes, smart cities and so on. Combination with IoT enlarges the application areas even more. Especially, in disaster applications like forest fire detection where real-time communication and control are much more important, WSNs offer more accurate, energy efficient, and low-cost solutions. It is possible to monitor forest areas by measuring various phenomena such as temperature, humidity, gas, smoke which can be very helpful in forest fire detection application. Multimedia data such as audio, video, still images can also be gathered using so-called multimedia sensors which are equipped with cameras and microphones. Easy and low-cost deployment (e.g. with aeroplanes) of small sensors is another advantage of WSNs. Especially in large scales, WSNs can be deployed densely and there is no need for manual setup and configuration thanks to recent self-configuring network protocols and mechanisms.

There is a wealth of literature on environment especially forest fire monitoring applications using WSNs and IoT paradigm. For example in [96], authors propose a framework to detect forest fire. Proposed framework includes wireless sensor network architecture, a sensor deployment scheme, clustering algorithms, and communication protocols. Important design goals of proposed framework are energy efficiency, early detection and accurate localization, forecast capability, and adapting to harsh environments. Evaluation and validation of the proposed framework are done by implemented simulator considering energy consumption of sensor nodes. According to the results of the study, deterministic deployment of sensor nodes extends lifetime of the sensor network compared to random deployment and it is suggested that clustered hierarchy should be used since it has benefits in terms of data aggregation, energy efficiency, sensor coordination, and management capabilities.

In the study of Bhosle et al.[97], the authors propose a WSNs based disaster management framework. The case study is forest fire detection. Study reviews available networking standards such as IEEE 802.15.4, ZigBee, 6LoWPAN, and Wibree for forest fire detection case. Proposed framework consists of BS (base station) which is responsible for communication with backbone network through gateway, sink node where measured data are collected, and wireless sensor nodes which are used to monitor environment. Sink nodes are capable of making decisions according to predefined threshold and send warning or alert signals to the end user via BS. No evaluation is conducted to test the proposed framework in this study.

In a similar study [98], WSNs based forest fire detection architecture is proposed. LTE-M (Long Term Evolution-Machines) based architecture includes LTE-M modules which are mounted on the belt of forest animals. In addition, there are some stationary sensor nodes on the trees which are used to measure temperature, humidity, light and CO(carbon monoxide). LTE-M module collects data from stationary modules while animals moving in the forest using ZigBee. After collecting data, it sends data to the cloud database. However, proposed system is not evaluated.

Pico et al. [99] propose hierarchical WSNs based forest fire early detection system. System has two types of nodes which are central nodes and sensor nodes. Sensor nodes are responsible for monitoring environment by measuring temperature and relative humidity levels. Central sensor, on the other hand, are responsible for collecting data from sensor nodes and delivering data to control center. Proposed framework is validated by simulation and real test bed results.

General WSNs framework for forest fire detection and surveillance is proposed by Xu et al. in [100]. Study considers sensor deployment aspects, clustered hierarchy, and energy aware intra and inter cluster protocols. Most important metrics in the study are energy consumption and fire detection delay of the system. NS-2 simulation is used to evaluate system performance. For sensor deployment, authors propose a new metric which represents the probability of fire at particular place, Fire Occur Degree(FOD). In addition to FOD, there are other parameters such as initial energy, network lifetime, and time required to detect fire. According to parameters, distance between sensor nodes are calculated and sensor deployment is performed accordingly. Clustered hierarchy is used as a network topology. In addition, E-LEACH which is the enhanced version of LEACH is proposed where each node maintains its own feature table. According to simulation results, proposed framework can be used for effective and energy efficient forest fire detection. In addition, they conclude that environmental and seasonal factors are very important for WSNs based forest fire monitoring systems.

Arduino MCU based forest fire detection is proposed by Basu et al. in [101]. In their study, authors propose a system architecture which includes arduino board, temperature sensor, gas sensor, ethernet module, buzzer, and LCD. The main objective of the proposed system is measuring temperature and gas levels of the environment and create fire alert if temperature and gas levels are higher than the threshold value. Proposed architecture is discussed by using system block diagram and operational flow chart. However, no evaluation is conducted either networking aspects or energy efficiency.

Another Arduino based application for forest fire detection is proposed by Singh[102]. Humidity, gas, and smoke sensors are used to monitor forest. Additionally, Bluetooth and GSM modules are used to achieve real time communication. Two different scenarios are proposed in the study which are during normal conditions and during fire conditions. In the first scenario, proposed system is used as monitoring system which measures humidity, gas and smoke levels of the environment and sends the information to BS using Bluetooth. In case of fire, where sensor readings are higher than threshold, GSM module is used to alert end-user via SMS and call. The system is implemented by programming the Arduino board. System design and necessary discussions are given, yet evaluation of the system is not available.

Another forest fire detection study in [103] proposes a prototype of automatic forest fire detection system using Raspberry Pi board and CM5000 sensor motes. Raspberry Pi board is used as sink node and CM5000 sensor motes are used to measure ambient temperature and carbon dioxide. Hierarchical approach is used to deploy sensor motes. Hierarchy is achieved by creating clusters of CM5000 motes and cluster heads are responsible for transmitting measurements to the sink node which is Raspberry Pi. According to measurements, Raspberry Pi is capable of collecting alerts and sends them to the base station. A real test bed is deployed, yet performance or energy efficiency evaluation of the system is not conducted in the study.

In study [104], another forest fire detection and verification system is proposed. Linksys WRT54GL router (Cisco Systems) which is an embedded system with IEEE 802.11 b/g interface, a FastEthernet interface, General Purpose Input/Output (GPIO), UART (JP2), and ETAJ (JP1) ports. This board is used to monitor environment using connected infrared radiation and smoke sensors. In addition to the board, IP cameras are used for live monitoring. According to sensor readings, IP cameras change directions and monitor the area where they receive forest fire alarm. The proposed system is tested in terms of bandwidth and power consumption using real test bed.

UAV based forest fire detection system using IR(infrared) images is proposed in [105]. Proposed system uses image processing techniques for UAV applications to detect forest fires. New image processing algorithm is presented to process IR images and extract the information whether there is a fire or not. Experiments are conducted using IR fire video sequences and system is validated in terms of accuracy.

When the existing work on forest fire detection is considered, we can see that some studies such as [97], [98], [101], [102], [103] fail to address the energy efficiency since they are not offering any evaluation method. On the other hand, while studies such as [96], [100], [99] use simulation tools to show the energy efficiency of their proposed architecture, they fail to consider the accuracy of detection. Nevertheless, the existing forest fire detection systems are heavily dependent on either scalar information processing or multimedia information processing. While scalar sensor based ones are more energy efficient, there are reported problematic conditions in terms of accuracy. Multimedia information based decisions on the other hand are reported to be more accurate, however, the main problem of these frameworks are the high energy requirement of the multimedia sensors as well as the transmission of multimedia information. In this work, the two approaches are combined and a heterogeneous architecture is proposed and evaluated for accuracy, efficacy as well as QoS. The contributions of the heterogeneous approach are emphasized and explained in detail in the following chapters of this thesis.

3.2 Energy Efficiency Perspective

Studies in the literature on environmental monitoring applications including forest fire detection can be categorized as shown in Table 3.1. The system design of environmental monitoring applications using Raspberry Pi and Arduino is proposed in the study by Ferdoush, Sheikh and Xinrong Li [106]. Scalar sensor readings such as temperature and humidity are used to monitor the environment. A systematic review is presented for wireless sensor network applications for monitoring coal mines in [107]. WSN based Forest Fire Monitoring is used to test the anomaly detection algorithm in [108]. The proposed method estimates the maximum number of malicious actions tolerated by the application. Sensor deployment design is automated using the proposed method. A similar study is presented in [109], where body temperature, respiration rate, heart rate and body movements are monitored using Raspberry Pi boards. A detailed system design is proposed, but no analysis in terms of accuracy or energy efficiency is performed.

In [110], a video sensor platform is described and high-quality video transmission over 802.11 networks is discussed. The proposed system is evaluated in terms of energy consumption. It is shown that the video transmission is performed with power requirement of approximately 5 Watts. A traffic monitoring system is proposed in [41]. Raspberry Pi board and HD camera are used for object detection and the system is analyzed in terms of energy consumption. A smart surveillance system using Raspberry Pi, USB camera and PIR sensor is proposed in [111] as well. PIR sensors are mainly employed to trigger the Raspberry Pi. In the study of Kulkarni et al.[112], SensEye system is proposed. In their proposed system, the efficiency of multi-tier heterogeneous networks is discussed in terms of energy and accuracy. The authors used multimedia sensors in tiers. SensEye project is discussed in terms of possible challenges of multi-tier networks in [113] as well. Multiple sensing modalities, protocols for multi-tier interaction and resource management, design trade-offs in multiple tier networks, and programming abstractions are discussed.

In [114], an object classification approach using multimedia sensors is proposed. The classification approach is in turn analyzed in terms of accuracy and energy efficiency. Study investigates possible research directions to utilize WSN techniques and applications for efficient monitoring. Another study related with security proposes techniques for Sybil attack detection for a forest wildfire monitoring application [115]. The proposed approach using two-tier detection system which has high-energy nodes at the lower tier and normal sensing nodes in upper tier. Sybil attack is detected using residual energy of high-energy nodes. If two or more incoming control packets have same residual energy, it is a sign of a Sybil attack. Multimedia sensors are used to classify moving objects in [116] as well. Proposed framework makes use of accuracy of multimedia sensors in monitoring applications. Experimental setup is implemented and related tests are conducted to show accuracy and energy efficiency of the framework. Multimedia and scalar sensors are used together in [117] for detection and classification of objects using visual and auditory data fusion. A practical implementation of the proposed system is presented and performance and energy consumption results are discussed.

Table 3.1: Literature categorization

Study	Sensors Used	Evaluation
[110],[41],[111]	Multimedia	Energy
[105]	Multimedia	Accuracy
[112],[113],[114],[116]	Multimedia	Energy & Accuracy
[106],[109],[97],[98],[101],[102],[103]	Scalar	Only System Design
[118],[96],[100],[99]	Scalar	Energy
[107]	Scalar	Review
[108], [115]	Scalar	Security
[117],[104]	Multimedia & Scalar	Energy & Accuracy

The literature presents numerous examples of surveillance and/or monitoring applications for scalar and multimedia sensors. Some of them focus on energy efficiency and use scalar sensors. On the other hand, some other studies focus on the accuracy of monitoring information and use multimedia sensors for more accuracy. In addition, the studies differ in terms of analysis of the proposed approaches.

Combining the energy efficiency of scalar sensors and accuracy of multimedia sensors has been considered in studies such as [104], [117] which are relatively quite recent. However their main domain is the surveillance applications rather than focusing on forest fire applications. To the best of our knowledge this work is the first one to attempt combining the two approaches and evaluate in terms of efficacy and accuracy using simulation as well as real test bed deployments.

3.3 Machine Learning Applications for (Forest) Fire Detection

Accuracy of forest fire detection systems should be as important as the energy efficiency. Considering wireless sensor networks, most of the applications use scalar sensor and scalar data to detect forest fire. Scalar sensors are perfectly suitable for such applications especially in terms of energy efficiency. Since the system is energy aware, maintenance cost of the system (e.g. battery replacement) can be decreased. On the other hand, accuracy of detection in disaster and surveillance applications is important as well. In this sense, scalar sensors become insufficient in terms of accuracy. In order to introduce a higher degree of accuracy reliability, it is possible to use more detailed data such as images, videos, and audios. Multimedia sensors are capable of gathering such data, yet the main constraint is energy consumption as discussed in previous chapters. It is possible to make use of edge computing in order to mitigate the energy related issues of WMSNs. Edge allows the computation of data to be performed on the edge devices instead of sending every information to the BS, which in turn leads to less data transmission. By decreasing the amount of transmission, it is possible to extend lifetime of sensor nodes since the most energy consuming activity is data transmission. Literature for forest fire detection generally includes studies of image processing, video processing, and rarely machine learning techniques for WSNs applications. Machine learning, especially deep learning neural networks are rarely applied to WSNs applications since they need considerably high computation power which is not possible with limited resources of IoT hardware. For example, the study of Vipin et al. [119] proposes image processing based forest fire detection algorithm. Authors detect forest fire from images using rule based color model and pixel classification. According to results of the study, they achieved the accuracy of 99% whereas 14% is the false alarm rate. Another image processing based detection system is proposed in [120]. Candidate smoke regions are extracted from the smoke motion using video frames. Proposed model is evaluated using experiments and rate of area change is used as a feature to distinguish smoke and non-smoke regions. In [121], early detection system for forest fire based on video processing is proposed. Flame and smoke detection is used to validate that there is fire. Image sequences are gathered from video and, flame and smoke pixels are extracted. According to processing results, system has the ability to raise alarm if there is fire. Another image processing based forest fire detection system is discussed in [122]. UAV images are used to detect forest fire. Two step detection is applied. In the first step, color based detection is used to detect flame and smoke pixels. As a second step or validation step, two-dimensional discrete wavelet transform is applied to distinguish flame and smoke areas from others. Unmanned substation environment fire detection study is proposed based on image processing in [123]. Fire area identification and flame extraction from images is applied to detect fire. Forest fire detection framework based on image processing is studied in [124] as well. Scale-Invariant Feature Transform (SIFT) is used for feature selection, then SVM and KNN classifiers are applied for classification of fire and non-fire images. Surit et al. [125] propose video based digital image processing for forest fire smoke detection. Framework consists of four stages which are area of change detection, segment the area of change, calculate static and dynamic characteristics, and check whether the changed object is smoke or not. Similarly in [126], authors use color based image processing to detect forest fire. Flame detection is used to identify if there is a fire or not.

In [127] fire image recognition is presented to detect fire. Image processing based approach uses infrared based images to detect flames from images. The double band method is used where distance is not a constraint compared to single band method. In [128], Premal and Vinsley propose forest fire detection framework based on image processing. Rule based color model is applied because of reduced amounts of complexity. Created model is capable of separating flame pixels and high temperature fire centre pixels. They achieved 99.4% detection accuracy and 12% false alarm rate. Zhang et al.[129] propose ANN (Artificial Neural Networks) based forest fire detection system. Video based images are used to detect forest fire. According to results, 98.94% accuracy is achieved. Combination of CNN(Convolution Neural Networks) and RNN(Recurrent Neural Networks) are used to create deep learning model for detecting fire from video sequences [130]. In average 92% accuracy achieved in the study. CNN based early fire detection method is proposed in [131]. Study makes use of processing capability of CCTV cameras which are already used in surveillance applications. Proposed method is applicable both indoor and outdoor environment. However, the model size is heavy (238 MB) to be used in WSNs applications. Another CNN based wildfire detection system is proposed by Lee et al.[132]. UAV images are used to train the network. Proposed system achieved the best accuracy with 99% with GoogLeNet.

Table 3.2: Forest fire detection with machine learning

Study	**Method**	**Study**	**Method**
[119]	Image Processing	[132]	CNN
[120]	Image Processing	[127]	Image Processing
[121]	Image Processing	[131]	CNN
[122]	Image Processing	[128]	Image/Video Processing
[123]	Image/Video Processing	[129]	Image Processing + ANN
[124]	Image Processing	[105]	ANN
[125]	Image Processing	[130]	CNN + RNN

Considering WMSNs applications of forest fire detection in the literature which make use of image detection, majority of the studies solely rely on image and video processing to detect forest fires. Only in few of the studies such as [129], [105], [130], use machine learning models including ANN, SVM, and RNN. The proposed machine learning models are not appropriate to be used in IoT edge devices since they come with relatively high requirements in terms of memory space and require high computing power. In this study, a lightweight CNN machine learning model is proposed to enable edge computing and to increase the accuracy as well as the efficacy of the system. Proposed model is trained using a newly created forest fire data set which is the first one created specifically for forest fires. The new data set is in turn used to train the lightweight CNN model.

3.4 Statistical Analysis of Sensor Lifetime

There are several tools and techniques to analyze the lifetime and reliability of sensor nodes. Most of the studies in the literature use simulation or analytical modelling for this purpose [133], [134], [135], [136], [137]. However, using statistical analysis to analyze the lifetime of sensors using real data is rare. There are some studies which discuss energy efficiency and the factors that affect the efficiency of sensor networks [138], [136]. However, most of them use comparison techniques or comparative analysis to show how efficient their system or approach. Rather than comparative or descriptive analysis of the system, applying more valuable statistics such as time series analysis, and survival analysis to such systems can draw more valuable conclusions in terms of lifetime analysis of low-power wireless sensors. In the literature, analysis are performed using several methods. It can be categorized as analytical modelling, experiment, and simulation as summarized in Table 3.3.

In the study of Chen et al.[133], a general formula for lifetime analysis is derived using analytical modelling. The proposed formula identifies two key variables which affect the lifetime of the network. These variables are channel state and residual energy of the sensor. Using similar approach, Duarte-Melo et al.[134] proposed a mathematical formulation to estimate energy consumption and lifetime of a sensor node based on a clustering mechanisms with parameters related to sensing field like

Table 3.3: Evaluation methods of sensor lifetime

Study	Year	Method Used
Chen, Yunxia, and Qing Zhao, "On the lifetime of wireless sensor networks."	2005	Analytical Modelling
Duarte-Melo, Enrique J., and Mingyan Liu, "Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks."	2002	Analytical Modelling
Shah, Rahul C., Sumit Roy, et al., "Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks."	2003	Analytical Modelling
Kumar, Santosh, Anish Arora, and Ten-Hwang Lai, "On the lifetime analysis of always-on wireless sensor network applications."	2005	Experiment
da Cunha, Adriano B., and Diógenes C. da Silva. "An approach for the reduction of power consumption in sensor nodes of wireless sensor networks: Case analysis of mica2."	2006	Experiment
Polastre, Joseph, Robert Szewczyk, Alan Mainwaring, David Culler, and John Anderson. "Analysis of wireless sensor networks for habitat monitoring."	2004	Experiment
Nguyen, Hoang Anh, Anna Förster, Daniele Puccinelli, and Silvia Giordano. "Sensor node lifetime: An experimental study."	2011	Experiment
Jung, Deokwoo, Thiago Teixeira, and Andreas Savvides. "Sensor node lifetime analysis: Models and tools."	2009	Simulation
Di Pietro, Roberto, Luigi V. Mancini, Claudio Soriente, Angelo Spognardi, and Gene Tsudik. "Catch me (if you can): Data survival in unattended sensor networks."	2008	Simulation
Dron, Wilfried, Simon Duquennoy, Thiemo Voigt, Khalil Hachicha, and Patrick Garda. "An emulation-based method for lifetime estimation of wireless sensor networks."	2014	Simulation
Ma, Zhanshan, and Axel W. Krings. "Insect population inspired wireless sensor networks: A unified architecture with survival analysis, evolutionary game theory, and hybrid fault models."	2008	Game Theory & Survival Analysis

size, distance, etc. Given formulation helps to quantify the optimal number of clusters and shows how to allocate energy between different layers.

