12-2022

IoT in smart communities, technologies and applications.

Muhammad Zaigham Abbas Shah Syed
University of Louisville

IOT IN SMART COMMUNITIES, TECHNOLOGIES AND APPLICATIONS

By
Muhammad Zaigham Abbas Shah Syed
B.Eng., M.Sc.

A Dissertation
Submitted to the Faculty of the
J.B. Speed School of Engineering of the University of
Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy
in Computer Science and Engineering

Department of Computer Science and Engineering
University of Louisville
Louisville, Kentucky

December 2022

IOT IN SMART COMMUNITIES, TECHNOLOGIES AND APPLICATIONS

By

Muhammad Zaigham Abbas Shah Syed
B.Eng., M.Sc.

Dissertation approved on

November 8, 2022

by the following dissertation Committee:

Dissertation Director
Dr. Adel S. Elmaghraby

Dr. Daniel Sierrasosa

Dr. Anup Kumar

Dr. Ibrahim Imam

Dr. Monica Gentili

DEDICATION

To my late grandmother, Shireen Shah who always championed the importance of education and personal values. She has had the biggest impact on the person that I have become.

To my parents, who worked hard to provide us with the best opportunities that can be provided to any child.

ACKNOWLEDGMENTS

I would like to thank my supervisor, Dr. Adel Elmaghraby for his support and guidance throughout the course of my research work here at the University of Louisville. Without his support and trust I would not have been able to succeed in this endeavor.

I would also like to thank Dr. Anup Kumar and Dr. Daniel Sierra-Sosa for their support in conducting my research and providing feedback on our experiments which helped improve the quality of the work.

I would like to thank all my friends, both in the United States and elsewhere who have been a tremendous support to me during my time in Louisville. They were like a family away from home.

Finally, I would like to say thanks to my family, who have supported me throughout think and thin. First and foremost my parents, my mother Naheed and father Sher Muhammad. They through their hard work and determination left no stone unturned to provide us with the best upbringing anyone can provide to their children. My wife Farwa, who has always been a pillar of support for me. My siblings, Zafi, Shehram and Fiza who have been the best brothers and sister anyone can ask for. My sisters in law, Madiha and Hina and my lovely niece Maryam, who is a cause of much joy in our life.

ABSTRACT

IOT IN SMART COMMUNITIES, TECHNOLOGIES AND APPLICATIONS

Muhammad Zaigham Abbas Shah, Syed

November 8, 2022

Internet of Things is a system that integrates different devices and technologies, removing the necessity of human intervention. This enables the capacity of having smart (or smarter) cities around the world. By hosting different technologies and allowing interactions between them, the internet of things has spearheaded the development of smart city systems for sustainable living, increased comfort and productivity for citizens. The Internet of Things (IoT) for Smart Cities has many different domains and draws upon various underlying systems for its operation, in this work, we provide a holistic coverage of the Internet of Things in Smart Cities by discussing the fundamental components that make up the IoT Smart City landscape, the technologies that enable these domains to exist, the most prevalent practices and techniques which are used in these domains as well as the challenges that deployment of IoT systems for smart cities encounter and which need to be addressed for ubiquitous use of smart city applications. It also presents a coverage of optimization methods and applications from a smart city perspective enabled by the Internet of Things. Towards this end, a mapping is provided for the most encountered applications of computational optimization within IoT smart cities for five popular optimization methods, ant colony optimization, genetic algorithm, particle swarm optimization, artificial bee colony optimization and differential evolution. For each application identified, the algorithms used, objectives considered, the nature of the formulation and constraints taken in

to account have been specified and discussed. Lastly, the data setup used by each covered work is also mentioned and directions for future work have been identified.

Within the smart health domain of IoT smart cities, human activity recognition has been a key study topic in the development of cyber physical systems and assisted living applications. In particular, inertial sensor based systems have become increasingly popular because they do not restrict users' movement and are also relatively simple to implement compared to other approaches. Fall detection is one of the most important tasks in human activity recognition. With an increasingly aging world population and an inclination by the elderly to live alone, the need to incorporate dependable fall detection schemes in smart devices such as phones, watches has gained momentum. Therefore, differentiating between falls and activities of daily living (ADLs) has been the focus of researchers in recent years with very good results. However, one aspect within fall detection that has not been investigated much is direction and severity aware fall detection. Since a fall detection system aims to detect falls in people and notify medical personnel, it could be of added value to health professionals tending to a patient suffering from a fall to know the nature of the accident. In this regard, as a case study for smart health, four different experiments have been conducted for the task of fall detection with direction and severity consideration on two publicly available datasets. These four experiments not only tackle the problem on an increasingly complicated level (the first one considers a fall only scenario and the other two a combined activity of daily living and fall scenario) but also present methodologies which outperform the state of the art techniques as discussed. Lastly, future recommendation have also been provided for researchers.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION AND DISSERTATION OVERVIEW

1 Motivation

The world is experiencing a bulging world population and increasing urbanization which is set to grow by more than 10% in the next 30 years [1] resulting in a total of 70% living in cities by 2050. Countries around the world are looking at equipping their cities to deal with the influx of people and the stress it will bring to current city systems [2]. This is to be performed keeping in view the UN Sustainable Development Goals 2030 [3]. In this regard, Smart Cities have come out as a major initiative by various governments in making cities more navigable and welcoming to the expected population increase and providing city dwellers a better living experience, as is evidenced by the multiple projects ongoing on both the public and private level [4, 5, 6, 7]. The Internet of Things (IoT) has been the prime enabler of smart cities as it has provided the means in terms of sensors, architectures, storage and other technologies to instrument cities. It allows stakeholders to gather data, process it and then make decisions based on the analyses performed. There are various domains within smart cities, these include Smart Agriculture, Smart City Services, Smart Grid, Smart Health, Smart Homes, Smart Industry, Smart Infrastructure and Smart Transportation.With improvement of Artificial intelligence (AI) capabilities in the last decade, there have been different applications of machine learning (ML) and deep learning (DL) algorithms within each of these domains providing for better decision making and improvement of services. Apart from various supervised and unsupervised learning applications, several tasks within different components of smart cities can be formulated into optimization problems and/or require heuristics to be incorporated in some form. However, in both such tasks, there is a dearth of an overall coverage of Smart City IoT based ML/DL and combinatorial optimization problems. Such a coverage would be very beneficial to researchers starting in the field and has been presented herein.

One of the major applications of the IoT Smart Cities is for health purposes. Many countries around the world, especially in the developed world are facing an aging population. With the retirement of the 'baby boomer' generation, countries around the world are facing a big question over caring for their elderly population. One of the major issues that elderly people face today are falls [8]. In the US, falls account for a majority of the deaths caused by injury in population aged 65 and above [9] and a fall related injury occurs every 19 minutes [10]. As indicated in [11], falls have been associated with estimated medical costs ranging from \$31.5 billion for 2013 to \$49.5 billion for 2017. Moreover, Floerence et al. [12] notes that in 2018, falls contributed to 8% of Medicaid expenses in the US for older adults, consequently falls have huge health as well as financial ramifications for people and governments. Determining when a person has experienced a fall is therefore an important task in the healthcare of people, more so in the old age population who tend to increasingly live alone and are at increased risk of suffering from falls. Apart from just detecting if a fall has occurred or not, the direction of a fall is an important consideration as well. Bourke et al. [13] notes that even though 60% of falls occurring in older adults are forward falls, falls occurring sideways may result in fracture and thus are important to determine. Quick dispatch of healthcare providers can result in timely treatment of fall related injuries, thereby reducing the overall impact to life and wealth. Additional information about the fall could benefit the medical dispatch team to help decide emergency procedures which could be matched up with patient history of potentially being injury prone in a certain part of the body. In light of this all, the development of fall detection systems has been of keen interest to researchers in the domain of human activity recognition, tele-medicine and smart health.