In the study of Shah et al.[135], simple analytical model is used to analyze the performance of the system. Proposed approach investigates the benefits of three-tier architecture for collecting sensor data. According to given results in the study, three-tier architecture approach can lead to substantial power savings at the sensors. Another study which analyzes network lifetime by experiment is the study of Kumar et al.[138]. The proposed approach is proved by deploying ExScal (a large-scale WSN for intrusion detection) to identify major components in the network lifetime analysis. Results of experiments show how to analyze the effects of using various non-sleep-wakeup power management schemes such as hierarchical sensing, low-power listening, and in network data aggregation on the network lifetime. The case study of Cunha et al.[139], analyzes wireless sensor node of Mica2 and proposes an approach to reduce power consumption which in turn increases lifetime. Proposed approach is verified with experiment. Another experimental study by Polastre et al.[140] analyzes system performance using environmental and node health data from experiment. The study of Nguyen et al. [141] also experimentally analyzes lifetime of TelosB sensors using different commercial batteries. Simulation is another method to analyze wireless sensor networks. Jung et al.[136] analyze two modes of operation of sensor nodes using models and study presents a MATLAB Wireless Sensor Node Platform Lifetime Prediction and Simulation Package (MATSNL). Dron et al. [142], use Contiki Cooja simulator to analyze and model complex battery characteristics and node lifetimes in WSNs. In the study of Zhanshan et al. [143], authors envision a WSN as an entity analogous to a biological population with individual nodes mapping to individual organisms and the network architecture mapping to the biological population. The interactions between individual WSN sensors, are captured with evolutionary game theory models. On the node level, survival analysis is introduced to model lifetime, reliability and survival probability of WSN nodes.

As already discussed, most of the WSNs studies in the literature evaluate proposed systems using analytical models, simulations, and experiments. These methods are widely used and well researched. Simulation and experiment methods are used by this study as well to evaluate the proposed framework. However, statistical analysis is also conducted as a first step to investigate the factors which have the potential to affect the lifetime of sensor nodes since one of the main objectives of this study is energy efficiency. Employed regression models and discussions for the results of the analysis are presented and explained in detail in the following chapters of this thesis.

CHAPTER 4

STATISTICAL ANALYSIS FOR LOW-POWER SENSOR MOTES

The main objective of this chapter, is to understand the effects of external factors such as humidity, light, and temperature on the discharge of battery of sensor nodes. For this purpose, statistical analysis is employed as the first step prior to our simulation studies and test bed experiments. Linear regression and ordered logit regression models are employed and results of the analysis are discussed and explained in detail in the following sections.

4.1 Dataset

The data set which is used in this chapter is Mica2Dot sensor data from Intel Berkeley Research Lab[144]. The raw data include 2.3 million records of Mica2Dot sensor. Variable names are date, time, epoch(sequence number), mote id, temperature, humidity, light and voltage values. The deployment of sensor nodes is given in Figure 4.1.

To understand and analyze the data, some models are used and results are discussed in detail. As a first model, linear regression is used without relying any difference between censored and uncensored spell lengths, i.e., treating all durations as uncensored ones. Dependent variable is "dur" which is duration. Independent variables are "temp", "humidity", "light", and "voltage". In total, we have 58 different sensors, yet for regression, we use "mote1". After giving linear regression results, ordered logit regression model is employed. Results are also discussed with relevant tables and figures.

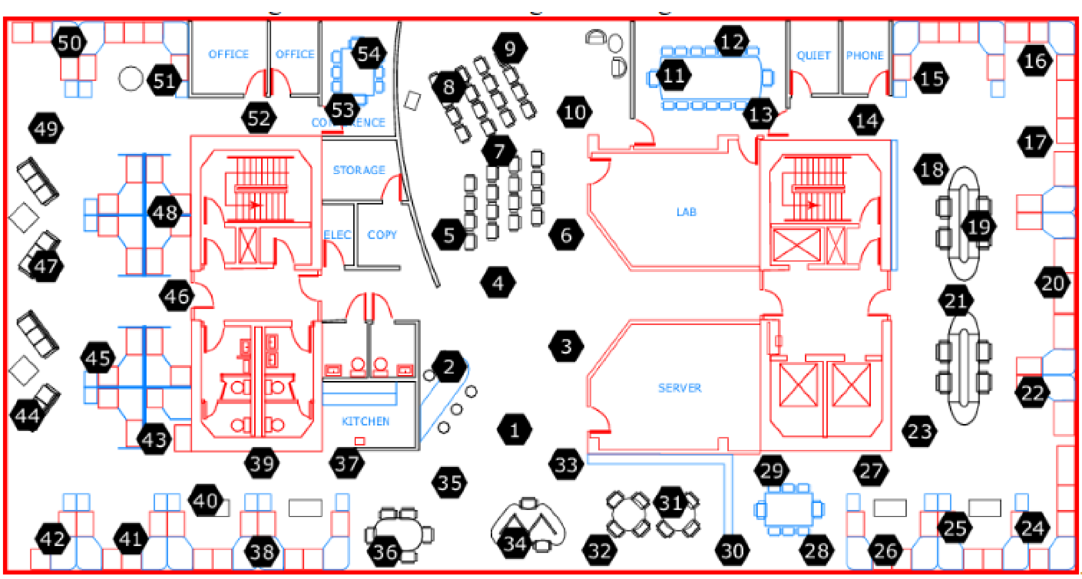


Figure 4.1: Deployment of sensors

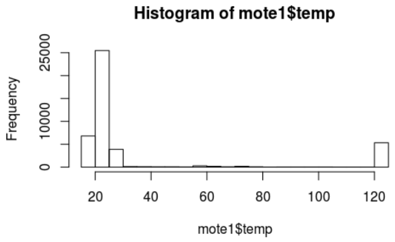
4.1.1 Descriptive Analysis

The data set is a wireless sensor data set with dimension of 2313682 observations and 8 variables. Summary of data is given in Table 4.1. It gives summary of numerical data in data set. When we analyze the summary table, there are extreme values for each variable in the data set which are the sign of malfunctioning of the sensor since Mica2Dot sensors work between 2.7 V - 3.3 V [145] and we observe that unexpected measurements are because of low voltage supply(i.e. less than 2.7 V). To specify the spell ends, the time that sensor node starts to malfunction can be accepted as failure event. In addition, there are missing (NA) values exist. NA value for this data set shows that sensor node stops sending data to the sink. Since NA value means the sensor node does not send measurements to the sink node, it can be assumed that the first NA value we have is the instance that the sensor node failed or failure event of sensor node. Since the data set is very large with 2313682 observations, it can be a

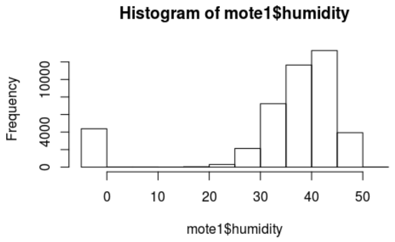
Table 4.1: Summary table

Variable	Min	1st Qu.	Median	Mean	3rd Qu.	Max	NA's
temp	-38.40	20.41	22.44	39.20	27.02	385.57	526
humidity	-8983.13	31.88	39.28	33.91	43.59	137.51	899
light	0.0	18.4	143.5	390.9	507.8	1847.4	903
voltage	0.01	2.39	2.53	2.49	2.63	3.16	93879

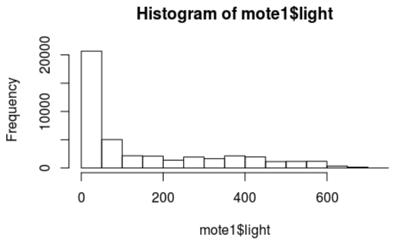
good approach to create subsets of it with respect to *"moteid"*. There are 58 unique motes in the original data set which means that we have 58 different data sets with same variables from original data set.



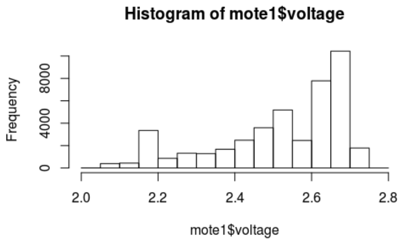
(a) Temperature



(b) Humidity



(c) Light



(d) Voltage

Figure 4.2: Histograms for mote 1

In the Figure 4.2, histograms for mote 1 are given for different variables. Histograms obviously show that except voltage and light variables, there are some extreme values for other variables. For temperature variable in Figure 4.2(a), approximate range for the expected values are between 20 and 45 centigrade degrees. The values out of this range can be expressed as extreme values since the measurements are from lab environment. Humidity variable again has some extreme measurement values according to Figure 4.2(b). Humidity should range between 0 and 100 as percentages. The negative values seen in histogram of humidity are the extreme values or wrong measurements which are the sign of sensor malfunctioning. Light variable in Figure 4.2(c) is measured in unit of lux. According to data set explanations from [144], 1 lux corresponds to moon light, 400 lux to bright office and 100,000 lux to full sunlight. Except from NA values there is no extreme values. As a last variable to analyze, the voltage variable in Figure 4.2(d) shows the voltage readings of sensor node. Voltage variable depicts supply voltage value in unit of volts.

Table 4.2: Correlation matrix

Correlation	temp	humidity	light	voltage
temp	1.0000000	-0.8019789	0.01672185	-0.8010284
humidity	-0.80197886	1.0000000	-0.10915454	0.5370372
light	0.01672185	-0.1091545	1.00000000	0.0817813
voltage	-0.80102843	0.5370372	0.08178130	1.0000000

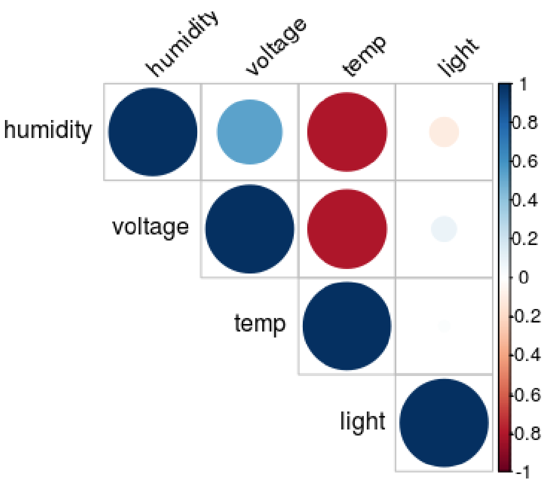


Figure 4.3: Correlation matrix representation

In Table 4.2 and Figure 4.3 correlations between variables are given. According to given statistics, humidity and temperature have strong negative correlation. In addition, humidity variable have weak positive correlation with voltage variable. Temperature variable has strong negative correlation with voltage variable. From given representations of correlations, it can be said that temperature and humidity variables can have effects on sensors' lifetime since they affect the voltage variable in negative or positive way.

4.2 Methodology and Results

4.2.1 Linear Regression Model

First of all, linear regression is employed for each independent variable temperature, humidity, light, and decreasing voltage one by one. Some abbreviations are used in tables below. "dur" is the dependent variable in the regression model which shows the lifetime (i.e. duration that sensor is alive) of sensor node. “dvoltage” is the decreasing voltage, and “temp” is temperature which are used as independent variables. Additionally, “R^2” is the statistical measure which shows how close the data are to the regression model. “F Statistic” is another statistical measure which tests the overall significance of the model. “Constant” value is the expected mean value of dependent variable when all independent variables are zero. The numbers next to the variables (which are not in parentheses) are regression coefficients. They show expected change in dependent variable for one unit change in independent variable. The numbers next to the variable (which are into the parentheses) are standard errors. “df” notation means the degrees of freedom. Total degrees of freedom is n-1 which means one less than the number of observations, and degrees of freedom of regression is number of dependent variables which is 1 in our case. “p” values, on the other hand, show the confidence intervals and each confidence interval is showed using stars(*). One star(*) shows confidence interval of 90%, two stars(**) 95%, three stars(***) 99%, and no stars means that the independent variable is not significant. According to the results, temperature variable does not have significant effect on lifetime. Similarly, humidity does not show any significant effect. On the other hand light has very significant positive effect on lifetime of a sensor. Similarly voltage has very significant effect on lifetime as shown in Table 4.3, Table 4.4, Table 4.5, and Table 4.6.

Table 4.3: Temperature

	Dependent variable:
	dur
temp	−0.272
	(1.339)
Constant	63.305**
	(29.598)
Observations	36,223
R^2	0.00000
Adjusted R^2	−0.00003
Residual Std. Error	590.431 (df = 36221)
F Statistic	0.041 (df = 1; 36221)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 4.4: Humidity

	Dependent variable:
	dur
humidity	0.610
	(0.624)
Constant	33.913
	(24.135)
Observations	36,223
R^2	0.00003
Adjusted R^2	−0.00000
Residual Std. Error	590.424 (df = 36221)
F Statistic	0.957 (df = 1; 36221)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 4.5: Light

	Dependent variable:
	dur
light	0.048***
	(0.017)
Constant	49.657***
	(4.153)
Observations	36,223
R^2	0.0002
Adjusted R^2	0.0002
Residual Std. Error	590.369 (df = 36221)
F Statistic	7.712*** (df = 1; 36221)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 4.6: Decreasing voltage

	Dependent variable:
	dur
dvoltage	127.716***
	(29.911)
Constant	386.750***
	(77.214)
Observations	36,223
R^2	0.001
Adjusted R^2	0.0005
Residual Std. Error	590.283 (df = 36221)
F Statistic	18.231*** (df = 1; 36221)
Note:	*p<0.1; **p<0.05; ***p<0.01

After regression results of each independent variable temperature, humidity, and light, another regression model is created. Only the external factors are used in this model. Our dependent variable is duration again. Independent variables are temperature, humidity and light. The results are depicted as shown in Table 4.7. According to results, only light shows significant effect in positive way for lifetime.

Table 4.7: Temperature + Humidity + Light

	Dependent variable:
	dur
temp	-1.910
	(1.730)
humidity	0.931
	(0.734)
light	0.069^{***}
	(0.020)
Constant	52.622
	(55.847)
Observations	36,223
R^2	0.0004
Adjusted R^2	0.0003
Residual Std. Error	590.344 (df = 36219)
F Statistic	4.259^{***} (df = 3; 36219)
Note:	*p$<$0.1; **p$<$0.05; ***p$<$0.01

Table 4.8 shows the effects of all combined independent variables. According to linear regression results, light and voltage show significant effect. They affect the lifetime in positive way. On the other hand, temperature has significant effect which is negatively related with lifetime.

Table 4.8: Temperature + Humidity + Light + Decreasing voltage

	Dependent variable:
	dur
temp	-3.476^{**}
	(1.764)
humidity	-1.091
	(0.860)
light	0.072^{***}
	(0.020)
dvoltage	159.286^{***}
	(35.374)
Constant	575.055^{***}
	(128.755)
Observations	36,223
R^2	0.001
Adjusted R^2	0.001
Residual Std. Error	590.187 (df = 36218)
F Statistic	8.265^{***} (df = 4; 36218)
Note:	*p$<$0.1; **p$<$0.05; ***p$<$0.01

In Table 4.9, summary of six regression models are showed. “+” (positive) sign shows that coefficient of variable is significant and affects the dependent variable in positive way. “-” (negative) sign means that coefficient of variable is significant and affects the dependent variable in negative way. “0” (zero) means that coefficient of the variable is not significant. If the cell is empty, it means that independent variable is not used in the model. Considering results of the regression models, temperature variable is not significant when it is the only independent variable in the model. However, as shown in model five and model six, temperature variable is negatively significant. Since the sign of the temperature changes from model to model, it is showed that linear regression models do not explain wireless sensor data.

Table 4.9: Linear regression results summary

	M1	**M2**	**M3**	**M4**	**M5**	**M6**
temp	0				-	-
humidity		0			0	0
light			+		+	+
dvoltage				+		+

4.2.2 Ordered Logit Regression

After linear regression model, ordered logit regression model is employed. In ordered logit regression model, the data are grouped by their duration values. There are 8 groups which are 0-24, 25-49, 50-74, 75-100, 100-124, 125-149, 150-174, 175-inf. The summary of data is given in Table 4.10 with the values of number of observations (N), mean, standard deviation (St.Dev.), minimum (min), 25 percentile (Pctl(25)), 75 percentile (Pctl(75)), and maximum (max).

Table 4.10: Summary of ordered logit data

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
moteid	36,223	1.000	0.000	1	1	1	1
spell	36,223	18,112.000	10,456.820	1	9,056.5	27,167.5	36,223
temp	36,223	21.987	2.317	17.195	20.459	23.066	30.935
humidity	36,223	38.382	4.975	22.465	34.883	42.317	50.739
light	36,223	158.197	177.702	0.020	41.400	279.680	713.920
voltage	36,223	2.579	0.104	2.293	2.506	2.663	2.762
dur	36,223	57.326	590.423	0	29	61	88,084
status	36,223	0.000	0.000	0	0	0	0
dvoltage	36,223	−2.579	0.104	−2.762	−2.663	−2.506	−2.293

In Table 4.11, ordered logit regression results are presented. Temperature, humidity, light, and decreasing voltage variables are used as independent variables in the model and dependent variable is duration, yet grouped duration (durG) are used this time. According to results of regression, humidity and voltage have significant effect on lifetime, yet temperature and light lose their significance.

Table 4.11: Temperature + Humidity + Light + Decreasing Voltage

	Dependent variable:
	durG
temp	-0.003
	(0.005)
humidity	0.009^{***}
	(0.002)
light	0.0001
	(0.0001)
dvoltage	2.251^{***}
	(0.047)
Observations	36,223
Note:	*p<0.1; **p<0.05; ***p<0.01

Since the main aim of the study is explaining variables which have effects on lifetime of low-power sensors, descriptive analysis of raw data is done to understand data well. After descriptive statistics, data are analyzed using linear regression model. According to results of linear regression model, it is shown that linear regression does not explain low-power sensor data well. After linear regression model, ordered logit regression is employed and results are discussed. In addition, probabilities of being in each group also shown. According to the results of ordered logit, humidity and voltage have effect on lifetime of sensors.