2 Aims

Citing the importance of IoT usage in communities, the aims of this dissertation are as follows:

1. To provide a wholistic coverage of IoT based Smart City system design considering important aspects such as sensors, technologies used, IoT architectures, the use of machine/deep learning methods and optimization schemes as well as the challenges.

2. Considering the case study of smart health and specifically fall detection, develop methodologies for detection of falls considering fall direction and severity.

3 Organization

This dissertation is organized in to six chapters. Chapter II presents a coverage of the IoT for Smart Cities in terms of the technologies used, its architectures and also the challenges towards IoT usage in Smart Cities. Chapter III discusses the the applications of machine and deep learning algorithms. This chapter provides a comprehensive discussion on IoT usage in Smart Cities by considering the different types of systems devised for each application as well. Chapter IV discusses the applications of optimization algorithms in IoT based smart cities for five popular algorithms. It thus provides a coverage of optimization applications under the IoT smart city umbrella. Chapter V presents work on a chosen case study of smart health based on IoT. The considered task is fall detection with direction and severity consideration. Four experiments have been performed in this regard considering three scenarios, one considering falls only with direction and severity and two others considering falls and activity of daily living (ADL) as a combined problem. The designed methodologies have shown to outperform the state of the art as has been discussed in the chapter. The conclusion and future work opportunities are discussed in chapter VI.

CHAPTER II

IOT SMART CITIES

1 Introduction

This chapter provides an overeview of the use of IoT in Smart Cities and discuss how it facilitates such initiatives. It starts by laying out the structure of Internet of Things in a Smart City context, discussing its various applications, components and architectures while also discussing the technologies used at the different levels of the IoT architecture. Lastly, a discussion of the technical challenges that exist in the deployment of IoT in the Smart City domain is provided along with identification of potential solutions to those challenges followed by future directions.

It is hard to define a Smart City, in fact, cities claim to be ‘smart’ based on a variety of criteria, for e.g. implementing novel e-governance schemes, creating social learning ventures and community engagement programs, focusing on sustainable living as well as the more typical application of Information and Communication Technologies for innovation [14]. In this work, Smart Cities are defined to be the application of various information and communication technologies (ICT) with the aim of creating a better living experience for a city’s population. This encompasses use of these technologies in all the domains discussed previously, including governance, transport, housing, business, sustainable living, social learning, community engagement, providing opportunities and more. In an ideal sense, the idea of a smart city transcends the typically set boundaries of a traditional city’s administrative and social structure by allowing interaction between the two, thereby enabling it to operate in a more cohesive and engaged manner. Smart cities offer several advantages (in terms of value) compared to a traditional city ecosystem:

1. *Climate goal achievement:* Smart cities are at the forefront of pioneering technologies to help enable countries meet climate goals. Smart city focuses on smart energy management, smart transportation systems and city administration which aim to reduce the carbon footprint of cities and enable development and use of new technologies for cleaner living.

2. *Money value:* Smart City ventures will be a market of USD 1 Trillion by 2025 [15], this provides a huge monetary incentive for not only governments but private companies to actively contribute to the development of technologies supporting smart city development.

3. *Societal impact:* The centerpiece of the smart city project is to improve the quality of life of a city’s inhabitants and help develop an inclusive society wherein every opinion is catered for and equal opportunity is provided. Information and

Communication Technologies in the smart city context are a fundamental component to the provision of public services by facilitating interactions of citizens with the city environment and making life easier.

2 Smart City Components

A smart city is made up of several components which are illustrated in Figure 1. Smart city applications typically have four aspects associated with them, the first is the collection of data, the next is its transmission/reception, third is the storage and fourth is analysis. The collection of data is application dependent and has been a real driver for sensor development in the various domains. The second part is the exchange of data, this involves data transmission from the data collection units towards the cloud for storage and analysis. This task has been achieved in several manners, many smart city ventures have city-wide Wi-Fi networks, 4G and 5G technologies are being used, as well as various types of local networks which can convey data either on a local level or a global level. The third stage is storage in the cloud, different storage schemes are used to arrange and organize data so as to make it usable for the fourth stage which is data analysis. Data Analysis refers to the extraction of patterns and inferences from the gathered data to guide decision making. In some situations, simple analysis such as basic decision making and aggregating would work too. For more complex decision making, the availability of the cloud allows not only for heterogeneous data gathering/storage and processing but also analysis through the use of statistical methods as well as machine and deep Learning algorithms in real-time [16].

2.1 Smart Agriculture

Food security is one of the most important parts of the United Nations Sustainable Development goals for 2030. With an increasing world population, worsening climate change causing erratic weather in food centers of the world, the race to ensure that food production is made sustainable and that dwindling resources such as water are utilized efficiently has been a high priority for countries around the world. Smart agriculture is the use of sensors embedded into plants and fields to measure various parameters to help in decision making and prevent/diseases, pests etc [17]. A part of the smart agriculture paradigm is precision agriculture, which involves sensors being placed in plants to provide targeted measurements and therefore allow for targeted care mechanisms to be deployed. Precision agriculture will be necessary for food security in the future [18] and therefore is an essential part of the fight for sustainable food production. The major applications of AI in IoT for agriculture are crop monitoring/disease detection and data driven crop care and decision making.

2.2 Smart City Services

Smart city services encompass the activities that sustain a city's population, these involve municipal tasks such as supply of water, waste management, environmental

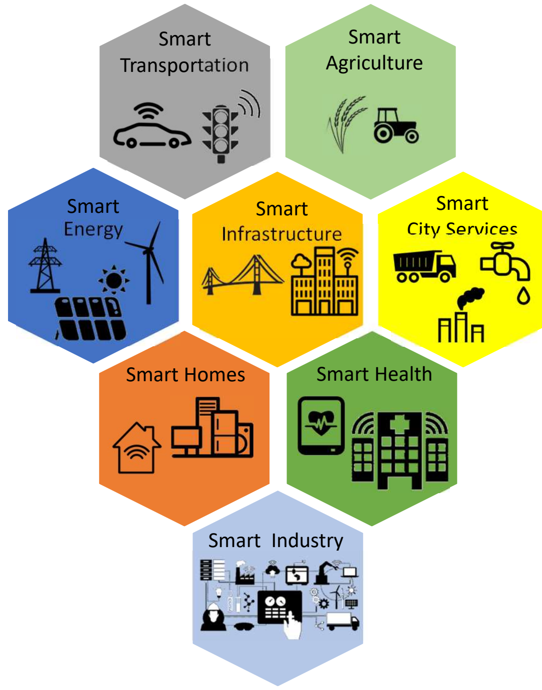

Figure 1. Smart City Components

control and monitoring etc. Sensors for water quality can be deployed to continually provide an update about the quality of water being used in the city and detect leaks [19]. One popular component of smart city initiatives is the management of waste, and it has been the part of many of the smart city initiatives mentioned earlier, from chutes in Barcelona to having bins equipped with sensors and connected to the cloud so as to not only inform the relevant authorities of the need to empty them but also using AI to determine the best route to reduce cost [20]. Sensors can also be used to monitor the environmental conditions in a city to determine pollution levels [21] and guiding citizens to the next free parking space to save fuel costs [22].

2.3 Smart Energy

Typical electrical systems have one-way energy flow from a main generator source, usually a hydroelectric or fossil fuel based power plant. Power generation is controlled via feedback from the substations, however, since there is no information feedback from the consumer end, the power generation scheme used with these systems requires that the power being produced by these sources outstrips the demand by a large amount to ensure continuous supply of power. The process of detecting faults and performing corrective actions in such systems is also a time taking process. Moreover, with renewable energy technologies becoming cheaper, the consumer of today not only has a supply from the main utility but also performs their own generation. Smart Grids is the use of ICT to make the current and newly installed grids more observable, allow for distributed energy generation, both at the consumer end as well as the utility end and introduce self-healing capabilities in to the grid. One aspect of smart grids is that real time power data is transmitted to utilities at different points on the grid throughout the supply lines till the customer. Since smart grids provide real-time data about consumer usage, it allows for better management of power generation using prediction models developed through acquired consumption data, integrating different energy sources as well as self-healing [23] of the network to ensure an uninterrupted supply.

2.4 Smart Health

Smart Health refers to the use of ICT to improve health care availability and quality. With an increasing population and rising costs of healthcare, this area has been of intense focus of researchers as well as healthcare providers. Current health systems are over burdened and therefore cannot cater to the increasing demand from the populace. In this regard, smart health aims to ensure that healthcare be available to as many people as possible through telemedicine services [24] and improved diagnosis assistance to doctors utilizing AI [25]. With the ubiquitousness of mobile phones and health trackers [26] that can capture real-time data about peoples health (ECGs, temperature, body oxygen saturation and other biosensors) while also recording daily activity and detecting abnormal movements using inertial sensors, it has become possible to leverage cloud capabilities for processing this data to make better healthcare decisions. Thus reducing the overall costs as well as burden on healthcare facilities.

2.5 Smart Home

One major component of Smart Cities is the Smart Home since it is central to the life of the city's inhabitants. Smart Homes involve the use of sensing units installed throughout a person's home that provide information about the home as well as its occupants. These sensors might include user activity monitors such as ambient sensors, motion trackers and power/energy consumption.

2.6 Smart Industry

Industries around the world are busy in a continuous pursuance of being more efficient and increasing productivity while reducing cost. The industry 4.0 paradigm entails the vision of a connected factory where all its intermediary functionaries are seamlessly integrated, working in tandem with each other. This is made possible because of the internet of things [27]. The use of IoT in manufacturing and production processes, cyber physical systems integrating workers and machines has led to several benefits to the industry such as faster and better innovation, optimization of manufacturing schemes (resources and processes), better quality of products and enhanced safety for factory workers. However, smart industries come with several challenges for IoT usage, working with a set of heterogeneous devices and machines has its own challenges and requires cyber physical systems to have flexibility in configuration, connectivity and quick implementation for use in IoT applications for Smart Industry [28]. Artificial Intelligence has gone hand in hand with IoT to spur the development and deployment of industry 4.0 services. With sensors being embedded in machines and other processes in the factory, data from these sources provide an opportunity for using AI techniques to increase automation, perform business intelligence operations and more. In fact, researchers have suggested frameworks for integrating AI within IoT for Smart Industry [28, 29, 30]. The major applications of AI in the industry are predictive maintenance, monitoring/fault detection (machine health) and production management.

2.7 Smart Infrastructure

The infrastructure of a city is paramount to its living quality, city governments need to construct new bridges, roads and buildings for the use of its inhabitants and also perform maintenance for uninterrupted usage. Smart infrastructure helps cities in ensuring their infrastructure is in shape and usable by utilizing sensors for measuring building/bridge structural state for structural health monitoring using accelerometers [31] and smart materials [32]. Data collected through these sensors allows for predictive maintenance of these essential units to maintain normal operation of the city.

2.8 Smart Transport

Many urban centers suffer from traffic problems, this includes congestion, pollution, scheduling and cost reduction issues for public transport. The rapid development and implementation of new Information and Communication Technologies, Vehicle-Infrastructure-Pedestrian communication has become commonplace. Whether it be Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I), Vehicle to Pedestrian (V2P) or Pedestrian to Infrastructure (P2I), such technologies have made the design of smart transportation systems possible. With cars having a GPS device and the commonality of cellphones with every driver, many approaches use GPS data to track driver behavior and traffic patterns [33]. This real-time data is already used for route mapping in applications such as Waze and Google Maps and used for trip scheduling in public transport. Parking systems equipped with sensors can also guide drivers to the nearest free parking spot.

3 Internet of Things for Smart Cities

At the heart of the smart city initiatives is the internet of things, it is the enabling technology that has allowed for the pervasive digitization that gives rise to the concept of smart cities. The internet of things refers to the ubiquitous connection of devices to the internet, allowing them to send information to the cloud and potentially get directions for performing actions. IoT involves the collection of data and performing data analytics operations to extract information in order to support decision and policy making. It is estimated that by 2025 more than 75 billion devices will be connected to the internet [34], spearheading even more application development. Within the smart city context, IoT allows for sensors to collect and send data about the city's state to a central cloud, which is then mined or processed for pattern extraction and decision-making purposes.

3.1 IoT Architectures for Smart cities

The Internet of Things unifies the operations of data sensing, transmission/reception, processing and storage through the use cloud services. Based on technology, a generic IoT architecture consists of five layers where successive layers operate on the information from the previous layer as shown in Figure 2. It also shows the three different architectures that exist for IoT systems.

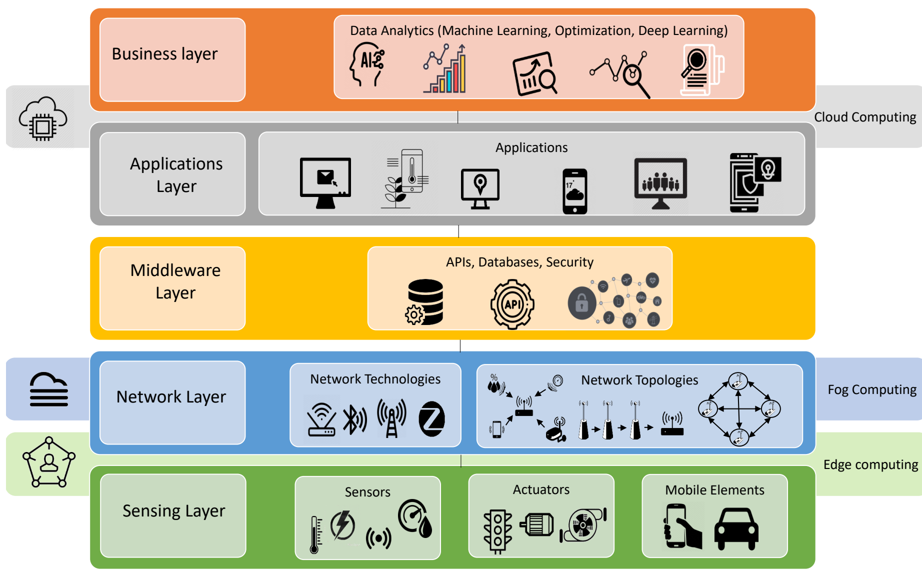

Figure 2. IoT Architecture

The Sensing layer, also called the Perception layer consists of sensors that can get information about physical quantities of interest in any application as well as actuators which can act upon physical objects, such as Radio Frequency IDentification (RFID) readers for reading RFID tags and other such devices. The data read by the sensing layer is passed onward to the Middleware layer using the networking layer through wireless network technologies such as Wi-Fi, cellular internet, Zigbee and Bluetooth etc. The Middleware layer provides a generic interface for the sensing layer hardware and the Application layer which uses the data through various API's and database management services to provide users with services. The Business layer is attached to the application layer and is used to develop strategies and formulate policies that help manage the complete system.