The concept of IoT and smart environment applications is becoming more and more popular. With smartness of environment and wide use of electronic devices and sensors, lifetime and energy consumption analysis of wireless sensor networks became a must. Wide application area of wireless sensor networks such as disaster surveillance, healthcare, etc. raises the importance of lifetime analysis. In this chapter, linear regression and ordered logit regression models are employed for lifetime of wireless sensors using the data from *“Intel Berkeley Research Lab"*. As a first step, literature review is conducted to show which methods are used to analyze sensor lifetime. The corresponding categorization is done and presented in Chapter 3. Generally, there are three main methods in the literature which are analytical modelling, experiment and simulation. Statistical analysis is rare in the area of wireless sensor networks in terms of lifetime analysis. After literature review, data set is described and explained in details by descriptive analysis. Results of descriptive analysis are proposed and discussed. Moreover, linear regression and ordered logit regression analysis are conducted. The results of analysis are also discussed. In conclusion, the importance of statistical analysis to understand variables that affect lifetime is presented. However, considering results of the analysis conducted, it is shown that linear regression and ordered logit regression models may not be able to explain the behavior observed through the data obtained for wireless sensor networks, and they are not sufficient by themselves to draw valuable conclusions. With this conclusion, it became inevitable to conduct more advanced statistical methods to explain effects of external variables on sensor lifetime. More advanced methods such as survival analysis and time series analysis will be considered as a future work.

CHAPTER 5

FOREST FIRE DETECTION AND EARLY DIAGNOSIS FRAMEWORK

In this chapter, the proposed early forest fire detection framework which makes use of scalar and multimedia sensors hierarchically is explained in detail. The main objective of the framework is to create an early forest fire detection system by considering energy efficiency as well as accuracy of the system. Energy efficient system is achieved by using scalar sensors in the first level of hierarchy. Scalar sensors are capable of sensing scalar data such as temperature, humidity, and light. In our approach, scalar sensors are responsible for monitoring forest environment using scalar measures. According to scalar measures, first phase detection is completed. In the second level of the hierarchy, multimedia sensors are used. Multimedia sensors are capable of sensing multimedia data such as audio, video, and instant images. Since multimedia data are more informative compared to scalar data, accuracy of the system is achieved by making use of multimedia sensors in the second level of hierarchy.

5.1 Scalar Sensors

XM1000 sensor motes are used in this study as scalar sensor motes. They consist of three sensors, communication module, memory, and micro-controller. The processor model of XM1000 is TI MSP430F2618 (Texas Instruments MSP430 family 16-Bit RISC Architecture). It has 116 KB of flash memory, 8 KB of data RAM, and 1 MB of external flash. In addition, it has UART, SPI, and I2C as serial interfaces and USB interface. Temperature, humidity, and light sensors are integrated on the XM1000 mote. It has two different light sensors. One of them is Hamamatsu S1087, which is visible range light sensor. It has 560 nm peak sensitivity wavelength. The other is

Hamamatsu s1087-01, which is visible and infrared range light sensor. It has 960 nm peak sensitivity wavelength. Temperature and humidity sensor is Sensirion SHT11. It has temperature range from -40 to 123.8°C with ±0.4°C accuracy. In addition, it has humidity range between 0 and 100% with ± 3% accuracy. As a communication unit, XM1000 has TI CC2420 RF chip. The frequency band is 2.4 GHz. The RF power is software configurable with range between -25 dBm and 0 dBm. It has range of 120 meters outdoor and 20-30 meters indoor. XM1000 uses 2 AA batteries as a power supply [64], [66].

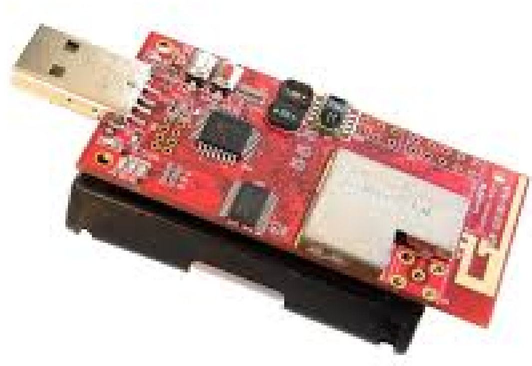


Figure 5.1: XM1000 sensor mote

5.2 Multimedia Sensors

Raspberry Pi 3s are employed as multimedia motes. These boards have Broadcom BCM2837 system-on-chip (SoC) which includes four high-performance ARM Cortex-A53 processing cores running at 1.2GHz with 32kB Level 1 and 512kB Level 2 cache memory, and a VideoCore IV graphics processor. The board is also linked to a 1GB LPDDR2 memory module and has Broadcom BCM43438 wireless radio communication chip. It also provides 2.4GHz 802.11n wireless LAN, Bluetooth Low Energy, and Bluetooth 4.1 Classic radio support. In addition, it has 40-pin general-purpose input-output (GPIO) header, HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface(CSI), and Display Serial Interface (DSI) ports. It uses 5.1 V 2.5 A power supply units [146].

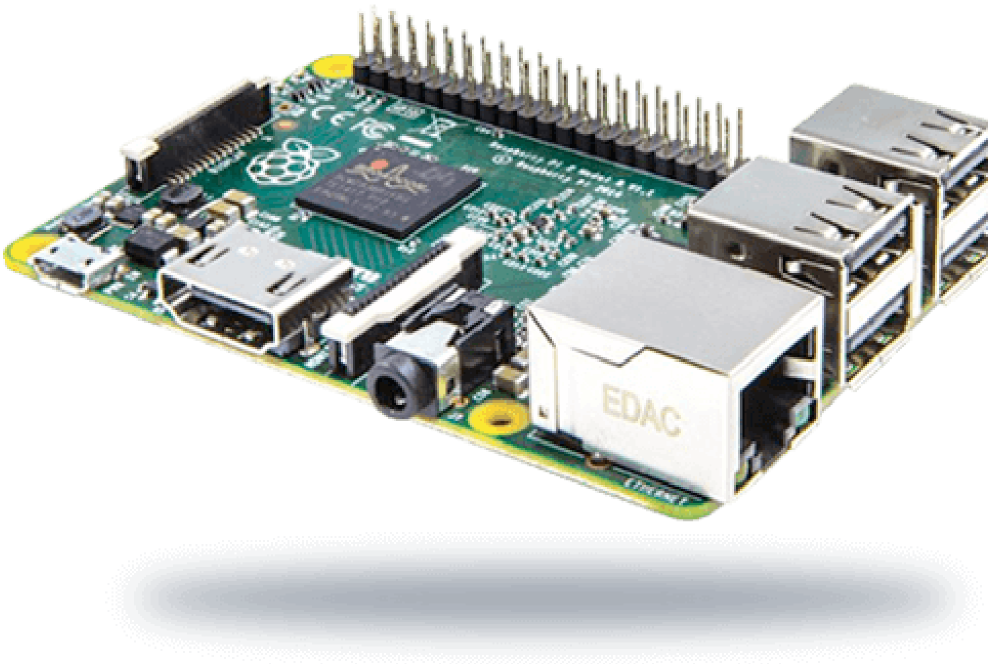


Figure 5.2: Raspberry Pi

5.3 System Design

The system design of the proposed framework is presented in this section and depicted in Figure 5.3. As shown in the figure, XM1000 sensor motes are used to monitor forest area by gathering scalar data. Gathered data are processed by each individual node. Processing data on the edge prevents unnecessary data transmission between nodes. In the first phase of the detection, scalar sensors are used to monitor forest environment by sensing temperature. Measured values are compared with threshold temperature. If there is any extreme temperature reading, all measured values are transmitted to the sink node via intermediate nodes. Transmitted packet includes date and time, id of sensor node, number of hops until reaching sink node, sequence number(i.e. the sequence number of packet send by same sensor node), size of packet, temperature, humidity, light, and the voltage level as shown in Figure 5.4.

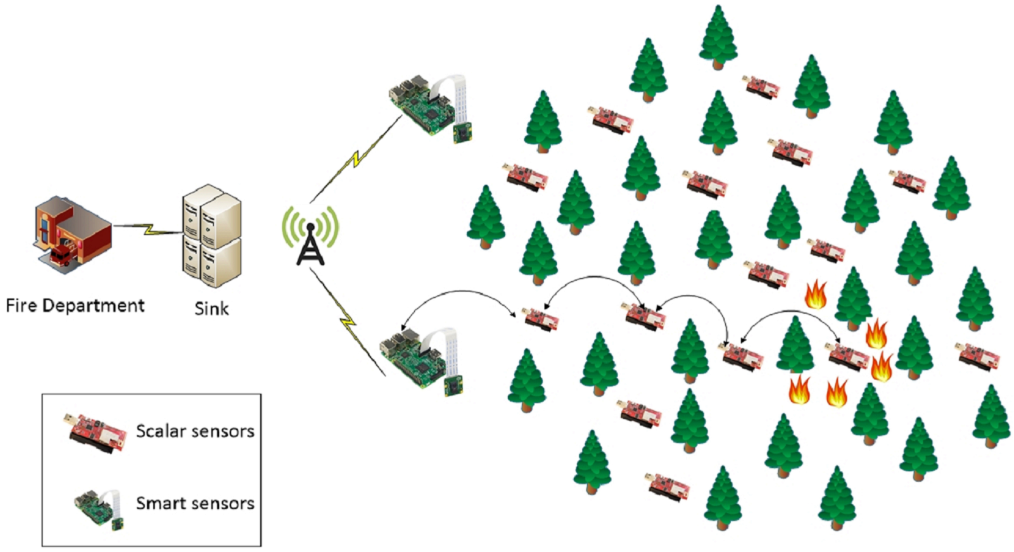


Figure 5.3: Forest fire detection system

datetime	id	hops	seq	size	temp	humid	light	volt

Figure 5.4: Transmitted packets by scalar sensors

Following this, the received packet is checked one more time by the sink node (Raspberry Pi) according to the temperature and humidity thresholds. If sink node confirms that there can be a fire because of high temperature and low humidity readings, second phase of detection is initiated. In the second phase, Raspberry Pi opens pi camera to capture an image of the environment. After capturing the image, a lightweight machine learning model is used to understand if there is a fire or not. In case the result is to have a forest fire, the fire department is informed. A flow chart is presented in Figure 5.5 for the proposed framework.

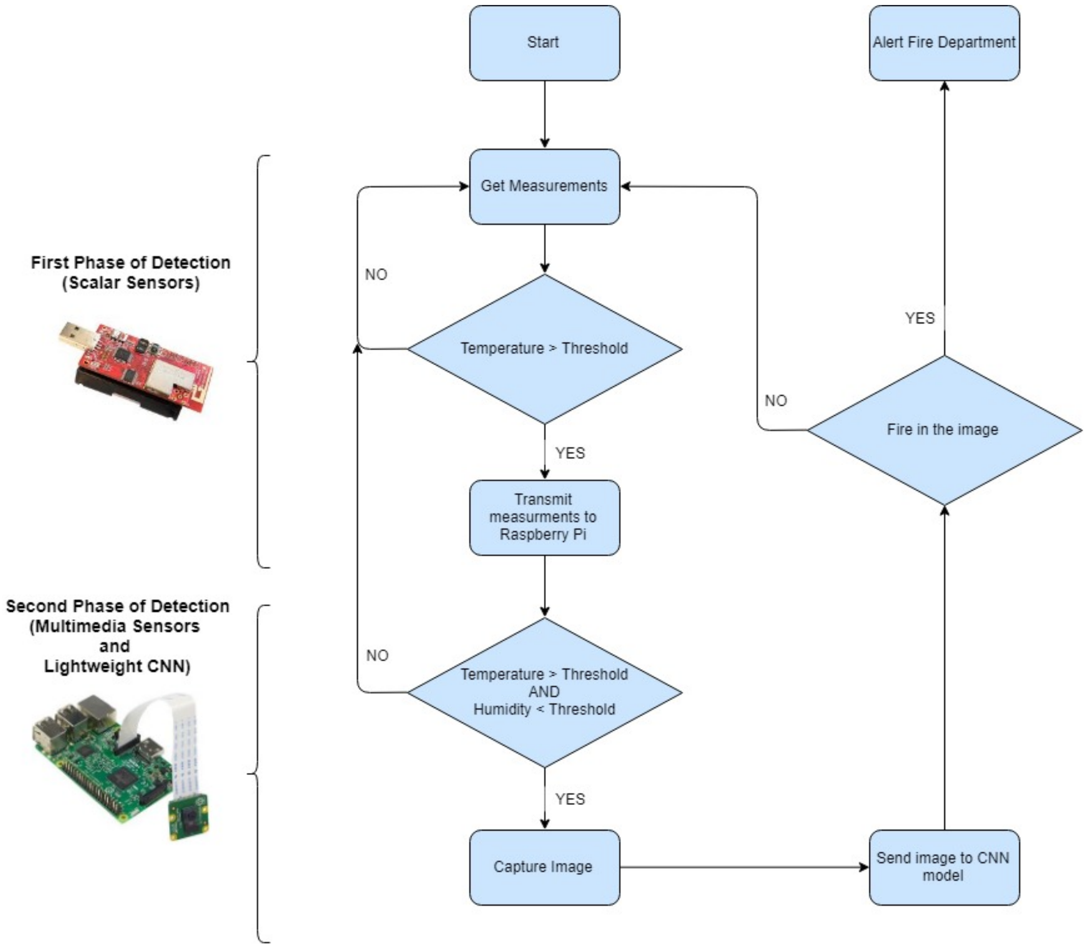


Figure 5.5: Overall system flowchart

In order to achieve energy efficiency, three novel features are employed within the presented framework for forest fire detection. The first feature is the hierarchy introduced in the system. As already discussed, scalar sensor nodes are used in the first level of hierarchy. They are much more energy efficient compared to multimedia sensor nodes since they measure and transmit scalar data. Scalar sensors are responsible for first phase of detection and multimedia sensors are not used to monitor environment unless it is necessary.

The second feature employed is heterogeneity. Heterogeneous wireless networks are more suitable solutions for surveillance and disaster management applications. In our case, forest environment conditions can be harsh to allow maintenance, deployment, and manual configuration of sensor nodes. In addition, there can be wild animals and bad weather conditions where deployed sensor nodes can be damaged or in case of fire it is possible to lose huge number of sensors. In this sense, it is better to use cheap, easily deployable, self configurable, and energy efficient sensor nodes. On the other hand, accuracy of fire detection is very crucial as well. By applying heterogeneity, energy efficiency is achieved by scalar sensors and accuracy is achieved by multimedia sensors in the proposed framework.

The last feature used to achieve energy efficiency is edge computing. As already known, the most energy consuming activity is transmission over wireless link. By processing data on edge devices (edge computing), transmission rate of the system is dramatically decreased.

In the first phase of detection, scalar sensors monitor environment by making measurements in every 10 seconds. Instead of sending packets in every 10 seconds, measurements are examined in each individual sensor node. No transmission is performed by scalar sensors unless there is a high temperature reading. By applying simple processing on the edge, we prevent our system to make millions of transmission. In the second phase of detection, CNN (Convolution Neural Network) based lightweight machine learning model is created. The main aim of the lightweight model is to perform prediction on the Raspberry Pi using captured image. Applying machine learning based process on the Raspberry Pi prevents our system to make transmission of the image data. Since its very expensive to transmit multimedia data, applying edge computing paradigm supports energy efficiency perspective of the proposed system. In addition to energy efficiency, accuracy of the system is increased by applying two phase of detection where detection is achieved both on scalar and multimedia level.

CHAPTER 6

EVALUATION OF THE SYSTEM

In this chapter, the proposed hierarchical forest fire detection system is evaluated using simulation and test bed implementation. Three scenarios are considered comparatively. In the first scenario, scalar sensors are used to detect forest fire by measuring temperature. XM1000 sensor motes are used as scalar sensors. Temperature level of forest area is monitored and in case there is an extreme temperature reading, the sink node takes the responsibility to alert the fire department. Architecture of scalar sensor scenario is depicted in Figure 6.1. The scenario considered is simulated for evaluation and a test bed is also established to consider the real system.

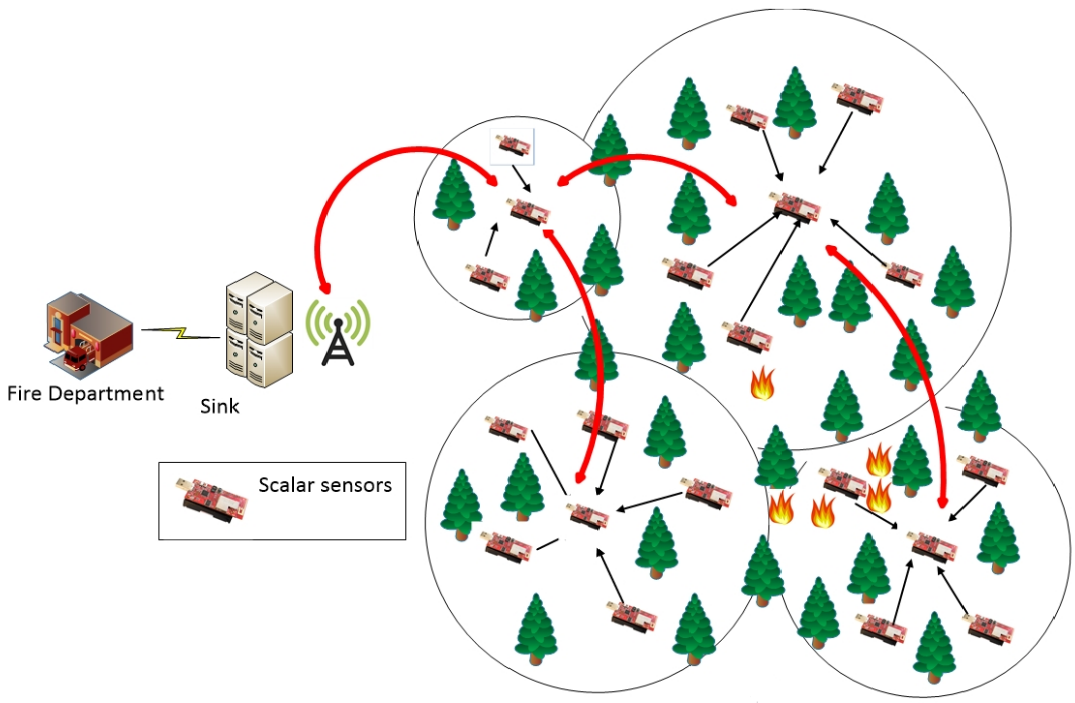


Figure 6.1: Forest fire detection using scalar sensors

Second scenario considered is multimedia sensor based forest fire detection. Raspberry Pi hardware and pi camera are used as multimedia sensors. Captured images by Raspberry Pi are transmitted to the sink node via intermediate multimedia sensors. Received images are transmitted to sink node and if there is a fire, the fire department is alerted. Scenario two is depicted using Figure 6.2.