In terms of architectures, typically, IoT architectures are categorized based on the type of operation responsibilities allocated to parts of the IoT system, this categorization is based mainly on processing of data responsibilities. There are three architectures of IoT systems with respect to the stage of the IoT framework where processing of data can be carried out and these are Cloud, Fog and Edge Models. Table 1 lists the traits of each of three layers of the IoT system. It is important to mention that the three IoT architectures discussed here are not mutually exclusive, instead the aim of this hierarchy is to complement the higher layer by providing it only useful information which makes the system more productive and dependable. For any IoT system designer, the aim is to establish a balance between the capabilities of the three layers keeping in view system costs and requirements.

3.1.1 Cloud Computing Model

This was the first proposed architecture for IoT systems and is based on the premise that processing of data from the various components in the IoT system should take place in the cloud. Cloud computing allows for the remote accessing of uninterrupted shared resources (computing, storage and services) over the network. It should be able to dynamically allocate these resources without human intervention, schedule or pool as necessary and be able to be accessed from a variety of different platforms [35]. The cloud can provide both hardware as well as software services for smart city applications. It has the advantage that it provides a central management platform from which to observe, control the IoT system as well as to disseminate command actions based on the received data. Moreover, this centralization also allows for cloud systems to have sufficiently large computing and storage capacities thereby allowing them to perform complex tasks of data mining, pattern extraction and making inferences from sensor data in smart cities to make use of it in the best manner possible. However, there are a few disadvantages with using the cloud computing model for the IoT. First, transmitting all gathered data to the cloud increases network traffic, even though this may not be true for applications in which measurements are not very frequent, but in other cases, this could increase network costs. Moreover, data transmission overheads may increase due to the large amount of data that needs to be transmitted by the many sensors existing in the smart city scenario. Another disadvantage that the cloud computing model suffers from is data latency, since the sensing units exist at the sensing layer and the decision making/data processing takes place in the cloud, this gives rise to data latency in the transmission of sensing information, especially when many devices start sending data at the same time. Network reliability can be an issue when using this model, with the large volume of data traffic on the network, it might not be possible to enforce robust data transmission schemes as IoT systems get bigger.

3.1.2 Fog Computing Model

Since most of the information produced with in IoT takes place towards the sensing end of the IoT system, also called the edge, Fog Computing was proposed in [36] to overcome some of the problems of the cloud computing model for the IoT. Fog computing provides a more diverse distribution of responsibilities than are dictated by the cloud computing architecture by moving some of the processing to devices on the local network. Typically, Fog computing refers to data processing that is carried out by routers and other network devices within in the Network layer in the IoT. Since network devices nowadays increasingly offer better computational capabilities, one can leverage them to perform rudimentary operations on data. Operations such as aggregation and collection of sensor data, simple processing operations and decision-making can be performed to reduce the amount of information flow towards the higher cloud layer. Questions that need to be answered for the decision-making process include but are not limited to for e.g., does the decision require the use of averaging for one quantity and instantaneous values of the other? Is it possible to extrapolate data received for one quantity and use the currently measured value for another one? Based on the previous data for a given period, can one provide higher layers with decision options rather than just data, thereby providing better quality information to the cloud layer thus resulting in better utilization of cloud resources. Fog layers can localize decision making since they have access to the local state of a given region [37]. This would be helpful in implementing distributed decision making mechanisms which might be necessary in some applications. Moreover, they also allow for local networks to be established using non-internet technologies such as Zigbee, Bluetooth, RFID etc where sensors and other end devices transmit data to the Fog node (also referred to as access points in such systems) which is connected to the cloud.

Fog computing results in reduced costs for deployment of IoT systems, increases robustness as latency, data overhead and errors in transmission are reduced. This also improves the efficiency of the applications as quicker decisions can be made on the received data, which is important in critical decision-making situations. Moreover, Fog devices have the capability to not only receive data from similar devices at the edge but also collect it from many different types of devices. This capability to measure different parameters in the edge environment enable for an application neutral IoT system architecture to be developed.

Data sent upward by the Fog layer in the IoT hierarchy would be used to gain insights in to system behavior and to guide new rules of system operation, this will typically be carried out in the cloud. Devices in the Fog layer may be provided decision making guidance from the higher cloud layer to ensure smooth system operation. However, a balance needs to be struck as to the division of responsibilities between the cloud and the fog layer keeping in view the costs involved.

3.1.3 Edge Computing Model

The purpose of Fog Computing was to push some of the decision making towards the edge of the network. In recent years, with increasingly capable devices being developed that are attached to 'edge' nodes, simple decision making, and processing of data has been increasingly moved on to these devices so as to reduce network and device costs even further at the fog level and make for even deeper distributed decision-making schemes. Edge computing refers to data processing that is done at the "thing" level, i.e. by sensors and other devices in the IoT system [38]. Another concept about Edge computing as discussed in [39] defines the Edge computing layer as an intermediary layer between the Fog and the 'things' (sensors) rather than edge nodes. The difference between them in this case is the Edge computing nodes act as aggregation and decision-making units on a smaller scale compared to fog devices which act to provide seamless connectivity and data integrity throughout the IoT network. The aim of the Fog and the Edge computing paradigms is to decentralize the IoT system for purposes of reducing cost, increasing scalability and increasing robustness.

Table 1. Comparison of Cloud, Fog and Edge Computing Models

Cloud Computing Model	**Fog Computing Model**	**Edge Computing Model**
Contextual awareness on a global level encompassing all aspects of the application	The Fog layer has contextual awareness of the local sensing scenario	Edge devices typically only have information about their own status. Exchange strategy possible but limited to local neighborhood
Farthest away from the edge and therefore decision making can be slow and latency is high	Being the closest unit to the edge, the Fog layer can respond much more quickly to the data being sent from sensors and other devices, as it can aggregate the information sent	Quickest decision making possible; however, decisions will be based on local states
Utilizes heterogeneous data from a variety of sensing devices	Utilizes heterogeneous data, but within a small region	Usually do not have access to different types of data
High network cost	Medium network cost as data flow is reduced	Least network cost
Potential privacy risk as raw data might be sent to the Cloud	Increased privacy compared to Cloud computing	Even greater privacy enforcement possible than Fog computing model
Least robust as decision making is centralized	More robust than Cloud computing model	Most robust as distributed decision making takes place
Best capabilities in terms of resources	Lesser capable than Cloud devices	Least capable
Scalability is low	Scalability is better than Cloud	Scalability is highest

3.2 IoT Challenges for smart Cities

The Internet of Things promises the digitization of all aspects of our lives. For smart cities, this digitization process entails the proliferation of sensing nodes in every domain of a city's operation mechanism. With an application scope this broad, the creation and subsequent deployment of IoT systems in smart cities carry enormous challenges that need to be considered. In this section, a discussion is provided of the challenges that IoT system designers face when making deployments in smart city applications. The focus in this work is on the technological challenges that pertain to IoT use in smart cities and has also been the focus of researchers. Figure 3 shows the different challenges which Smart City IoT system deployment encounters, namely Security and Privacy, Smart Sensors, Networking and Big Data Analytics. A summary of the discussion in this section is presented in Table 2.

Figure 3. Challenges for IoT in Smart Cities.

3.2.1 Security and Privacy

Security, along with Privacy is the primary concern in smart cities. Smart cities involve having essential city infrastructures online, any aberration in the operation of the city's services will bring inconvenience to its citizens and put human lives and property at risk. Therefore, security is a big concern in smart cities. In today's age where cybercrime and warfare have become a tactic in world politics, smart cities are at an ever-greater risk of being the target of such malicious attacks. Encryption of data transmitted over the network is necessary in this scenario. For smart city projects to be successful, they require the trust and participation of citizens. The proliferation of sensors in smart cities, which continuously collect data about the activity of people may expose the daily activities of citizens to unwanted parties. Moreover, companies and corporations on the IoT network may use citizen data without their approval for things like targeted advertising and may perform acts such as eavesdropping etc. Solutions to this will require processes that anonymize data collection while retaining the integrity of the context of the measured task so that apt decision making is possible. Security and Privacy has been covered in detail in a later section.

3.2.2 Smart Sensors

Smart sensors are the hardware components that gather data in smart cities. These devices are manufactured by a host of different vendors that adhere to different sensing mechanisms, standards of measurement, data formats and connectivity protocols. Smart city deployment will require all these devices to exchange data, perform scheduling of tasks between them and be able to aggregate data together for making inferences. A solution to this issue is to develop and use open protocols and data formats that will enable manufacturers to create equipment that can communicate between each other, further spurring IoT system deployment. Another solution could be the development of 'standard' access point nodes for IoT systems that can communicate to devices using several different communication protocols and are able to decode the information received. Some manufacturers have, in fact, made their products compatible with other protocols as mentioned in [40].

Another challenge for smart sensors is reliability and robustness. Reliability and robustness refer to the dependability and correctness of the IoT system. IoT is the backbone of future smart cities and being imperative to their operation, the IoT system needs to provide a smooth experience to its users. This requires that service requests from users of the application receive an accurate and timely response. The quality of service needs to be ensured for every citizen in the smart city. Systems that deliver critical utilities such as transport, electricity etc. should be decentralized. The distributed connection points will allow for robustness and increase reliability. One such example is self-healing in Smart Grids [23].

Many current networking protocols are developed for infrastructure networked devices which have access to continuous power, however, sensors in smart cities will be mobile in many scenarios and thus be battery powered. Moreover, they will need to measure, transfer and in some cases save data they have collected. This requires the development of not only low power, low overhead data transmission schemes but also development of new memory and storage technologies as well as low power devices that extend battery life as much as possible. Storing this large amount of data would require development of compression algorithms which will be employed and database schemes that will need to be developed in the future as smart cities and IoT are scaled up. Solutions for power issues necessitate the development of new battery technologies and perhaps the incorporation of energy harvesting mechanisms in such devices to make long lasting service provision possible.

3.2.3 Networking

The IoT depends on the capability of sensing and other devices to be able to send and receive information to each other and the Cloud. With new smart city applications coming up, providing networking to these devices to remain connected is a big challenge. Current networking methods are not optimized to providing networking services for smart city components. Many devices in smart cities have mobility and data throughput requirements which need to be met to provide an acceptable quality of service. Different approaches have been suggested in terms of defining access points, local networks etc. to solve this problem. Another aspect of networking would be working on efficient and dynamic routing protocols that can serve IoT requirements capable of working with stationary as well as devices in motion, which many current protocols do not offer sufficiently [41].

3.2.4 Big Data Analytics

IoT connected devices generated 13.6 Zetta Bytes of data in 2018 and this is expected to grow to 79.4 Zetta Bytes till 2025 [34]. To make use of this data and continuously improve the services that are delivered in smart cities, new data analytics algorithms need to be developed. With the myriad of the different parameters that are measured in smart cities, these algorithms need to be applicable to data of varying nature (both structured and unstructured), better data fusion techniques need to be developed as well so as to combine them in meaningful ways and be able to extract inferences and recognize patterns. Deep learning has been of interest in this area as it can leverage on this large amount of data to provide better results for different applications. Another important consideration would be to ensure that the developed algorithms are scalable in that they have enough generality and can be used through out the intended application. For, e.g., for the purposes of activity recognition, the authors in [42] found that a CNN trained for activity recognition on one dataset failed to perform well on others or in [43] where the deep learning network performs poorly when the color of tomatoes is different from what it was trained on. Concept drift is another issue of concern as with the continuous acquisition of data, the properties of data may change over time. Techniques such as incremental learning may be useful in this respect. Explainability is another important factor for Smart City analytics to be widely acceptable, specially in the area of smart health. There have been some approaches suggested towards this end, in [44] a hybrid deep learning classifier and semantic web technologies based solution is demonstrated for the application of flood monitoring. In [45], the authors present an explainable deep learning based healthcare system at the Edge for COVID-19 care based on a distributed learning paradigm with promising results. However, more work needs to be performed to incorporate explainability techniques such as distillation, visualization, and intrinsic methods into Machine and Deep Learning based smart city applications in order to increase smart city application proliferation.

3.3 Sensing Technologies

Sensing is at the heart of smart city technologies. Sensors provide the knowledge and data from which smart city innovations are created. With the vastly different nature of Smart City projects and its various components, there are numerous sensors which are used as part of these initiatives. The authors in [58] have provided a framework for the comparison of IoT sensors and have listed sensors they found in use for the Internet of Things. We use their work to direct our survey of the sensing technologies used in smart cities. Sensors within IoT can be divided in to several groups, these are ambient, motion, electric, biosensors, identification, presence, hydraulic and chemical

Table 2. Smart City IoT Challenges and Mitigation

Challenge	Mitigation/research direction	References
Security and Privacy	Encryption	[46][47][48][49][50][51][52]
	New authentication mechanisms New standards to anonymize data Prevent data leakage	[53][54][55]
Smart Sensors	Interoperability: Open Standards Reliability and Robustness: Decentralized and distributed architectures and decision making Power and Memory: Energy harvesting, Low power sensors, New database storage systems	[46][47][48][52][53][54][55][56]
Networking	Low power networks, Network schemes that ensure fluent mobility and routing	[41][52]
Big data analytics	New algorithms which work with different natured data, Develop scalable and explainable AI	[47][52][57]

sensors as shown in Figure 4. Sensors are the key component in smart city IoT systems which provide the interaction between the smart city system and the city's inhabitants and allow for new services to be developed. One thing to note is that many of the sensors find applications in multiple areas as discussed. Furthermore, any given application will require measuring different physical quantities and will require the use of many different types of sensors. For, e.g., ambient, motion, electric, identification, position, chemical and hydraulic sensors have been found to be used in smart homes. As noted in the challenges, working with different sensors which might have different output data types is a task that needs to be dealt with when working with multiple types of sensors. Table 3 presents a summary of the sensors used in each smart city component.

3.3.1 Ambient Sensors

Ambient sensors include sensors used to measure physical quantities indicating to environmental conditions such as temperature, humidity, light intensity and pressure. Ambient sensors are used in a variety of smart city applications including smart homes where they are used to regulate the comfort level, they are also used for smart city services.