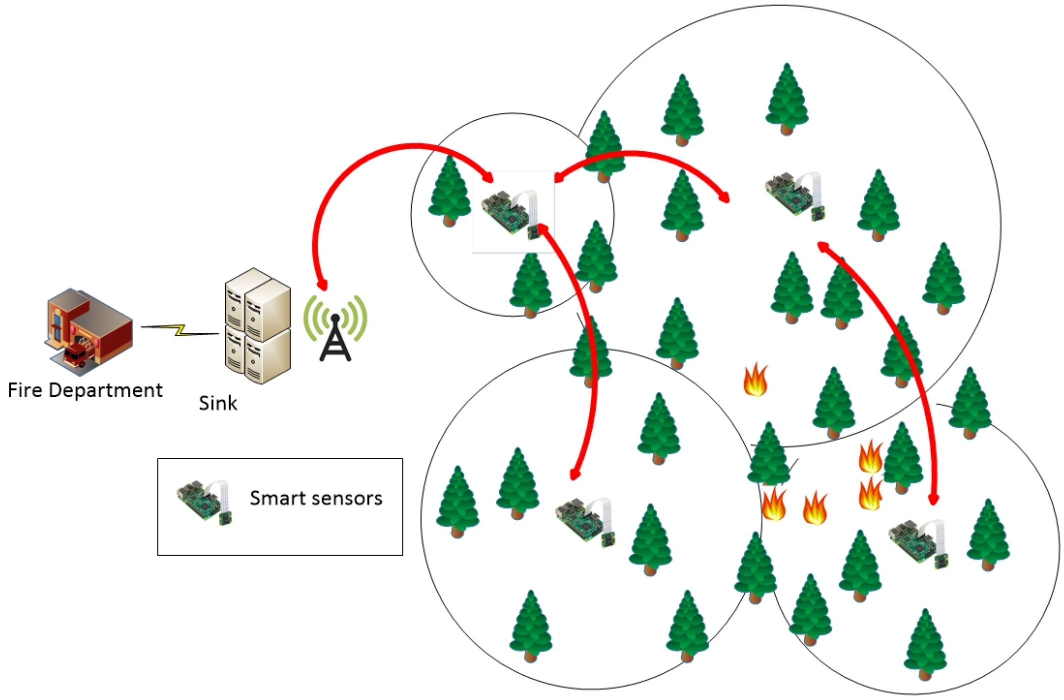


Figure 6.2: Forest fire detection using multimedia sensors

The final scenario shows the proposed hierarchical and heterogeneous system in which energy efficiency of scalar sensors and high accuracy of multimedia sensors are utilized together. As discussed earlier and depicted in Figure 5.3, scalar sensors are deployed in forest to monitor environment. Scalar sensors do not transmit anything unless there is an extreme temperature reading. Pis are used as multimedia sensors and cluster heads. In case of high temperature reading, the packets in which the contents are illustrated in Figure 5.4 are sent to the Raspberry Pi. As explained through the flow chart illustrated in Figure 5.5, Raspberry Pi checks the measurement values, and in case of high temperature and low humidity readings it opens the pi camera and captures an image. Instead of transmitting image to the sink node, the probability of fire is predicted using the proposed lightweight CNN which is discussed in Chapter

7 in detail. In case of fire, fire department is alerted. Simulations as well as test beds are employed for evaluation of all three scenarios.

6.1 Simulation

The parameters employed in simulations are given in Table 6.1. Simulation time is different for each different scenarios since it runs till the first node is dead. The sensors are deployed on a square field uniform randomly. In scalar and multimedia scenarios, there are six nodes in total where one of them is sink. In proposed model scenario, six nodes are used again, however sink node is Raspberry Pi and source nodes are XM1000 for each cluster. In scalar and multimedia scenarios, the packet rate is assumed to be 1 packet in every 10 seconds to ensure early observation of potential calamities similar to the existing studies in the literature [147], [148]. On the other hand, event triggered packet transmission is applied in the proposed model. The considered test bed scenario is also implemented using the same packet rate. The packet size of XM1000 is 200 bytes since a temperature value is sent. In case of Raspberry Pi 3, the packet size is 2 MB since the captured images are communicated. Initial energy of each node is 8.5 Wh for XM1000 sensor mote which is the initial energy of 2 AA batteries, and 65 Wh for Raspberry Pi which is the initial energy of 13000 mAh power bank. Initial energy is computed using the approach presented in [149].

Table 6.1: Simulation parameters

	XM1000	**Raspberry Pi 3**	**Proposed Approach**
Sensor Deployment	uniform random	uniform random	uniform random
Deployment Field	50x50 m	50x50 m	50x50 m
Number of nodes	6	6	6
Packet rate	0.1 pkt/s	0.1 pkt/s	Event Triggered
Packet size	200 bytes	2000000 bytes	200 bytes
Tx Output Power	0 dBm	0 dBm	0 dBm
Initial Energy	8.55 Wh	65 Wh	65 Wh
Power Consumption	8.18 mW	5000 mW	3527.8 mW

To calculate the power consumption of the XM1000, the duty cycle of the radio is calculated using the approach introduced in [118]. The following equations are used to calculate power consumption:

$$P_{cons} = V * I_m \tag{61}$$

$$I_m = \delta_{idle} * I_{idle} + \delta_{tx} * I_{tx} + \delta_{rx} * I_{rx} \tag{62}$$

$$\delta_{idle} + \delta_{tx} + \delta_{rx} = 1 \tag{63}$$

To find the power consumption P_{cons} in equation (61), the supply voltage and the mean current drawn by the sensor are calculated. To calculate the mean current drawn I_m, the time fractions δ_{idle}, δ_{tx}, δ_{rx} for each inactive state, tx, and rx are respectively calculated. As indicated in equation (63), time fractions are between 0 and 1 and represent the fraction of time during which the node remains in the corresponding state. The approach presented in [150] is commonly used to calculate the time fractions. As discussed in [150], “powertrace" library in Contiki OS can be used to observe time fractions that node stays in each state. XM1000 mote is tested using “powertrace" library in Contiki OS and time fractions for each state is extracted. As shown in Figure 6.3 , the time fractions are calculated to find mean current consumed by the sensor node. The current consumption used for each state is taken from manufacturer’s website [64].

Table 6.2 shows the parameters and the corresponding values to calculate the power consumption. I_{idle}, I_{tx}, and I_{rx} values are also retrieved from the manufacturer website [64]. I_{idle} includes current consumed by sensing and CPU operations. I_{tx}, and I_{rx} are currents used for transmitting and receiving data respectively. The mean current consumed I_m, is calculated using equation (62) and power consumption P_{cons}, is calculated using equation (61).

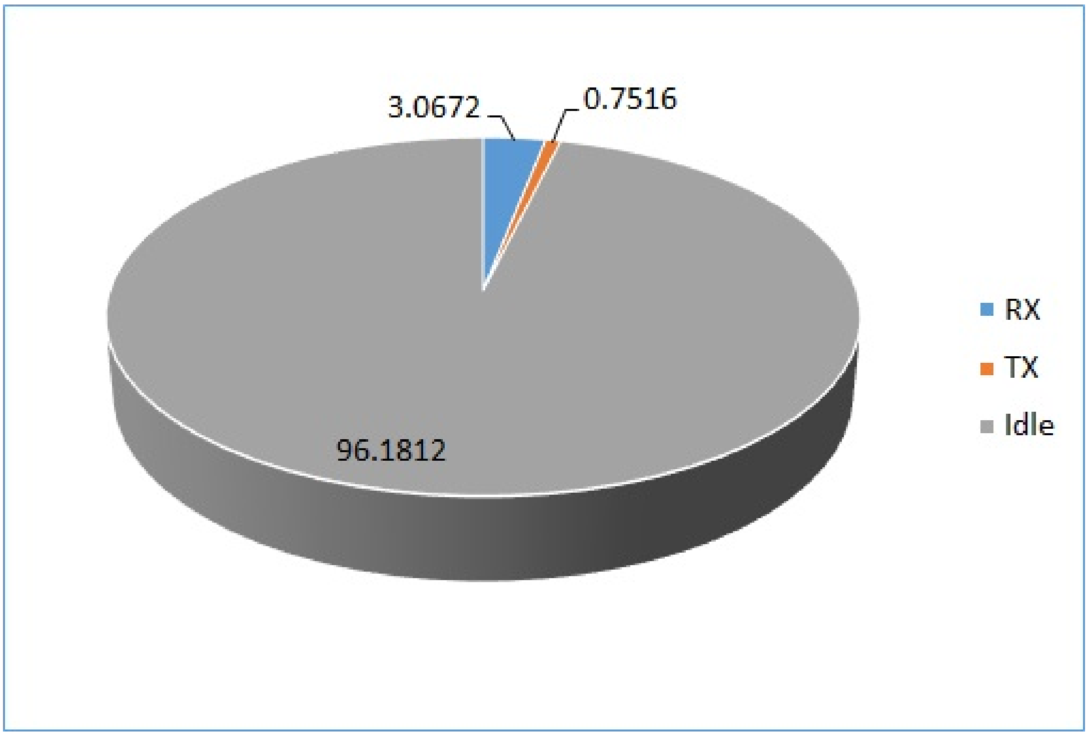


Figure 6.3: Time fractions for XM1000 (%)

Table 6.2: Calculated values for XM1000

Parameter	Value
I_{idle}	2 mA
I_{tx}	17.4 mA
I_{rx}	18.8 mA
δ_{idle}	0.9619
δ_{tx}	0.0075
δ_{rx}	0.0306
I_m	2.726 mA
P_{cons}	8.18 mW

To calculate the power consumption of the Raspberry Pi 3, the specifications provided by the manufacturer are employed. According to [146] and [151], Raspberry Pi 3 uses 700 mA without any peripherals. In addition, the pi camera consumes 250-300 mA. In total, the Raspberry Pi 3 with Pi camera consumes approximately 1000 mA of current. According to equation (64), its energy consumption is 3.50 W in idle state and 5.0 W while pi camera is open. In the multimedia scenario, 5.0 W power consumption is used for Raspberry Pi since pi camera will always be on.

$$Power(W) = Voltage(V) * Current(A) \tag{64}$$

In our proposed model, Raspberry Pi always listens the serial port. If it is triggered with high temperature and low humidity reading, it opens pi camera to capture an image. In order to find time fraction that the pi camera is on, the forest fire data for Cyprus is used from the technical report "Forest fire statistics for the period 2000-2017" [3]. Data include number of forest fires for the period 2000-2017 as shown in Figure 6.4. According to the data available, the average number of forest fires in Cyprus is 167.06 per year.

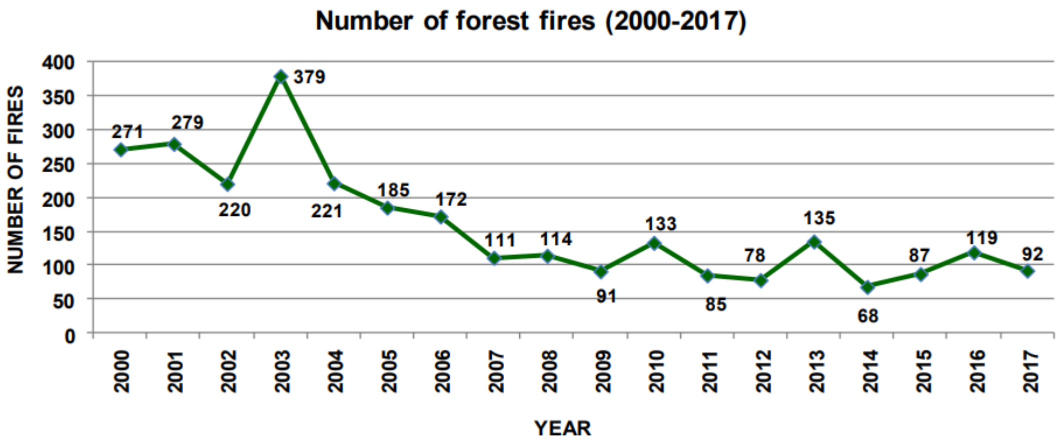


Figure 6.4: Number of forest fires between 2000-2017 [3]

By considering this statistics, we compute the time fraction of fire occurrence in the lifetime of Raspberry Pi. According to our calculations, pi camera is open in 1.85% of lifetime of proposed system as shown in Figure 6.5 and simulation parameters for all three scenarios are given in Table 6.1.

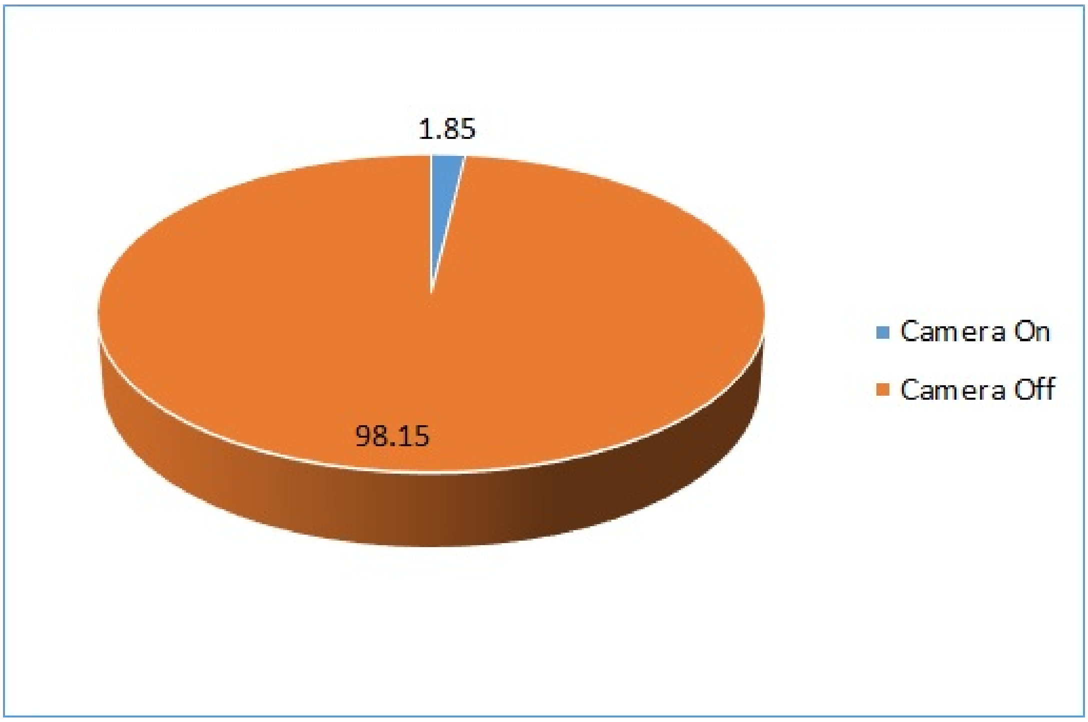


Figure 6.5: Raspberry Pi time fractions in proposed model

6.1.1 Simulation Results

Castalia simulation tool [152] is used to simulate discussed scenarios with the simulation parameters shown in Table 6.1. Castalia is based on OMNET++ platform and it is generally used for networks of low power sensor devices.

In Figure 6.6, results of scenario one (scalar) are depicted. In the figure, x-axis shows the lifetime of scalar scenario in the unit of days. On the other hand y-axis shows the scenario considered. The lifetime depicted is the amount of time from beginning to first node to die. According to simulation results, the lifetime of XM1000 sensor mote is 43.55 days with two AA batteries. Since scalar data is transmitted in this scenario, system performs well in terms of energy efficiency. However, the main constraint in this scenario is accuracy of the system. This is mainly because, the scenario with only scalar sensors is dependent on only one metric which may not be sufficient to detect forest fire accurately.

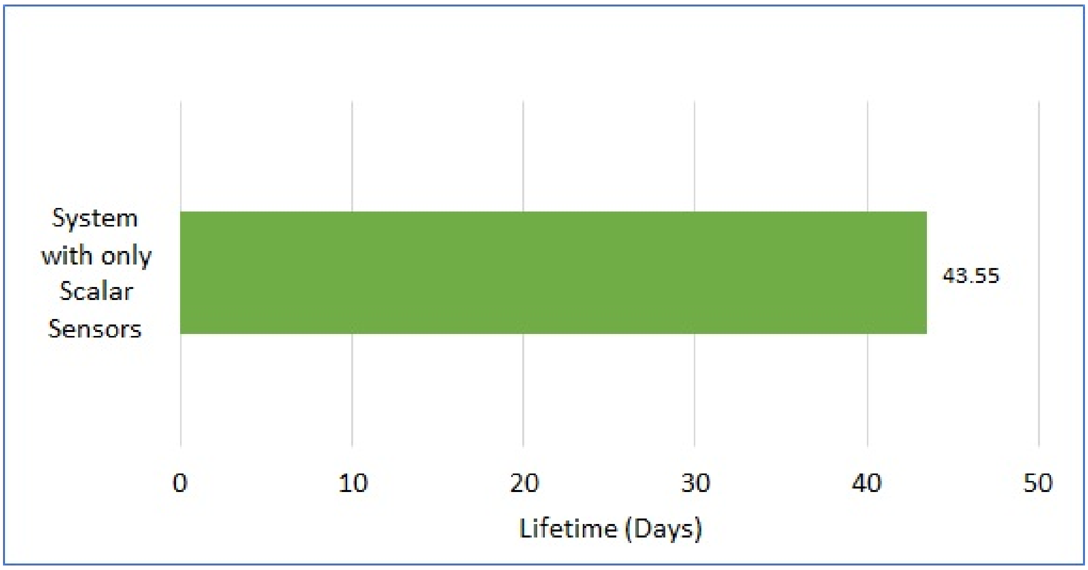


Figure 6.6: Simulation results of scalar scenario

The second and third scenarios, in which only the multimedia sensors (Raspberry Pi with pi camera) and multimedia sensors together with scalar sensors are considered respectively are also simulated in Castalia. The results of the simulations are shown in Figure 6.7 comparatively. According to simulation results, proposed framework is 29.94% more efficient than the scenario in which only the multimedia sensors are employed. Instead of monitoring environment by keeping camera always on, proposed framework uses scalar sensor readings to trigger the camera. In addition, introducing lightweight CNN model to enable edge computing on Raspberry Pi decreases unnecessary image data transmission and increases the efficiency of the system. By applying hierarchy, heterogeneity, and edge computing paradigm, we achieved more accurate system compared to scalar and more energy efficient system compared to multimedia scenario.

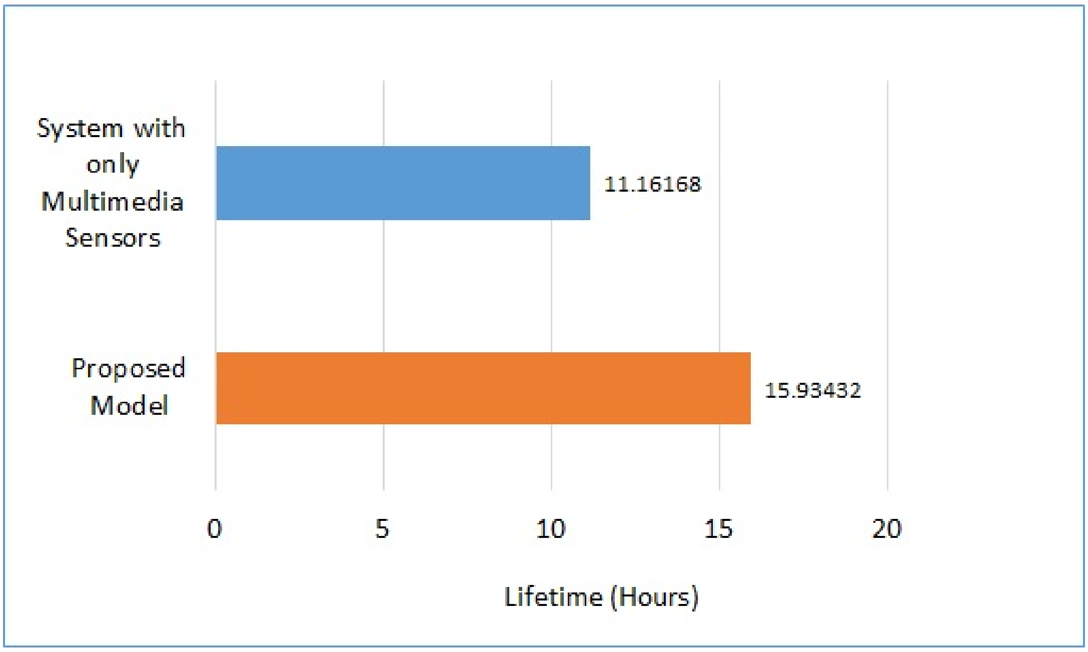


Figure 6.7: Simulation results for multimedia and proposed framework

In addition to lifetime analysis, some Quality of Service (QoS) related measures are also considered since edge computing facility has the potential to decrease the overall traffic load which would be observed in case all the messages are sent to the BS for decision making. Compared to scalar and multimedia scenarios, proposed framework performed better in terms of packet reception rate. In proposed framework, packet transmission rate is not periodic. Packet transmission takes place according to measurements of scalar sensors. Since we prevent unnecessary packet transmission, the channel traffic also decreases. In brief, by preventing unnecessary packet transmission, packet loss rate caused by interference is decreased. The results are comparatively shown in Figure 6.8. For the results presented in Figure 6.8, the packet reception rates are shown as a function of the number of nodes in the system. According to the results, the proposed framework performs significantly better considering packet reception rates especially in case of large numbers of nodes which means higher loads of traffic.

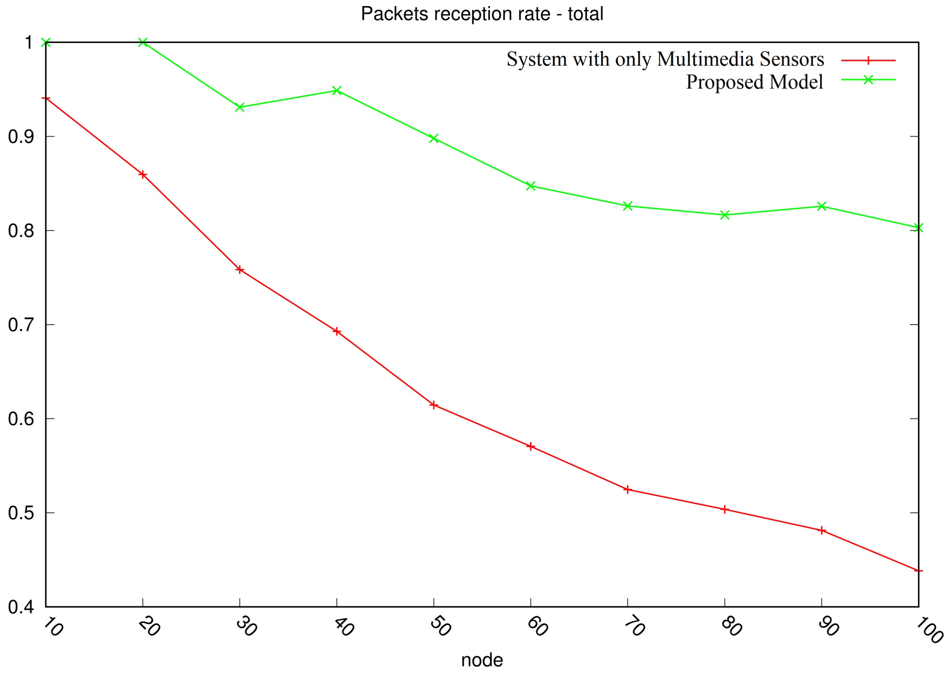


Figure 6.8: Packet reception rates

Additionally, RX packet breakdown is presented in Figure 6.9. Percentages of failed packets with interference, failed packets with non RX state, received packets despite interference, and received packets without interference are shown for multimedia scenario and the proposed model as a function of number of nodes in the system. According to results, the proposed model performs better since event triggered packet transmission is applied. Considering our case study which is forest fire detection, transmitted packet and information are very critical. Results clearly show that the proposed approach is more suitable for emergency applications since the reception process is more reliable.

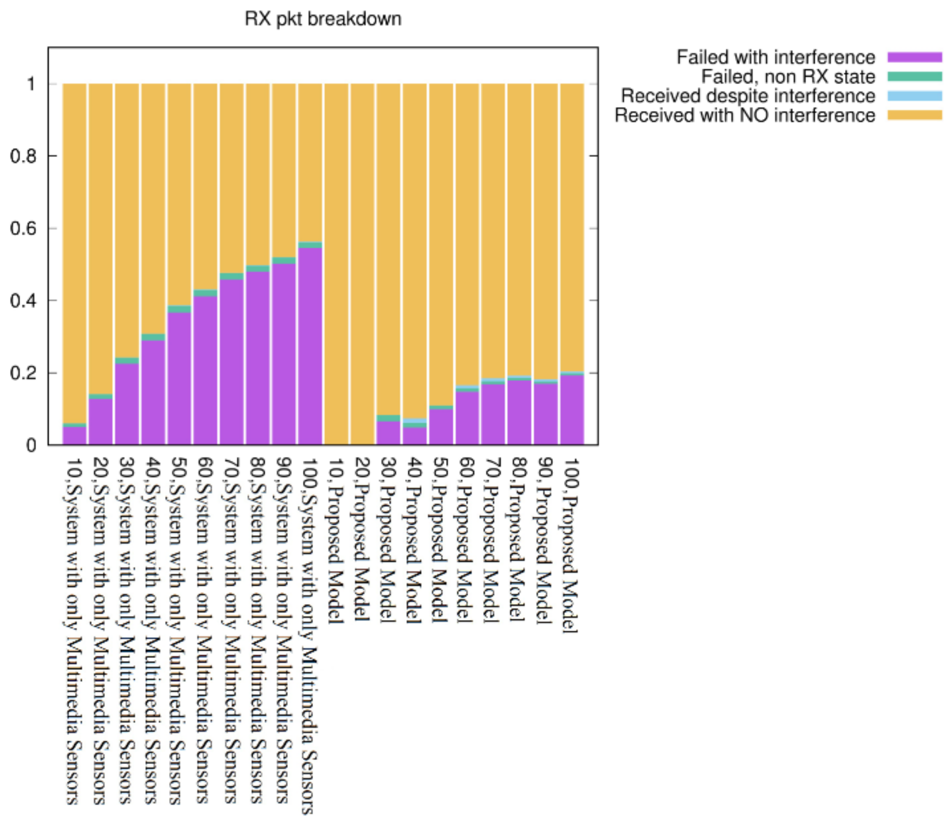


Figure 6.9: RX packet breakdown

6.2 Test Bed Implementation

The three scenarios considered in previous sections are used to conduct benchmarking experiments as well as to validate the simulation results presented for the lifetimes of sensor nodes. Figure 6.10 shows the scalar and multimedia sensors used to implement test bed.

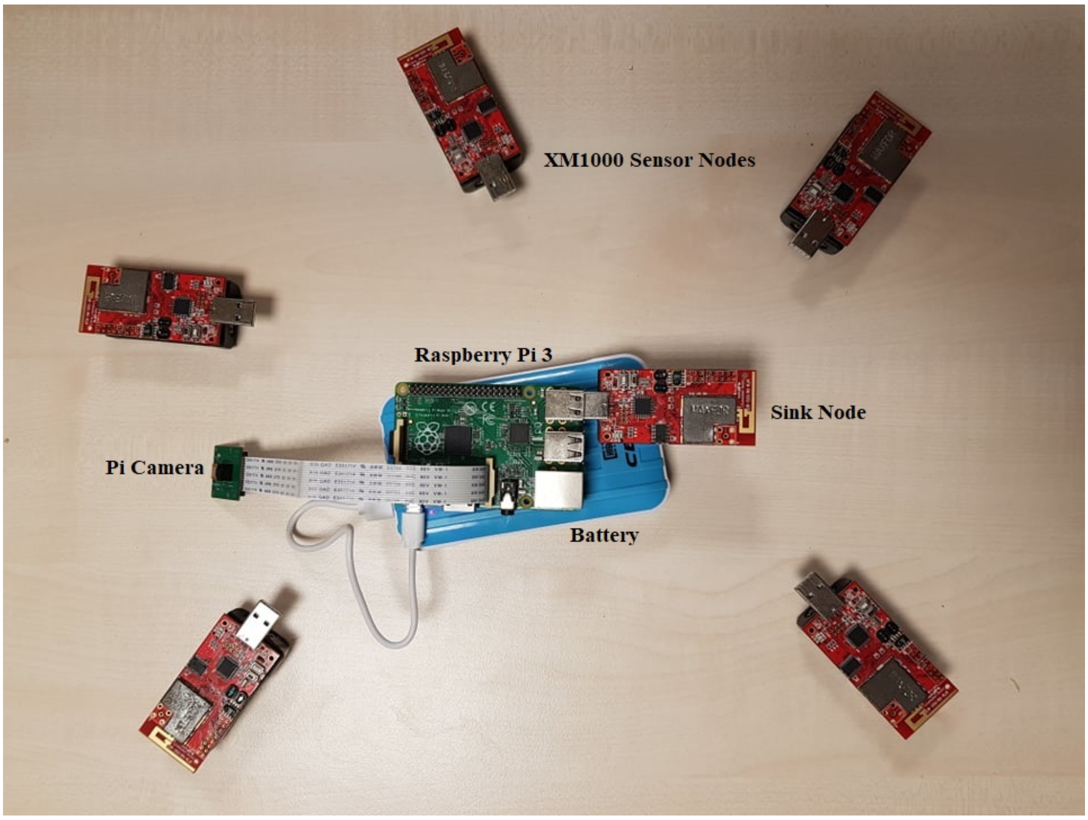


Figure 6.10: Experimental setup

As shown in the Figure 6.10, XM1000 sensor nodes are used to monitor the environment and communicate with each other using radio communication (CC2420 2.4 GHz IEEE 802.15.4 RF Transceiver). They are distributed to the lab environment uniform randomly. In order to make sure that the sensors are distributed uniformly, the area is divided to sub areas considering the number of sensors. In turn, it is guaranteed that each sub area has exactly one sensor node. One of the scalar sensors is sink and it is connected to the Raspberry Pi 3 board and communicates with Raspberry Pi using serial port. The main responsibility of sink node is to deliver received packets to the Raspberry Pi. Raspberry Pi listens the serial port for coming packets and triggers pi camera according to readings. To power Raspberry Pi, 13000 mAh power bank is used. XM1000 sensor nodes are powered by two AA batteries.

All three scenarios are implemented and lifetime analysis is conducted to validate simulation results. The voltage readings of scalar sensors scenario are depicted in Figure 6.11. To monitor the voltage levels, "battery-sensor" library is used in Contiki OS. According to test bed results, XM1000 sensor motes discharge two AA batteries in 42.50 days which is very close to our simulation results obtained as 43.55 days. The discrepancy between the simulation results and the test bed is less than 3.5%.

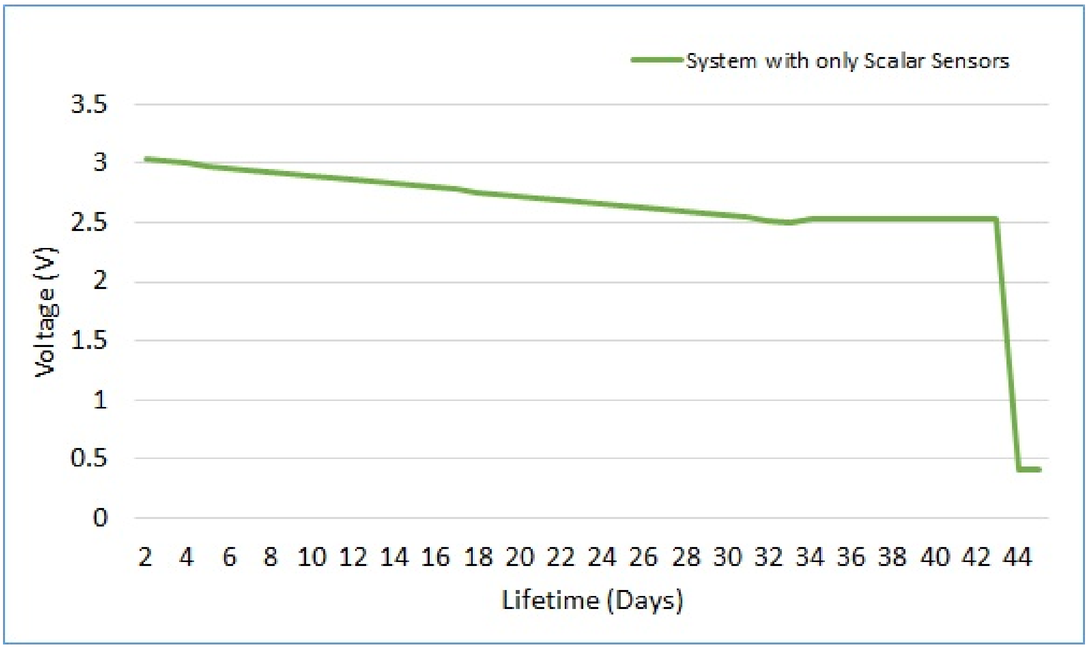


Figure 6.11: Test bed results of scalar scenario

For the second scenario, a setup is created where only the multimedia sensors are employed. As a power source, 13000 mAh power bank is used and voltage values are monitored. To monitor voltage levels Arduino Uno board is used. Basic circuitry as shown in Figure 6.12 is created to read voltage levels of battery while feeding Raspberry Pi. Arduino Uno is used to read analog signal from battery and convert it to digital signal. Converted voltage values are read by a Python script.

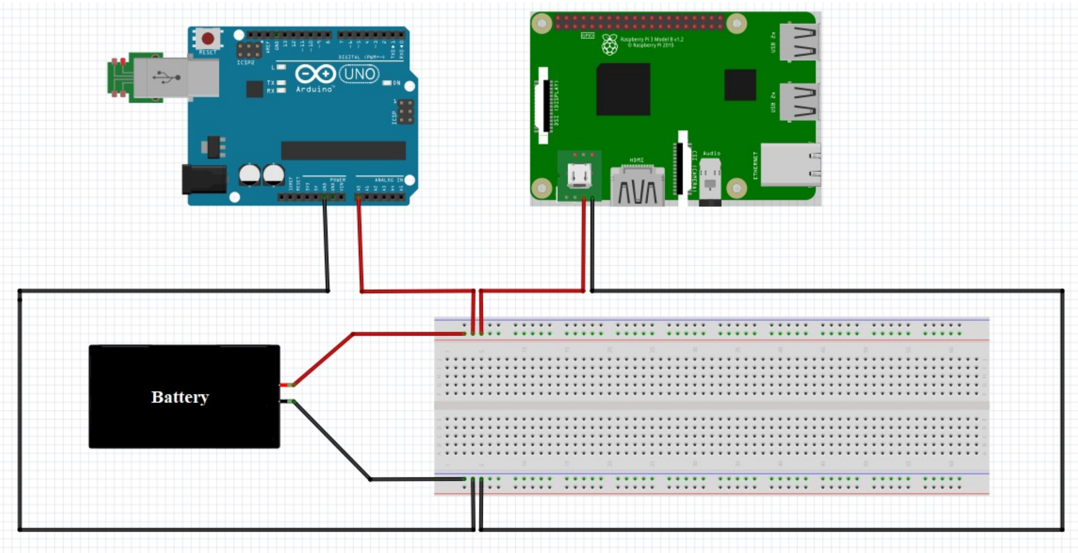


Figure 6.12: Voltage monitoring circuitry

The result of the experiment is shown in Figure 6.13. As shown in the figure, the lifetime of the multimedia sensors is 11.88 hours which is very close to simulation results which was 11.16 hours. The discrepancy between the simulation results and the test bed results is this time less than 6.1%.