3.3.2 Bio-Sensors

Bio-sensors are used for measuring health parameters of living things. Bio-sensors in smart cities are used for monitoring patients for healthcare purposes. Such sensors include ElectroEncepheloGram (EEG), ElectroMyoGram (EMG), ElectroCardioGram

Figure 4. Sensing Technologies for IoT Smart Cities.

(ECG), skin resistance, heart beat, breath sensors, pulse oximetry, blood pressure and more.

3.3.3 Chemical

Chemical sensors are used to measure chemical properties of materials, this includes gas sensors which can measure/detect carbon monoxide (CO), carbon dioxide (CO2) and other gases for air quality monitoring, sensors for detecting smoke, pH and other sensors for water quality monitoring etc.

3.3.4 Electric Sensors

Electric sensors allow for the measurement of electrical power and are widely used in smart grids and smart homes to monitor the power consumption of consumers/appliances. Types include current transformers and voltage sensors to measure current and voltage, respectively.

3.3.5 Hydraulic

Hydraulic sensors refer to sensors used for liquid measurements such as level, flow, leak detection. These are used for measurement of liquid levels in tanks [59].

3.3.6 Identification

Identification sensors refer to RFID tags and Near Field Communication (NFC) devices. These sensors are used in applications involving payments, data exchange in the domain of smart transportation and smart city services.

3.3.7 Motion Sensors

Motion sensors refer to sensors that can be used for the detection of motion. Sensors for motion sensing involve inertial sensors such as accelerometers and gyroscopes. These sensors are used in smart health applications such as activity tracking as well as applications like vibration sensing in smart homes and industry.

3.3.8 Presence

Presence sensors indicate to the presence of a humans or objects. Passive InfraRed (PIR) sensors are very popular and are used to detect human motion, reed switches can be used on windows and doors for security purposes, inductive loop sensors which use electromagnetic induction can be used to detect presence in transport systems. Ultrasonic sensors are also used to determine the distance of objects. Capacitive sensors are also included in this type, these may be used to determine position.

3.3.9 Other Sensors

Various smart city applications make use of different sensing modalities such as audio or visual information or other signal measurement devices, for, e.g., bluetooth and Wi-Fi signal strength. Since the sensors for these modalities capture raw information about signals (visual, sound or signal strength etc.), the gathered data is typically processed further before it indicates to the target variable being measured.

3.4 Networking Technologies

The internet of things in smart cities depends on the aggregation of data measured by individual sensing units placed throughout the smart city environment. Systems that can use these measurements individually have long existed and provided automation for small projects. However, the 'smart' in smart city comes from the collective usage of the data from these individual sensing units to perform complex decisions while delivering services to citizens. The collective use of this data enables its analysis over a wider scope compared to individual levels so as to determine long term patterns and provide meaningful insights to support services. The number of such IoT devices currently present in the world [34] are multiple times that of the world population. To enable these devices to exchange data, wireless technologies need to be used as physical connections would, for one, be too costly (where ever they can be used), second, would not satisfy the mobility requirements that are typical of many smart city applications. The internet has provided connectivity to computers, smartphones and other electronic devices around the world with each other, allowing for instant

Table 3. Sensing Technologies for IoT Smart Cities by Smart City Component

Smart City Component	Sensor Type	References
Smart Agriculture	Ambient, Chemical, Hydraulic, Other sensors	
Smart City Services	Ambient, Chemical, Hydraulic, Presence, Other sensors,	[60][61][62]
Smart Energy	Ambient, Electric, Motion	[63][64]
Smart Health	Biosensors, Identification, Motion, Other sensors	[65][66][67]
Smart Home	Ambient, Chemical, Electric, Hydraulic, Identification, Motion, Presence, Other sensors,	[63][68]
Smart Industry	Ambient, Biosensors, Electric, Hydraulic, Identification., Motion, Other sensors	
Smart Infrastructure	Ambient, Motion, Electric, Other sensors,	[60]
Smart Transportation	Ambient, Chemical, Identification, Motion, Presence, Other sensors	[69][70][71]

transfer of information between them. However, for IoT the internet may not necessarily be the only communication method [72] as many applications do not possess edge devices that can connect to the internet. An application may consist of a local network of sensing units which can exchange data between them and rely on a multi-hop communication protocol to send data to a central node, hub or gateway. The gateway might be fixed and would be connected to the internet, thereby relaying any monitored data to the cloud for further processing or use. It might also be possible that devices within an application may use many different protocols with the central node having the capability to communicate with all of them, a common case for such architectures is the smart home where manufacturers produce devices using propriety or incompatible protocols for which a hub may be used, an example of such a system was provided in [73] and several hubs are commercially available. In this section, we discuss the network types, topologies and protocols used in the Internet of Things for Smart City applications as illustrated in Figure 5. We later provide a comparison of these protocols in Table 4.

3.4.1 Network Topologies

There are three IoT network topologies, point to point, star and mesh [74]. The first type of topology is the point to point topology where devices are connected to each other sequentially in a point to point manner. Point to point networks introduce data hops for packets that need to be sent to other nodes as data needs to go through each node in the path of the two nodes wanting to exchange data. Point to point networks are not very popular in IoT systems as it ranks low on fault resiliency and will breakdown if there is a fault in any of the intermediate nodes. In Star topology, all units in a network are connected to a central node or gateway and cannot send data to each other directly. In order to perform an exchange of data among themselves, the devices need to send it through the central node. Star topology networks, with their central node structure provide a natural aggregation scheme for data collection within the internet of things, however, large networks consisting of many devices, which can be the case in most smart city applications, may result in high latency and possible bottlenecks in high information throughput scenarios. Star topology has been used in various applications including disaster management [75] and environmental sensing [76]. The last type of network topology that is used in IoT is the Mesh network topology, mesh networks allow all individual devices to communicate between them. By enabling communication between the nodes in a network, mesh topology offers a larger range as data transmitted towards a certain node can make multiple hops through the network, this also increases the networks resilience as alternate paths could be used if packet delivery fails due to any node becoming faulty. In fact, such topologies have been used in smart homes [77] as well as in smart grids [78]. There are other topologies which have not been mentioned, for, e.g., tree (which has multiple star networks connected in a point by point fashion).

3.4.2 Network Architectures

Network Architecture refers to the structure of the network used for a given application. As discussed earlier, the 'things' in IoT may not necessarily be connected through the internet, in fact a distributed connectivity structure may be implemented with only one unit in the network being capable of sending data to the cloud depending on the requirement. Work in [72] mentions three types of network architectures that are used for smart cities based on IoT. These are Home Area Networks (HANs), Wide Area Networks (WANs) and Field/Neighborhood Area Networks (FANs/NANs). Home Area Networks are short range networks and are usually used to transmit information to a central node which is responsible for data collection before it is sent to the cloud. Communication within the network is performed using some low power communication protocol such as Zigbee, Bluetooth, Wi-Fi etc. HANs are very popular in smart homes where they are used to gather power consumption and times of operation data from a multitude of appliances which are then sent to a smart meter as part of a smart grid [79]. The second type of network architecture is Field Area Networks (FANs), sometimes also called Neighborhood Area Networks (NANs). Field Area Networks have a larger communication range than HANs and are used to provide connection between a customer (for, e.g., in a smart grid) to the utility company. Wide Area Networks are used for network structures that require communication over large distances. These networks are not as dense as HANs or FANs and utilize technologies such as cellular services, wired connections such as fiber optics as well as a class of low power protocols designed for WANs themselves [80]. WANs are used in a variety of smart city applications including Smart Grids where they are used to connect multiple substations together or exchanging data between the customer and the substation [81].