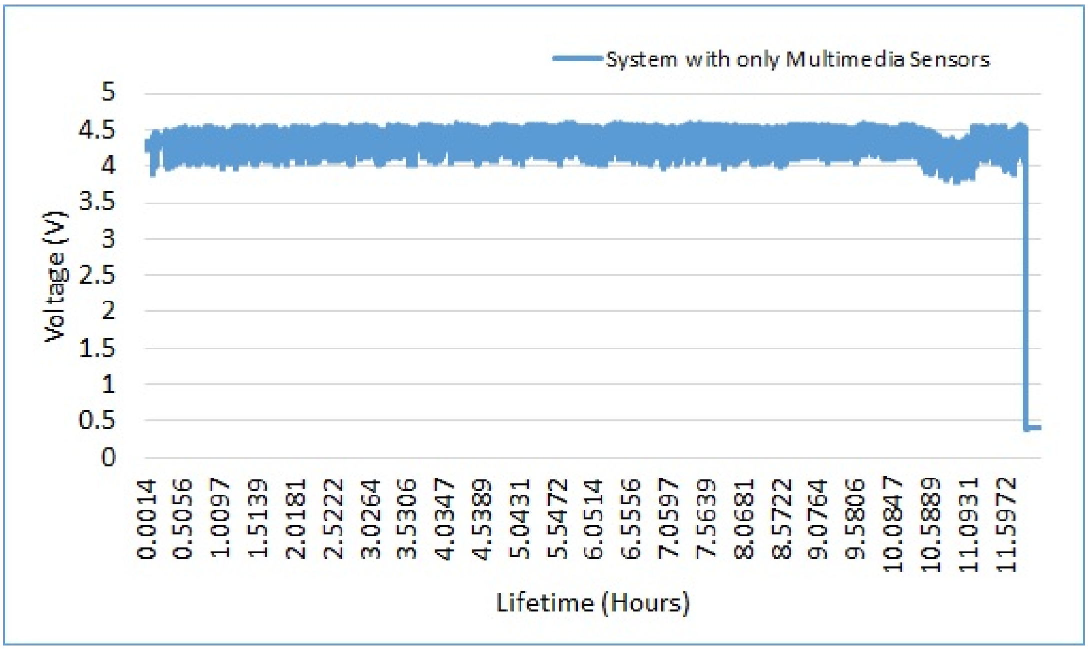


Figure 6.13: Test bed results of multimedia scenario

The proposed scenario is considered using a test bed implementation as well. The power sources used are same as previous experiments for scalar and multimedia sensors. The voltage levels are monitored in order to specify the lifetime of the motes as discussed in multimedia sensors scenario.

The results are shown in Figure 6.14. The result clearly show that compared to multimedia scenario, the proposed framework performs significantly better in terms of energy efficiency. Furthermore, the test bed also validates the simulation results. While the lifetime of the proposed model is 15.93 hours in the simulation, it is 15.66 for the test bed. The discrepancy between the results is less than 1.8%.

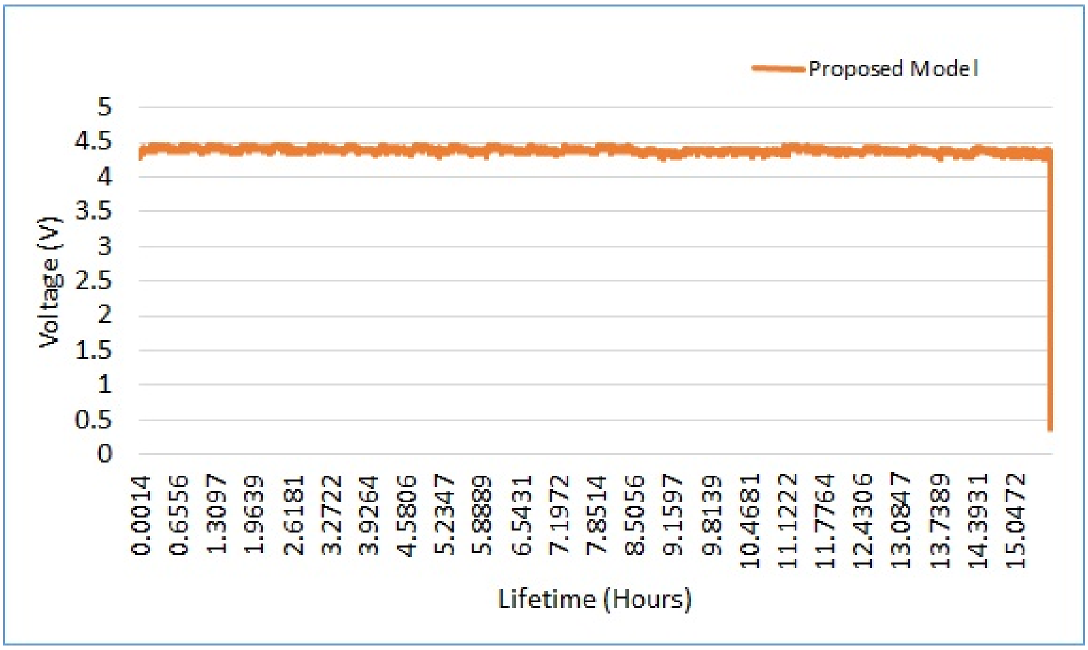


Figure 6.14: Test bed results of proposed framework

Comparative graph for multimedia and proposed framework is given in Figure 6.15 in order to further emphasize the energy efficiency of the proposed framework compared to the frameworks which are solely dependent on multimedia sensors.

In Table 6.3, results of simulation and test bed implementations for all three scenarios are summarized. As seen in the table, scalar sensors are much more energy efficient than multimedia sensors and proposed framework. However, as discussed in the previous sections, scalar sensors may not be sufficient or accurate enough for real time emergency applications.

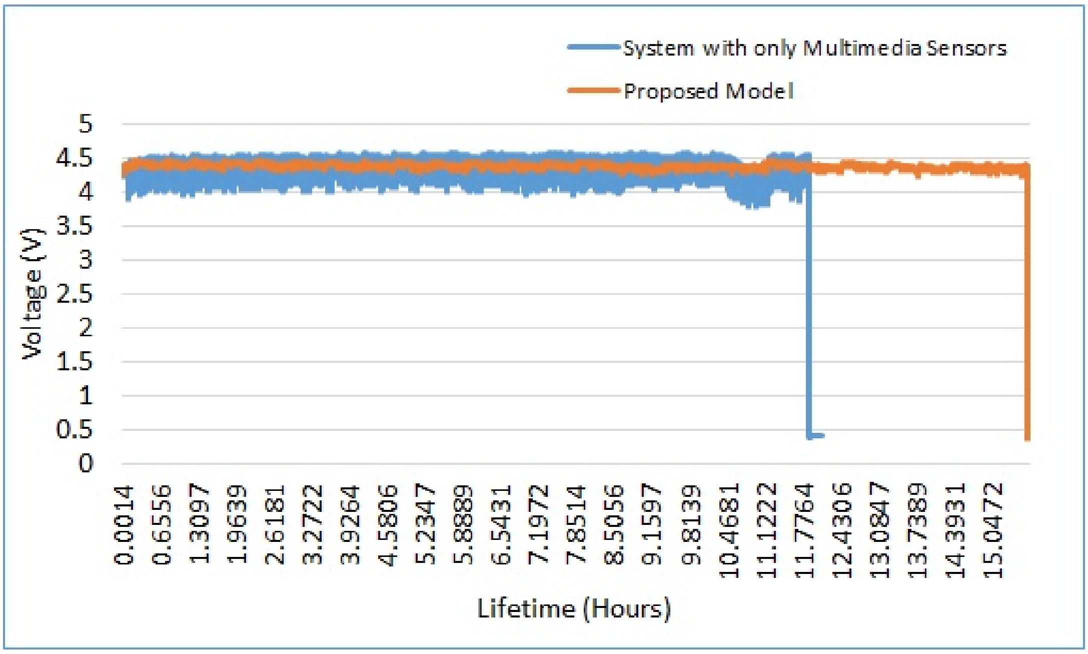


Figure 6.15: Test bed comparative results

On the other hand, with the help of larger amounts of information provided by multimedia sensors such as the image of scene, it is possible to reach the desired levels of accuracy. Since multimedia sensors are capable of sensing multimedia data, it is possible to gather more useful information from multimedia sensors by using more advanced techniques such as image processing, signal processing, and machine learning. Proposed framework makes use of efficiency of scalar sensors and accuracy of multimedia sensors. As shown in the summary table, efficiency of the system is improved by approximately 29% with the proposed framework. In Chapter 7, proposed framework is evaluated in terms of accuracy and it is shown that proposed approach is able to provide a more energy efficient solution while the accuracy is in acceptable ranges to detect forest fires.

	Scalar	**Multimedia**	**Proposed Framework**
Test Bed	42.50 days	11.88 hrs	15.66 hrs
Simulation	43.55 days	11.16 hrs	15.93 hrs
Discrepancy	$< 3.5\%$	$< 6.1\%$	$< 1.8\%$

Table 6.3: Summary of results

CHAPTER 7

FOREST FIRE DETECTION USING MACHINE LEARNING

In this chapter proposed lightweight CNN model is discussed. The main aim is to provide a robust model to classify forest fires. Since it is a crucial problem, finding autonomous solution to detect forest fires becomes inevitable. The most challenging part is the creation of the data set which is in turn employed with the machine learning model since the literature is quite limited in terms of available data. Generally, there are data sets which are used to detect fires using smoke images. Our aim is detecting fire from still images and especially to detect forest fires. To achieve the tasks specified, firstly, well-known image classification neural networks are examined and the best combination of hyper parameters along with the most suitable architecture are researched for the model. Secondly, for the data set generation, close-up forest fire videos are examined and the necessary frames are extracted with a label. Labeling operation is verified via double checking the images by hand. Once the labeling of the images with fire is completed, a similar approach is also employed for non-fire forest images. In addition to lack of data set, another constraint of our approach is the size of the model. In order to use the created CNN model at the edge of the network within the Raspberry Pi or similar hardware where resources are limited, it should be lightweight.

7.1 Dataset

In order to form an accurate model, variety of images from forest fires are required. To be able to gather those, multiple image sources (videos) are used since there is no public image data set to use for forest fire detection. Forest fire videos are collected from YouTube and sampled based on the camera's movement. Manual elimination is performed to avoid incorrect labeling. Generated data set has 3400 images in total. 1111 fire images are extracted from 16 videos and 2289 non-fire images are extracted from 9 videos. Extracted images have varying pose and illumination on different locations such as autumn forests, summer forests, winter forests, different forest fires from Canada, California, Turkey, etc. Some example images from data set are shown in Figures 7.1 and 7.2.

Figure 7.1: Fire Image Examples

Figure 7.2: Non-Fire Image Examples

7.2 Methodology

Two lightweight Convolutional Neural Network (CNN) models are proposed to classify whether a forest image contains fire or not. Even though deep learning and machine learning became popular in various image classifications tasks, forest fire detection as a specific domain has limited amount of work that proposes any neural network architecture for forest fire detection from images. In addition, as the scope of this study is forests, the widely used and studied hardware should have taken into consideration while developing any model for such systems due to computational drawbacks in those hardware platforms. Hence, this study presents two lightweight models that can work on small computers such as Raspberry Pi, Orange Pi, and Hikey 960. By using lightweight models, no transmission is required to a server to make classification task which enables any system to work on the edge with higher energy efficiency and fast response time.

In convolutional neural networks, there are four basic building blocks[153]. The first block is convolution. Every CNN model has convolutional layers. The main aim of convolutional layer is extracting features from input images. An input image is stacked three 2 dimensional matrices where each matrix has pixel values for each color (red, blue, and green). In gray scale, there is only one 2 dimensional matrix. In each convolutional layer, 2 dimensional matrices of the input image are filtered using filter or kernel matrix by sliding kernel over the matrix. This convolution operation is performed to create feature map. Feature map is the product of convolution operation. Number of filters decides how many feature map is produced at the end of each convolution layer since each filter produces different feature map. After every convolution operation, ReLU operation is used. ReLU (Rectified Linear Unit) is a non-linear operation[154]. ReLU outputs zero for negative numbers and number itself for positive numbers. The main aim of the ReLU operation is replacing all negative values with zero in image pixels. After ReLU operation, pooling operation is applied. There are different types of pooling such as max, average, and sum [155]. The main objective of the pooling step is making the input representation smaller and easily manageable. In max pooling a window size is defined (e.g. 2x2 window) and the largest element in this window is taken from the feature map. Same operation is applied in sum or average pooling, however this time it takes sum of features or average of features respectively. Each convolutional layer consists of set of convolution, ReLU, and pooling operations. After convolutional layers, CNN has fully connected layers. Fully connected layers are the multi layer perceptrons and fully connected means that every neuron in previous layer is connected to the next layer. The outputs of convolutional layers are used as input of fully connected layers. The outputs of convolution layers are high-level features of input images. These high level features are used to train the network and do classification. Briefly, convolution and pooling layers are used as feature extractors and fully connected layers are used as classifier.

The proposed models are developed and tested in Keras Framework [156]. Each model has 4 convolutional layers, and each convolutional layer is followed by a max pooling operation with a 2x2 kernel. Details of each model are described in the following subsections.

7.2.1 CNN Model 1

In the first CNN model, 64x64 image is used. It has four convolutional layers where each convolutional layer is followed by a max pooling layer. In max pooling operation 2x2 kernel is used. Proposed lightweight CNN model has three fully connected layers after convolutional layers. As shown in Figure 7.3, proposed CNN model is kept as shallow as possible considering the number of convolutional and fully connected layers since the deeper networks are more complex in terms of time and memory complexity [157]. Presented lightweight CNN model is trained using newly created forest fire data set. 80-20% train test split ratio is applied. Stochastic gradient descent is used as an optimizer since it is widely used in CNN applications and studies such as [158], [159], and [160]. As an activation function RELU (Rectified Linear Unit) function is used.

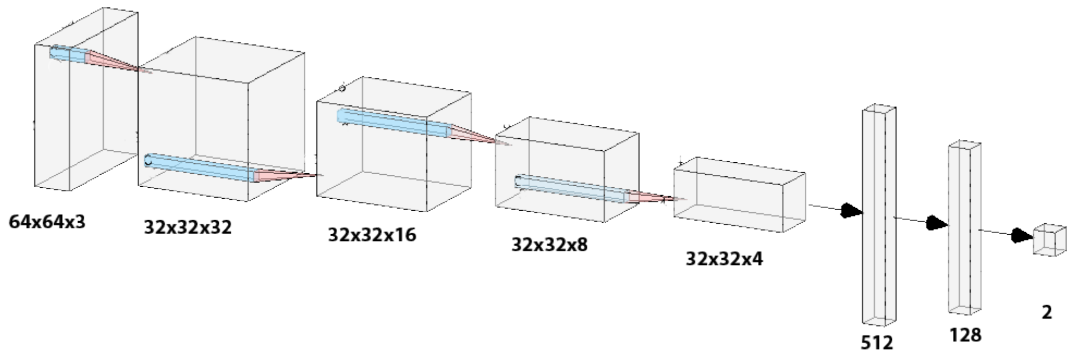


Figure 7.3: CNN architecture of model 1

Details of the Architecture;

- **Image Size:** 64x64
- **Batch Size:** 32
- **Number of Convolutional Layers:** 4
- **Number of Fully Connected Layers:** 3
- **Dropout Rate:** 0.25
- **Train-Test Split Ratio:** 80-20%

- **Pooling:** Max Pooling(2x2)
- **Optimizer:** Stochastic Gradient Descent
- **Activation Function:** RELU
- **Loss Function:** Cross-Entropy Loss
- **Learning Rate:** 0.01
- **Epochs:** 100
- **Early Stop Condition:** No decrease in validation loss for ten consecutive epochs.

7.2.2 CNN Model 2

Second proposed CNN model has one additional fully connected layer compared to the first model. Max pooling is applied with 2x2 kernel. It uses 64x64 image size as well. Same forest fire data set is used to train the second model. Train test split ratio is 80-20% with stochastic gradient descent optimizer, RELU activation function, and cross entropy loss function. Since second model has an additional fully connected layer, accuracy of detection is increased. However, second model is more complex than the first one.

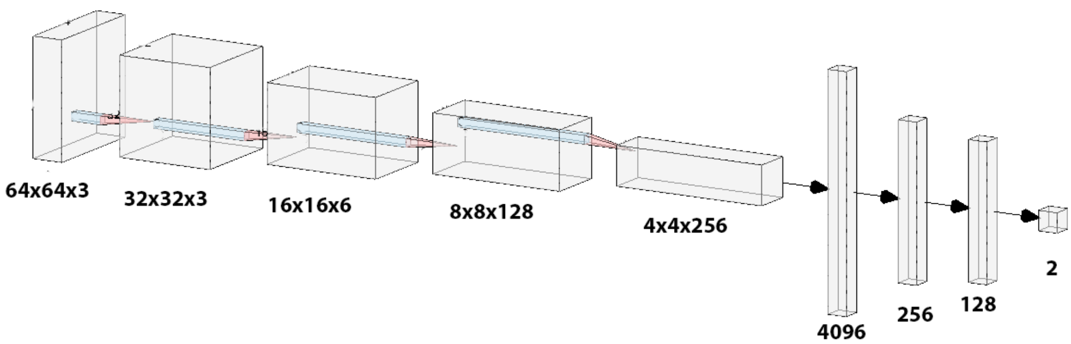


Figure 7.4: CNN architecture of model 2

Details of the Architecture;

- **Image Size:** 64x64
- **Batch Size:** 32
- **Number of Convolutional Layers:** 4
- **Number of Fully Connected Layers:** 4
- **Dropout Rate:** 0.25
- **Train-Test Split Ratio:** 80-20%
- **Pooling:** Max Pooling(2x2)
- **Optimizer:** Stochastic Gradient Descent
- **Activation Function:** RELU
- **Loss Function:** Cross-Entropy Loss
- **Learning Rate:** 0.01
- **Epochs:** 100
- **Early Stop Condition:** No decrease in validation loss for ten consecutive epochs.

7.3 Results

As discussed in previous sections, new data set is presented which contains forest images with fire and without fire along with two lightweight CNN models to classify those images. To achieve so, a single experiment to test proposed models and various well-known models such as Resnet50, DenseNet, and VGG16 [161] using transfer learning is conducted. In addition, to make sure that the models presented are not overfitting, 10-Fold Cross validation is implemented. All the results are presented in Table 7.1 and Table 7.2.