3.4.3 Network Protocols

The type of network to use depends on the requirements of the application. It is imperative that the communication protocol used in a smart city application meet the desirable quality of service (QoS). Several protocols have been used in the internet of things for smart cities [48, 82, 83, 84], herein, we discuss the traits of the most popular wireless networking protocols used in smart cities.

3.4.3.1 RFID Radio Frequency Identification (RFID) utilizes radio frequencies to transmit and receive data. RFID communication consists of two types of devices, one device is the Reader and the other is called the Tag. The Reader is usually powered and once a tag comes in the vicinity of the reader, an exchange of information takes place after authorization as the tag harvests the energy from the reader. Such tags are called passive tags, there are also active tags which are do not depend on the reader for their power. Depending on the standard, RFID can operate on different frequencies in the radio frequency spectrum between 125 KHz to 928 MHz and can be used over short ranges. They are used in applications such as smart transport (toll tax collection, parking), smart health and more.

3.4.3.2 Near Field Communication Near Field Communication (NFC) is very similar to RFID, however, the structure of NFC communication doesn't consist of tags and readers. Unlike RFID, both devices which want to communicate using NFC need to be powered and data tranmission/reception can take place in both directions unlike RFIDs. This enables the use of NFC to control and configure devices unlike the RFID which cannot be used for measurement or control tasks. NFC utilizes similar frequencies to RFID but is used for very short distances. NFC devices are popular for applications involving payment using smart phones and are also used in smart homes.

3.4.3.3 Bluetooth Bluetooth is a low energy protocol popular in IoT applications as it can support an unlimited number of nodes [82]. The protocol is designed for short range, low bandwidth communication in an arrangement where devices can easily exit or enter the network. Bluetooth natively supports the star topology as it has a master device at the center of the communication mechanism. It operates in the 2.4 GHz ISM band and can have maximum data rates of 2 Mbps. Bluetooth has been widely used in smart home due to it providing a direct connection interface to smart phones without the need for any intermediary hub device.

3.4.3.4 Z-Wave Z-Wave or Zensys wave is a low power protocol developed to be used in home automation applications. It is a low speed protocol with a short range, operating in the frequencies of 868 MHz and 900 MHz. It operates in a master slave fashion where a master can have multiple slave devices which can respond to commands from the master node. Therefore this is well suited for applications where a central control element is present and needs to gather data from multiple sensing units such as smart homes and smart healthcare systems.

3.4.3.5 Li-Fi Li-Fi (Light Fidelity) uses visible light instead of radio frequency (RF) to exchange data. The advantage with using Li-Fi over RF communication is that it can utilize already present lighting systems which also results in conservation of power [85]. It offers very high speeds of data transfer for short distances and has been used in parking systems.

3.4.3.6 Wi-Fi Wi-Fi (Wireless Fidelity) operates using wireless frequencies in the 2.4 GHz and 5 GHz bands to provide high speed internet connectivity in a limited distance. Wi-Fi is popular in many smart city applications as it provides ready to use interface to smart phones, computers and other wearable gadgets.

3.4.3.7 Zigbee The ZigBee protocol was developed as a low power low cost protocol for wireless sensor networks (WSNs) and has evolved to be used in the Internet of Things. The ZigBee protocol operates in the 868 MHz/915 MHz/2.4 GHz band and offers moderate data transfer speeds with distances similar to Wi-Fi in a multi-hop data transfer scheme. Zigbee radios are low cost devices and therefore it is a popular protocol used by many manufacturers of smart home, smart healthcare devices. A ZigBee network will have three devices, one called the coordinator which is the controller of the network, the router which is responsible for moving data to other devices and the ZigBee end device (sensors and actuators).

3.4.3.8 Wi-SUN The Wireless Smart Utility Network (Wi-SUN) is a network approved by Institute of Electrical and Electronic Engineers (IEEE) and is used in field area networks for utility metering, automation of distribution for utilities such as electricity, gas etc. and also for demand response systems for utility-based applications. It supports IPv6 addressing and can be used in star or mesh configuration where it also allows for multi-hop communication [86].

3.4.3.9 Cellular Cellular technologies refers to 3G, 4G and 5G communications. Along with Bluetooth and Zigbee, they are the most popular IoT enabling technologies. Cellular communication provides high data rate and supports more content rich applications compared to the other protocols. With the long range they provide, they are preferred for a variety of applications where power is not an issue. Depending on the technology, cellular bands range from 600 MHz to 80 GHz with very high data rates.

3.4.3.10 LoRaWAN LoRaWAN stands for Long Range Wide Area Network (LoRaWAN) and it is a Low Power Wide Area Network (LPWAN) that consists of several gateways and multiple end devices with the gateways connected to a back-end network server. The back-end server provides connection to the cloud. End devices do not have a fixed association with a specific gateway and may send data to multiple gateways when it needs to transfer data to the cloud.

3.4.3.11 6LoWPAN 6LoWPAN which is short for IPv6 over Low Power Networks was created by the Internet Engineering Task Force (IETF) specifically for internet of things applications with the aim of making it possible for providing internet connectivity to small devices. It is an IP based network and leverages IPv6 communication. This is a short-range network operating in Industrial, Scientific and Medical (ISM) bands.

3.4.3.12 SigFox SigFox is a proprietary standard developed by SigFox Inc., France. It uses unlicensed bands to perform ultra-narrowband bidirectional communication with low speeds [87]. SigFox has a similar architecture to LoRaWAN and like LoRaWAN and 6LoWPAN, SigFox is a popular LPWAN in the IoT domain offering sufficiently large distances of communication of up to 50 km. SigFox finds applications in security in buildings, smart lighting and environmental monitoring.

3.4.3.13 NB-IoT NB-IoT (Narrow Band IoT) is a type of LPWAN which operates on Global System for Mobile Communications (GSM) and Long-Term Evolution (LTE) bands. In fact, it can operate using the same hardware with a software upgrade as it is considered a bare bones version of LTE. It allows for connecting up to 100,000 devices per cell.