Model	Accuracy Score (%)
Model 1	99.12
Model 2	99.56
ResNet50	83.70
DenseNet	99.00
VGG16	98.00

Table 7.1: Accuracy scores of the experiment with different models

To increase the forest fire detection accuracy of the models, created data set is enriched by adding external manually found different images. These images are completely different from videos that are collected. By increasing diversity of the images, accuracy of detection is also increased. For Model 1, 98.70% detection accuracy is achieved where it is 99.50% for Model 2. Considering similar studies in the literature such as [162] with 97%, [163] with 96.62%, and [164] with 91.96% fire detection accuracy results, proposed models are quite promising.

	Model 1	Model 2
Mean Accuracy (%)	98.70	99.50
Mean Validation Accuracy (%)	93.20	99.00
Mean Loss	0.03556	0.01520
Mean Validation Loss	0.22593	0.02951

Table 7.2: 10-Fold cross validation results of both models

This chapter focuses on two main tasks where the first one is generating a data set and the latter is building a proper model for classification. Although the obtained accuracy results are very high, they might be biased because of the data set as the data set contains limited number of images for each class. On the other hand, the model is also tested with random image sets for both classes and obtained good predictions for those random image sets. In addition, a validation set is used to to minimize overfitting probability. In other words, model is saved considering the decrease in validation loss, not the training loss. Obviously, the most challenging part is collecting data since there is no available image data set specifically for forest fire detection. Especially, close-up videos for forest fires are very rare. Available forest fire images and videos are generally from UAVs, helicopters, and airplanes. Compared to forest fire images, it is easier to find no fire forest images with different pose, illumination, location, and season. In conclusion, considering other studies in this area, this study contributes to literature by proposing lightweight CNN models which enable edge computing in turn, accuracy and efficacy in energy aware forest fire detection systems. As a future work, it is possible to achieve more accurate and generic lightweight machine learning models by enriching our image data set with more comprehensive data.

CHAPTER 8

CONCLUSIONS

In this thesis, an energy efficient hierarchical approach is introduced for forest fire detection. Unlike the existing studies, multimedia sensors with machine learning algorithms are employed together with scalar sensors and the detection is performed using fusion of information in various levels. Furthermore, the efficiency of the communication is improved by introducing edge computing for decision making. Proposed framework makes use of efficiency of scalar sensors and accuracy of multimedia sensors. Scalar sensors continue operation for longer duration compared to multimedia sensors, however since it is solely based on the conditions such as temperature, humidity, and light, the detection accuracy may not reach to desirable levels. By applying hierarchy, the new framework balances the energy efficacy and accuracy of detection and offers a sustainable emergency monitoring system. As a case study, forest fire detection system is presented. To achieve proposed tasks, comprehensive literature review is conducted. Existing approaches are studied critically. Forest fire detection and environment monitoring systems are analyzed comparatively. Proposed energy aware approaches, type of sensors used, applied machine learning techniques, and applied statistical techniques are investigated. Proposed framework is evaluated using simulation and real life experiments. Simulation and real life experiment results are presented and discussed comparatively. According to the results of the study, 29.94% energy saving is achieved compared to multimedia sensor based surveillance systems. Moreover, a new machine learning model using CNN is proposed to enable processing on edge devices. The main constraint with CNN model is that it should be lightweight model so that it can run on devices which have limited resources. In addition, collecting image data to train and test proposed model is another challenging part of the study. According to test results of the model, 98.20% validation accuracy is achieved.

To investigate factors which affect the lifetime of sensor nodes, statistical analysis is conducted. Linear regression and ordered logit regression models are employed. Results of the analysis are presented and discussed.

In conclusion, proposed approach is evaluated in terms of energy efficiency and accuracy. Achieved improvements are discussed in detail in corresponding chapters. As future works, proposed approach can be tested using more comprehensive setup since six XM1000 and one Raspberry Pi 3 are used in this study. In addition, proposed machine learning model can be further improved by expanding the employed data set further. For the statistical analysis, more complex models can be created for better results. Our aim is conducting survival analysis as the next step.

Bibliography

[1] Tifenn Rault, Abdelmadjid Bouabdallah, and Yacine Challal. Energy efficiency in wireless sensor networks: A top-down survey. *Computer Networks*, 67:104–122, 2014.

[2] Girish Bekaroo, Chandradeo Bokhoree, and Colin Pattinson. Impacts of ict on the natural ecosystem: A grassroot analysis for promoting socio-environmental sustainability. *Renewable and Sustainable Energy Reviews*, 57:1580–1595, 2016.

[3] Forest fire statistics for the period 2000-2017. http://www.moa.gov.cy, 2018.

[4] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[5] Carlos F García-Hernández, Pablo H Ibarguengoytia-Gonzalez, Joaquín García-Hernández, and Jesús A Pérez-Díaz. Wireless sensor networks and applications: a survey. *IJCSNS International Journal of Computer Science and Network Security*, 7(3):264–273, 2007.

[6] Th Arampatzis, John Lygeros, and Stamatis Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, pages 719–724. IEEE, 2005.

[7] Ian F Akyildiz, Tommaso Melodia, and Kaushik R Chowdhury. A survey on wireless multimedia sensor networks. *Computer networks*, 51(4):921–960, 2007.

[8] Ian F Akyildiz, Tommaso Melodia, and Kaushik R Chowdury. Wireless multimedia sensor networks: A survey. *IEEE Wireless Communications*, 14(6):32–39, 2007.

[9] Rob Holman, John Stanley, et al. Applying video sensor networks to nearshore environment monitoring. *IEEE Pervasive Computing*, (4):14–21, 2003.

[10] Fei Hu and Sunil Kumar. Qos considerations in wireless sensor networks for telemedicine. In *Internet Multimedia Management Systems IV*, volume 5242, pages 217–228. International Society for Optics and Photonics, 2003.

[11] Jason Campbell, Phillip B Gibbons, Suman Nath, Padmanabhan Pillai, Srinivasan Seshan, and Rahul Sukthankar. Irisnet: an internet-scale architecture for multimedia sensors. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 81–88. ACM, 2005.

[12] AA Reeves et al. Remote monitoring of patients suffering from early symptoms of dementia. In *Intl. Workshop on Wearable and Implantable Body Sensor Networks, London, UK*, 2005.

[13] Molly Webb. Smart 2020: enabling the low carbon economy in the information age, a report by the climate group on behalf of the global esustainability initiative (gesi). *Creative Commons*, 2008.

[14] https://edition.cnn.com/2018/07/23/europe/athens-wildfires-intl/index.html CNN. Athens wildfire. `https://edition.cnn.com/2018/07/23/europe/athens-wildfires-intl/index.html`, 2018.

[15] Wildfires in California. California wildfires. `https://en.wikipedia.org/wiki/ListofCaliforniawildfires`, 2018.

[16] Kuralay Mazarzhanova, Arailym Kopabayeva, NESİBE KÖSE, and ÜNAL AKKEMİK. The first forest fire history of the burabai region (kazakhstan) from tree rings of pinus sylvestris. *Turkish Journal of Agriculture and Forestry*, 41(3):165–174, 2017.

[17] Priyanka Rawat, Kamal Deep Singh, Hakima Chaouchi, and Jean Marie Bonnin. Wireless sensor networks: a survey on recent developments and potential synergies. *The Journal of supercomputing*, 68(1):1–48, 2014.

[18] Ishfaq Ahmad, Khalil Shah, and Saif Ullah. Military applications using wireless sensor networks: A survey. *Int. J. Eng. Sci*, 6(6):7039, 2016.

[19] Milica Pejanović Ðurišić, Zhilbert Tafa, Goran Dimić, and Veljko Milutinović. A survey of military applications of wireless sensor networks. In *2012 Mediterranean conference on embedded computing (MECO)*, pages 196–199. IEEE, 2012.

[20] Ismail Butun, Salvatore D Morgera, and Ravi Sankar. A survey of intrusion detection systems in wireless sensor networks. *IEEE communications surveys & tutorials*, 16(1):266–282, 2013.

[21] Kevin G Kinsella and David R Phillips. *Global aging: The challenge of success*, volume 60. Population Reference Bureau Washington, DC, 2005.

[22] Hande Alemdar and Cem Ersoy. Wireless sensor networks for healthcare: A survey. *Computer networks*, 54(15):2688–2710, 2010.

[23] Minh-Thanh Vo, TT Thanh Nghi, Van-Su Tran, Linh Mai, and Chi-Thong Le. Wireless sensor network for real time healthcare monitoring: network design and performance evaluation simulation. In *5th International Conference on Biomedical Engineering in Vietnam*, pages 87–91. Springer, 2015.

[24] Shah Haque, Mustafizur Rahman, and Syed Aziz. Sensor anomaly detection in wireless sensor networks for healthcare. *Sensors*, 15(4):8764–8786, 2015.

[25] Prosanta Gope and Tzonelih Hwang. Bsn-care: A secure iot-based modern healthcare system using body sensor network. *IEEE sensors journal*, 16(5):1368–1376, 2015.

[26] Taiyang Wu, Fan Wu, Jean-Michel Redouté, and Mehmet Rasit Yuce. An autonomous wireless body area network implementation towards iot connected healthcare applications. *Ieee Access*, 5:11413–11422, 2017.

[27] Samaher Al-Janabi, Ibrahim Al-Shourbaji, Mohammad Shojafar, and Shahaboddin Shamshirband. Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications. *Egyptian Informatics Journal*, 18(2):113–122, 2017.

[28] Mohd Fauzi Othman and Khairunnisa Shazali. Wireless sensor network applications: A study in environment monitoring system. *Procedia Engineering*, 41:1204–1210, 2012.

[29] Roland Kays, Sameer Tilak, Margaret Crofoot, Tony Fountain, Daniel Obando, Alejandro Ortega, Franz Kuemmeth, Jamie Mandel, George Swenson, Thomas Lambert, et al. Tracking animal location and activity with an automated radio telemetry system in a tropical rainforest. *The Computer Journal*, 54(12):1931–1948, 2011.

[30] Soledad Escolar Díaz, Jesús Carretero Pérez, Alejandro Calderón Mateos, Maria-Cristina Marinescu, and Borja Bergua Guerra. A novel methodology for the monitoring of the agricultural production process based on wireless sensor networks. *Computers and electronics in agriculture*, 76(2):252–265, 2011.

[31] Yingli Zhu, Jingjiang Song, and Fuzhou Dong. Applications of wireless sensor network in the agriculture environment monitoring. *Procedia Engineering*, 16:608–614, 2011.

[32] Siuli Roy, D Anurag, and Somprakash Bandyopadhyay. Testbed implementation of a pollution monitoring system using wireless sensor network for the protection of public spaces. In *Networking and Telecommunications: Concepts, Methodologies, Tools, and Applications*, pages 820–833. IGI Global, 2010.

[33] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97. Acm, 2002.

[34] M López, S Martinez, JM Gomez, A Herms, L Tort, J Bausells, and A Errachid. Wireless monitoring of the ph, nh4+ and temperature in a fish farm. *Procedia Chemistry*, 1(1):445–448, 2009.

[35] Mingfei Zhang, Daoliang Li, Lianzhi Wang, Daokun Ma, and Qisheng Ding. Design and development of water quality monitoring system based on wireless sensor network in aquaculture. In *International Conference on Computer and Computing Technologies in Agriculture*, pages 629–641. Springer, 2010.

[36] Carles Gomez and Josep Paradells. Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, 48(6):92–101, 2010.

[37] Muhammad Raisul Alam, Mamun Bin Ibne Reaz, and Mohd Alauddin Mohd Ali. A review of smart homes—past, present, and future. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1190–1203, 2012.

[38] Ian F Akyildiz, Tommaso Melodia, and Kaushik R Chowdhury. Wireless multimedia sensor networks: Applications and testbeds. *Proceedings of the IEEE*, 96(10):1588–1605, 2008.

[39] Kapileswar Nellore and Gerhard Hancke. A survey on urban traffic management system using wireless sensor networks. *Sensors*, 16(2):157, 2016.

[40] Konstantinos Kanistras, Goncalo Martins, Matthew J Rutherford, and Kimon P Valavanis. Survey of unmanned aerial vehicles (uavs) for traffic monitoring. *Handbook of unmanned aerial vehicles*, pages 2643–2666, 2015.

[41] Michal Kochláň, Michal Hodoň, Lukáš Čechovič, Ján Kapitulík, and Matúš Jurečka. Wsn for traffic monitoring using raspberry pi board. In *2014 Federated Conference on Computer Science and Information Systems*, pages 1023–1026. IEEE, 2014.

[42] Huu-Quoc Nguyen, Ton Thi Kim Loan, Bui Dinh Mao, and Eui-Nam Huh. Low cost real-time system monitoring using raspberry pi. In *2015 Seventh International Conference on Ubiquitous and Future Networks*, pages 857–859. IEEE, 2015.

[43] Mohammad Mehedi Hassan, Kai Lin, Xuejun Yue, and Jiafu Wan. A multimedia healthcare data sharing approach through cloud-based body area network. *Future Generation Computer Systems*, 66:48–58, 2017.

[44] Lorena Parra, Sandra Sendra, José Miguel Jiménez, and Jaime Lloret. Multimedia sensors embedded in smartphones for ambient assisted living and e-health. *Multimedia Tools and Applications*, 75(21):13271–13297, 2016.

[45] Shamma Alqassim, Madhumeta Ganesh, Shaheen Khoja, Meher Zaidi, Fadi Aloul, and Assim Sagahyroon. Sleep apnea monitoring using mobile phones. In *2012 IEEE 14th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 443–446. IEEE, 2012.

[46] Abderrahim Bourouis, Ali Zerdazi, Mohammed Feham, and Abdelhamid Bouchachia. M-health: skin disease analysis system using smartphone's camera. *Procedia Computer Science*, 19:1116–1120, 2013.

[47] Abhilasha Tyagi, Kimberly Miller, and Myles Cockburn. e-health tools for targeting and improving melanoma screening: a review. *Journal of skin cancer*, 2012, 2012.

[48] Tarun Wadhawan, Ning Situ, Hu Rui, Keith Lancaster, Xiaojing Yuan, and George Zouridakis. Implementation of the 7-point checklist for melanoma detection on smart handheld devices. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3180–3183. IEEE, 2011.

[49] Salvatore Manfreda, Matthew McCabe, Pauline Miller, Richard Lucas, Victor Pajuelo Madrigal, Giorgos Mallinis, Eyal Ben Dor, David Helman, Lyndon Estes, Giuseppe Ciraolo, et al. On the use of unmanned aerial systems for environmental monitoring. *Remote Sensing*, 10(4):641, 2018.

[50] Abbas Javed, Hadi Larijani, Ali Ahmadinia, Rohinton Emmanuel, Mike Mannion, and Des Gibson. Design and implementation of a cloud enabled random neural network-based decentralized smart controller with intelligent sensor nodes for hvac. *IEEE Internet of Things Journal*, 4(2):393–403, 2016.

[51] Jan Beutel, Oliver Kasten, and Matthias Ringwald. Btnodes–a distributed platform for sensor nodes. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 292–293. ACM, 2003.

[52] Michael Healy, Thomas Newe, and Elfed Lewis. Wireless sensor node hardware: A review. In *SENSORS, 2008 IEEE*, pages 621–624. IEEE, 2008.

[53] Fatma Karray, Mohamed W Jmal, Alberto Garcia-Ortiz, Mohamed Abid, and Abdulfattah M Obeid. A comprehensive survey on wireless sensor node hardware platforms. *Computer Networks*, 144:89–110, 2018.

[54] Naveen Madabhushi. Kmote-design and implementation of a low cost, low power hardware platform for wireless sensor networks. *Indian Institute of Technology*, 2007.

[55] AS Advanticsys. Xm1000 resource manual.

[56] TSmoTe. Tecnologas, s.t.y. sistemas, s. a., tsmote: development tool for wireless monitoring, remote control and m2m applications, 2008. `http://www.tst-sistemas.es/Docs/TSmoTe-EN.pdf.`, 2008.

[57] IRIS datasheet. Iris, memsic, 2007. rev a,. `http://www.memsic.com/userfiles/files/Datasheets/WSN/IRIS_Datasheet.pdf.`, 2007.

[58] WiSMote. a. systems, wismote, 2012,. `http://www.aragosystems.fr/produits/wisnet/wismote/.`, 2012.

[59] WiSense. Wisense, system overview, 2014. `http://www.wisense.in/api/html/SystemOverview.html.`, 2012.

[60] LOTUS. Memsic, lotus - next-generation mote platform for high-performance wireless sensor networks, 2011. `https://globenewswire.com/news-release/2011/04/28/445684/220233/en/MEMSIC-Introduces-LOTUS-Next-Generation-Mote-Platform/-for-High/-Performance-Wireless-Sensor-Networks.html.`, 2008.

[61] Node+. Variable, node+ ios and android wireless sensor platform, 2015., 2015.

[62] Michele Magno, Davide Brunelli, Lukas Sigrist, Renzo Andri, Lukas Cavigelli, Andres Gomez, and Luca Benini. Infinitime: Multi-sensor wearable bracelet with human body harvesting. *Sustainable Computing: Informatics and Systems*, 11:38–49, 2016.

[63] Mago-Node+. Node+ ios and android wireless sensor platform, variable, 2011. `http://media.wix.com/ugd/54926a_d74fdd41d95d41218dfa5900e727ca4e.pdf.`, 2011.

[64] As-xm1000 802.15.4 mote module including temperature, humidity and light sensor with upgraded 116kb-eeprom product model: Xm1000. `https://www.advanticsys.com/shop/asxm1000-p-24.html`, 2018.

[65] B Kontagora Nuhu, OT Arulogun, IA Adeyanju, and IM Abdullah. Wireless sensor network for real-time flood monitoring based on 6lowpan communication standard. *APTIKOM Journal on Computer Science and Information Technologies, ISSN*, pages 2528–2417, 2016.

[66] Payam Barnaghi and Chuan H Foh. Module: Internet of things lecture 3: Hardware platform.

[67] Tarek Farah and Safya Belghith. A new chaotic encryption algorithm for wsn and implementation with sensors as-xm1000. In *2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pages 684–689. IEEE, 2017.

[68] Luca Catarinucci, Danilo De Donno, Luca Mainetti, Luca Palano, Luigi Patrono, Maria Laura Stefanizzi, and Luciano Tarricone. An iot-aware architecture for smart healthcare systems. *IEEE Internet of Things Journal*, 2(6):515–526, 2015.

[69] Edgar M Silva, Pedro Maló, and Michele Albano. Energy consumption awareness for resource-constrained devices. In *2016 European Conference on Networks and Communications (EuCNC)*, pages 74–78. IEEE, 2016.

[70] Oliver Hahm, Emmanuel Baccelli, Hauke Petersen, and Nicolas Tsiftes. Operating systems for low-end devices in the internet of things: a survey. *IEEE Internet of Things Journal*, 3(5):720–734, 2015.

[71] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *29th annual IEEE international conference on local computer networks*, pages 455–462. IEEE, 2004.

[72] Adam Dunkels, Oliver Schmidt, Niclas Finne, Joakim Eriksson, Fredrik Österlind, and Nicolas Tsiftesand Mathilde Durvy. The contiki os: The operating system for the internet of things. *Online], at http://www. contikios. org*, 605, 2011.

[73] Muhammad Omer Farooq and Thomas Kunz. Operating systems for wireless sensor networks: A survey. *Sensors*, 11(6):5900–5930, 2011.

[74] Nicolas Tsiftes, Adam Dunkels, Zhitao He, and Thiemo Voigt. Enabling large-scale storage in sensor networks with the coffee file system. In *2009 International Conference on Information Processing in Sensor Networks*, pages 349–360. IEEE, 2009.

[75] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, et al. Tinyos: An operating system for sensor networks. In *Ambient intelligence*, pages 115–148. Springer, 2005.

[76] Shah Bhatti, James Carlson, Hui Dai, Jing Deng, Jeff Rose, Anmol Sheth, Brian Shucker, Charles Gruenwald, Adam Torgerson, and Richard Han. Mantis os: An embedded multithreaded operating system for wireless micro sensor platforms. *Mobile Networks and Applications*, 10(4):563–579, 2005.

[77] Anand Eswaran, Anthony Rowe, and Raj Rajkumar. Nano-rk: an energy-aware resource-centric rtos for sensor networks. In *26th IEEE International Real-Time Systems Symposium (RTSS'05)*, pages 10–pp. IEEE, 2005.

[78] Anand Nayyar and Rajeshwar Singh. A comprehensive review of simulation tools for wireless sensor networks (wsns). *Journal of Wireless Networking and Communications*, 5(1):19–47, 2015.

[79] Teerawat Issariyakul and Ekram Hossain. Introduction to network simulator 2 (ns2). In *Introduction to network simulator NS2*, pages 1–18. Springer, 2009.

[80] Kevin Fall, Kannan Varadhan, et al. The ns manual (formerly ns notes and documentation). *The VINT project*, 47:19–231, 2005.

[81] Mohammed Humayun Kabir, Syful Islam, Md Javed Hossain, and Sazzad Hossain. Detail comparison of network simulators. *International Journal of Scientific & Engineering Research*, 5(10):203–218, 2014.

[82] OMNeT++. Omnet++, discrete event simulator. `https://omnetpp.org/`.

[83] Gitanjali Pradhan, Rajni Gupta, and Suparna Biswasz. Study and simulation of wban mac protocols for emergency data traffic in healthcare. In *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, pages 1–4. IEEE, 2018.

[84] Khoa Anh Ngo, Trong Thua Huynh, and De Thu Huynh. Simulation wireless sensor networks in castalia. In *Proceedings of the 2018 International Conference on Intelligent Information Technology*, pages 39–44. ACM, 2018.

[85] M Bennani Mohamed Taj and M Ait Kbir. The impact of mac protocols in energy consumption of transferring multimedia contents using castalia simulator. In *2016 International Conference on Electrical and Information Technologies (ICEIT)*, pages 521–525. IEEE, 2016.

[86] Gerard Kevin B Cagatan, Jo-Ann V Magsumbol, Renann Baldovino, Edwin Sybingco, and Elmer P Dadios. Connectivity analysis of wireless sensor network in two-dimensional plane using castalia simulator. In *2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, pages 1–8. IEEE, 2017.

[87] K Kalaiselvi and A Angel Cerli. A pragmatic implementation of energy efficiency in the wireless body area network using castalia. In *2017 International Conference on Inventive Computing and Informatics (ICICI)*, pages 6–8. IEEE, 2017.

[88] Abhilash Hegde and Durga Prasad. Evaluation of phy and mac layer in wban using castalia: Lower layer parameters of wban in castalia. In *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, pages 395–399. IEEE, 2017.

[89] Ahmad AA Alkhatib. A review on forest fire detection techniques. *International Journal of Distributed Sensor Networks*, 10(3):597368, 2014.

[90] FIRE LOOKOUT TOWERS. B.c. fire lookout towers. `http://www.firelookout.com/bc.html`, 2018.

[91] Eric Den Breejen, Marcel Breuers, Frank Cremer, Rob Kemp, Marco Roos, Klamer Schutte, and Jan S De Vries. *Autonomous forest fire detection*. Coimbra, Portugal: ADAI-Associacao para o Desenvolvimento da Aerodinamica ..., 1998.

[92] NOAA satellite and information service. Advanced very high resolution radiometer—avhrr. `http://noaasis.noaa.gov/NOAASIS/ml/avhrr.html.`

[93] National Aeronautics MOIDS and space administration. National aeronautics and space administration. `http://modis.gsfc.nasa.gov/.`

[94] Zhanqing Li, Serge Nadon, and Josef Cihlar. Satellite-based detection of canadian boreal forest fires: Development and application of the algorithm. *International Journal of Remote Sensing*, 21(16):3057–3069, 2000.

[95] Mohamed Hefeeda and Majid Bagheri. Forest fire modeling and early detection using wireless sensor networks. *Ad Hoc & Sensor Wireless Networks*, 7(3-4):169–224, 2009.

[96] Yunus Emre Aslan, Ibrahim Korpeoglu, and Özgür Ulusoy. A framework for use of wireless sensor networks in forest fire detection and monitoring. *Computers, Environment and Urban Systems*, 36(6):614–625, 2012.

[97] Anand S Bhosle and Laxmikant M Gavhane. Forest disaster management with wireless sensor network. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 287–289. IEEE, 2016.

[98] Sunny Samanta. An architecture of future forest fire detection system. *International Journal on Recent and Innovation Trends in Computing and Communication*, 4(8):139–140, 2016.

[99] Antonio Molina-Pico, David Cuesta-Frau, Alvaro Araujo, Javier Alejandre, and Alba Rozas. Forest monitoring and wildland early fire detection by a hierarchical wireless sensor network. *Journal of Sensors*, 2016, 2016.

[100] Yi-Han Xu, Qiu-Ya Sun, and Yu-Tong Xiao. An environmentally aware scheme of wireless sensor networks for forest fire monitoring and detection. *Future Internet*, 10(10):102, 2018.

[101] M Trinath Basu, Ragipati Karthik, J Mahitha, and V Lokesh Reddy. Iot based forest fire detection system. *Int. J. Eng. Technol*, 7:124–126, 2018.

[102] Harjinder Singh. Forest fire detection using wireless sensor. *Int. J. Sci. Eng. Res*, 7:554–562, 2016.

[103] Benamar Kadri, Benamar Bouyeddou, and Djillali Moussaoui. Early fire detection system using wireless sensor networks. In *2018 International Conference on Applied Smart Systems (ICASS)*, pages 1–4. IEEE, 2018.

[104] Jaime Lloret, Miguel Garcia, Diana Bri, and Sandra Sendra. A wireless sensor network deployment for rural and forest fire detection and verification. *sensors*, 9(11):8722–8747, 2009.

[105] Chi Yuan, Zhixiang Liu, and Youmin Zhang. Fire detection using infrared images for uav-based forest fire surveillance. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 567–572. IEEE, 2017.

[106] Sheikh Ferdoush and Xinrong Li. Wireless sensor network system design using raspberry pi and arduino for environmental monitoring applications. *Procedia Computer Science*, 34:103–110, 2014.

[107] Lalatendu Muduli, Devi Prasad Mishra, and Prasanta K Jana. Application of wireless sensor network for environmental monitoring in underground coal mines: a systematic review. *Journal of Network and Computer Applications*, 106:48–67, 2018.

[108] Vittorio P Illiano, Andrea Paudice, Luis Muñoz-González, and Emil C Lupu. Determining resilience gains from anomaly detection for event integrity in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 14(1):5, 2018.

[109] R Kumar and M Pallikonda Rajasekaran. An iot based patient monitoring system using raspberry pi. In *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, pages 1–4. IEEE, 2016.

[110] Wu-Chi Feng, Ed Kaiser, Wu Chang Feng, and Mikael Le Baillif. Panoptes: scalable low-power video sensor networking technologies. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 1(2):151–167, 2005.

[111] Sanjana Prasad, P Mahalakshmi, A John Clement Sunder, and R Swathi. Smart surveillance monitoring system using raspberry pi and pir sensor. *Int. J. Comput. Sci. Inf. Technol*, 5(6):7107–7109, 2014.

[112] Purushottam Kulkarni, Deepak Ganesan, Prashant Shenoy, and Qifeng Lu. Senseye: a multi-tier camera sensor network. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 229–238. ACM, 2005.

[113] Purushottam Kulkarni, Deepak Ganesan, and Prashant Shenoy. The case for multi–tier camera sensor networks. In *Proceedings of the international workshop on Network and operating systems support for digital audio and video*, pages 141–146. ACM, 2005.

[114] Hakan Oztarak, Turgay Yilmaz, Kemal Akkaya, and Adnan Yazici. Efficient and accurate object classification in wireless multimedia sensor networks. In *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, pages 1–7. IEEE, 2012.

[115] Mian Ahmad Jan, Priyadarsi Nanda, Xiangjian He, and Ren Ping Liu. A sybil attack detection scheme for a forest wildfire monitoring application. *Future Generation Computer Systems*, 80:613–626, 2018.

[116] M Civelek and A Yazici. Automated moving object classification in wireless multimedia sensor networks. ieee sensors j 17: 1116–1131, 2016.

[117] Murat Koyuncu, Adnan Yazici, Muhsin Civelek, Ahmet Cosar, and Mustafa Sert. Visual and auditory data fusion for energy-efficient and improved object recognition in wireless multimedia sensor networks. *IEEE Sensors Journal*, 19(5):1839–1849, 2018.

[118] Athanasia Panousopoulou, Mikel Azkune, and Panagiotis Tsakalides. Feature selection for performance characterization in multi-hop wireless sensor networks. *Ad Hoc Networks*, 49:70–89, 2016.

[119] V Vipin. Image processing based forest fire detection. *International Journal of Emerging Technology and Advanced Engineering*, 2(2):87–95, 2012.

[120] Jiye Qian, Jin Fu, Jide Qian, Weibin Yang, Ke Wang, and Pan Cao. Automatic early forest fire detection based on gaussian mixture model. In *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, pages 1192–1196. IEEE, 2018.

[121] Thou-Ho Chen, Ping-Hsueh Wu, and Yung-Chuen Chiou. An early fire-detection method based on image processing. In *2004 International Conference on Image Processing, 2004. ICIP'04.*, volume 3, pages 1707–1710. IEEE, 2004.

[122] Zhentian Jiao, Youmin Zhang, Jing Xin, Yingmin Yi, Ding Liu, and Han Liu. Forest fire detection with color features and wavelet analysis based on aerial imagery. In *2018 Chinese Automation Congress (CAC)*, pages 2206–2211. IEEE, 2018.

[123] Jun-zhe Zhou and Ge Song. Research on the technology of fire detection based on image processing in unmanned substation. In *2010 Third International Conference on Intelligent Networks and Intelligent Systems*, pages 108–111. IEEE, 2010.

[124] Nargess Ghassempour, Ju Jia Zou, and Yaping He. A sift-based forest fire detection framework using static images. In *2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pages 1–7. IEEE, 2018.

[125] Surapong Surit and Watchara Chatwiriya. Forest fire smoke detection in video based on digital image processing approach with static and dynamic characteristic analysis. In *2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering*, pages 35–39. IEEE, 2011.

[126] Li Jie and Xiao Jiang. Forest fire detection based on video multi-feature fusion. In *2009 2nd IEEE International Conference on Computer Science and Information Technology*, pages 19–22. IEEE, 2009.

[127] Wang Yuanbin and Ma Xianmin. Early fire detection for high space based on video-image processing. In *2014 International Symposium on Computer, Consumer and Control*, pages 785–788. IEEE, 2014.

[128] C Emmy Premal and SS Vinsley. Image processing based forest fire detection using ycbcr colour model. In *2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014]*, pages 1229–1237. IEEE, 2014.

[129] Dengyi Zhang, Shizhong Han, Jianhui Zhao, Zhong Zhang, Chengzhang Qu, Youwang Ke, and Xiang Chen. Image based forest fire detection using dynamic characteristics with artificial neural networks. In *2009 International Joint Conference on Artificial Intelligence*, pages 290–293. IEEE, 2009.

[130] JiSeong Han, GwangSu Kim, ChanSeo Lee, YeongKwang Han, Ung Hwang, and SungHwan Kim. Predictive models of fire via deep learning exploiting colorific variation. In *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 579–581. IEEE, 2019.

[131] Khan Muhammad, Jamil Ahmad, and Sung Wook Baik. Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing*, 288:30–42, 2018.

[132] Wonjae Lee, Seonghyun Kim, Yong-Tae Lee, Hyun-Woo Lee, and Min Choi. Deep neural networks for wild fire detection with unmanned aerial vehicle. In *2017 IEEE international conference on consumer electronics (ICCE)*, pages 252–253. IEEE, 2017.

[133] Yunxia Chen and Qing Zhao. On the lifetime of wireless sensor networks. *IEEE Communications letters*, 9(11):976–978, 2005.

[134] Enrique J Duarte-Melo and Mingyan Liu. Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks. In *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE*, volume 1, pages 21–25. IEEE, 2002.

[135] Rahul C Shah, Sumit Roy, Sushant Jain, and Waylon Brunette. Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks*, 1(2-3):215–233, 2003.

[136] Deokwoo Jung, Thiago Teixeira, and Andreas Savvides. Sensor node lifetime analysis: Models and tools. *ACM Transactions on Sensor Networks (TOSN)*, 5(1):3, 2009.

[137] Roberto Di Pietro, Luigi V Mancini, Claudio Soriente, Angelo Spognardi, and Gene Tsudik. Catch me (if you can): Data survival in unattended sensor networks. In *2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 185–194. IEEE, 2008.

[138] Santosh Kumar, Anish Arora, and Ten-Hwang Lai. On the lifetime analysis of always-on wireless sensor network applications. In *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005.*, pages 3–pp. IEEE, 2005.

[139] Adriano B da Cunha and Diógenes C da Silva. An approach for the reduction of power consumption in sensor nodes of wireless sensor networks: Case analysis of mica2. In *International Workshop on Embedded Computer Systems*, pages 132–141. Springer, 2006.

[140] Joseph Polastre, Robert Szewczyk, Alan Mainwaring, David Culler, and John Anderson. Analysis of wireless sensor networks for habitat monitoring. In *Wireless sensor networks*, pages 399–423. Springer, 2004.

[141] Hoang Anh Nguyen, Anna Förster, Daniele Puccinelli, and Silvia Giordano. Sensor node lifetime: An experimental study. In *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 202–207. IEEE, 2011.

[142] Wilfried Dron, Simon Duquennoy, Thiemo Voigt, Khalil Hachicha, and Patrick Garda. An emulation-based method for lifetime estimation of wireless sensor networks. In *2014 IEEE International Conference on Distributed Computing in Sensor Systems*, pages 241–248. IEEE, 2014.

[143] Zhanshan Ma and Axel W Krings. Insect population inspired wireless sensor networks: A unified architecture with survival analysis, evolutionary game theory, and hybrid fault models. In *2008 International Conference on BioMedical Engineering and Informatics*, volume 2, pages 636–643. IEEE, 2008.

[144] Samuel Madden et al. Intel lab data. *Web page, Intel*, 2004.

[145] MICAz Datasheet. Crossbow technology inc. *San Jose, California*, 50, 2006.

[146] Pi, raspberry. "model b.". `https://www.raspberrypi.org/products/raspberry-pi-3-model-b/`, 2018.

[147] Alireza Khadivi and Martin Hasler. Fire detection and localization using wireless sensor networks. In *International Conference on Sensor Applications, Experimentation and Logistics*, pages 16–26. Springer, 2009.

[148] Evangelia Kolega, Vassilios Vescoukis, and Christos Douligeris. Forest fire sensing and decision support using large scale wsns. 2010.

[149] Khalida Nisimova. Energy of a 1.5 v battery, the physics factbook. `https://hypertextbook.com/facts/2001/KhalidaNisimova.shtml`, 2001.

[150] Adam Dunkels, Joakim Eriksson, Niclas Finne, and Nicolas Tsiftes. Powertrace: Network-level power profiling for low-power wireless networks, 2011.

[151] Power supply. raspberry-pi-power. `https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/`, 2018.

[152] Athanassios. Boulis. Castalia user manual. http://castalia. npc. nicta. com. au/pdfs/Castalia-User Manual, 2009.

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

[154] Wei Di, Anurag Bhardwaj, and Jianing Wei. *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. Packt Publishing, 2018.

[155] Yi-Tong Zhou and Rama Chellappa. Computation of optical flow using a neural network. In *IEEE International Conference on Neural Networks*, volume 1998, pages 71–78, 1988.

[156] François Chollet et al. Keras, 2015.

[157] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5353–5360, 2015.

[158] Jiawei Liu, Zheng-Jun Zha, QI Tian, Dong Liu, Ting Yao, Qiang Ling, and Tao Mei. Multi-scale triplet cnn for person re-identification. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 192–196. ACM, 2016.

[159] Linnan Wang, Yi Yang, Renqiang Min, and Srimat Chakradhar. Accelerating deep neural network training with inconsistent stochastic gradient descent. *Neural Networks*, 93:219–229, 2017.

[160] Lingxi Xie, Jingdong Wang, Zhen Wei, Meng Wang, and Qi Tian. Disturblabel: Regularizing cnn on the loss layer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4753–4762, 2016.

[161] Keras applications and available models. https://keras.io/applications/.

[162] Qingjie Zhang, Jiaolong Xu, Liang Xu, and Haifeng Guo. Deep convolutional neural networks for forest fire detection. In *2016 International Forum on Management, Education and Information Technology Application*. Atlantis Press, 2016.

[163] Henry Cruz, Martina Eckert, Juan Meneses, and José-Fernán Martínez. Efficient forest fire detection index for application in unmanned aerial systems (uass). *Sensors*, 16(6):893, 2016.

[164] C Emmy Prema, SS Vinsley, and S Suresh. Multi feature analysis of smoke in yuv color space for early forest fire detection. *Fire technology*, 52(5):1319–1342, 2016.