Table 4. Comparison of Network Technologies for IoT Smart Cities

Architecture	Technology	Frequency/Medium	Data rate	Range	Topology
Home Area Networks (HANs)	NFC	125 KHz, 13.56 MHz/860 MHz	106 Kbps, 212 Kbps or 424 Kbps	10 cm	Point to Point
	RFID	125 KHz, 13.56 MHz/902-928 MHz	4 Mbps [82]	3 - 10 m	Point to Point
	Li-Fi	LED Light	1 – 3.5 Gbps [85]	10 m	Point to point, Star, Mesh
	Bluetooth	2.4 GHz	Up to 2 Mbps	240 m	Star
	Z-wave	868 MHz/900 MHz	40-100 Kbps	30 – 100 m	Mesh
	Zigbee	868 MHz/915 MHz/2.4 GHz	250 Kbps	Up to 100 m	Mesh, Star, Tree
	Wi-Fi	2.4 GHz/5 GHz	54 Mb/s, 6.75 Gb/s	140 m , 100 m	Tree
	6LOWPAN [82]	868 MHz/915 MHz/2.4 GHz	Up to 250 Kbps	10 - 100 m	Mesh, Star
Field/Neighborhood Area Networks (FANs/NANs)	Wi-SUN	868 MHz/915 MHz/2.4 GHz	Up to 300 Kbps	Up to 4 Km	Star, Mesh
Wide Area Networks (WANs)	NB-IOT	Licensed LTE bands	200 Kbps	1 - 10 Km	Tree
	LoRaWAN	433 MHz/868 MHz/915 MHz	50 Kbps	5 - 20 Km	Star of Star (nested star)
	Sigfox	433 MHz/868 MHz/915 MHz	100 bps	10 – 50 Km	One hop star
	3G	1.8 – 2.5 GHz	2 Mbps	-	Tree
	4G	600 – 5.925 GHz	up to 1 Gbps	-	Tree
	5G	600 - 80 GHz	Up to 20 Gbps	-	Tree

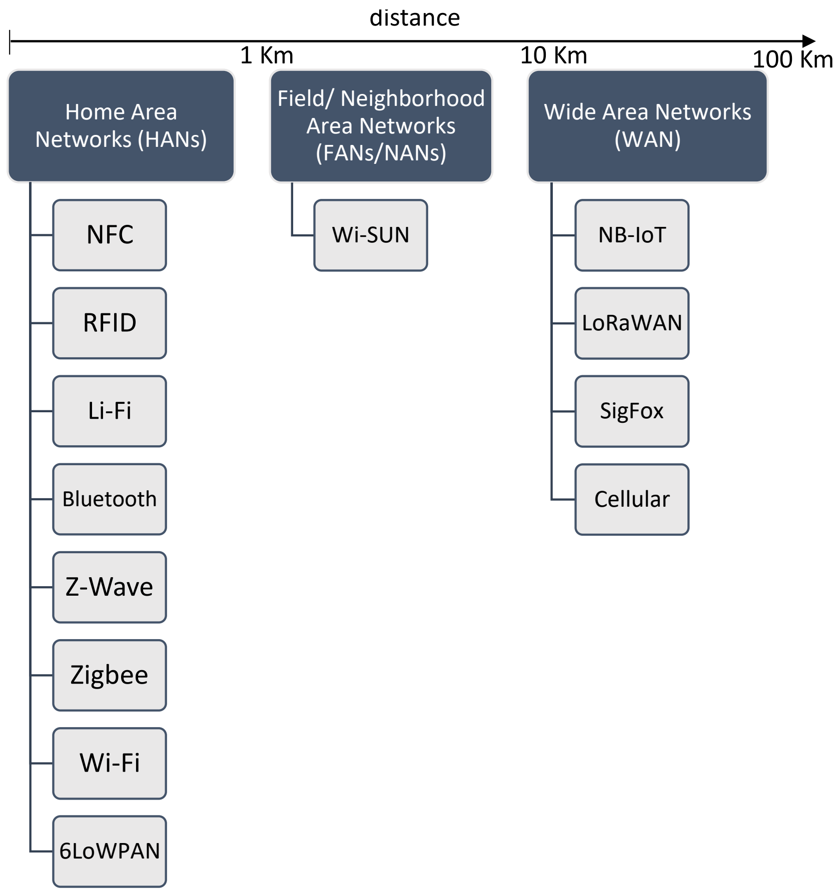

Figure 5. Network Technologies for IoT Smart Cities.

3.5 Security and Privacy in Smart City IoT

Smart Cities involve the transmission of sensing data, control information through the internet as well as local networks. Moreover, several components in smart cities tend to critical aspects of a citys operation and are highly intertwined with the social and private life of its citizens. Consequently, security and privacy in Smart Cities is of great importance and has been of high interest to researchers [51, 88, 89, 90, 91, 92]. The topic of Security of IoT has been covered in [93] who deliberate upon the challenges faced in the different architectures of the IoT and present issues and solutions. We cover security for IoT in smart cities so as to highlight issues that are pertinent to in the Smart City context and to complete our discussion on this topic.

Smart Cities are enabled by collecting data through sensors within a city as well as its populace, process it and then mine it to provide a better quality of life to the people living. These sensors can provide an estimate about the internal state of a city's components such as transportation, power system, building condition/state, human mobility and more. All of this data is sent to the cloud where it is processed and mined. However, there are several issues that pertain to how these data are sent and used and raises questions about integrity, protection and the confidentiality

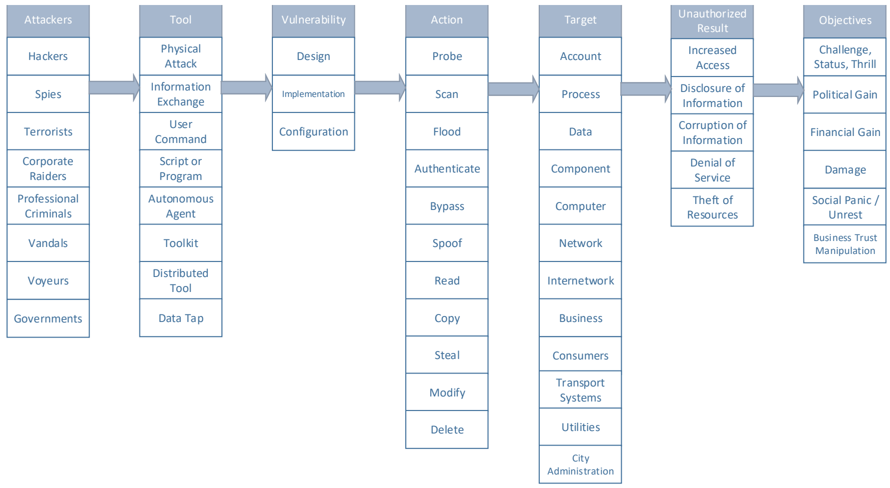

Figure 6. Modified CERT Attack Taxonomy for Smart Cities.

of this process. In fact, this concern is not unwarranted for, in 2015 an attack on the Ukranian power grid which left 225,000 people without power [94] opened the worlds eyes to the very real threat posed by cyber attackers. Data gathered in Smart City applications can be used to perform many undesired acts, GPS devices that are present in every phone and most vehicles are susceptible to providing information about a persons location, habits as well as lead to privacy issues as discussed in [91], power consumption and ambient sensor data from a building may indicate to occupancy [95] and even indicate to the individual identities [96]. This information may be used by bad actors to carry out unlawful acts causing risk to life and property.
To secure the Internet of Things for Smart Cities, typical security schemes might not be as effective in many cases and new methods will need to be developed to cope with the security and privacy issues in IoT for Smart Cities. In order to provide a standardized framework and terminology for discussing security attacks, we adapt the standard attack incident taxonomy [97] suggested by the Computer Emergency Response Team (CERT) which was established by Defense Advanced Research Projects Agency (DARPA) for use towards IoT for Smart Cities. This is shown in Figure 6.

There are different types of security and privacy issues in IoT in Smart Cities, they exist on each of the three levels of the IoT architecture, application software layers, network layer and the perception layer along with some system wide issues, a discussion is provided for each of them. Furthermore, Table 5 provides a summary of the security and privacy issues in IoT for Smart Cities and the counter measures that one can take to mitigate them.

3.5.1 Application Software Layers

Application software layers are comprised of the Middleware, Application and Business Layers. Security and privacy in the application software layers tend to issues relating to storage of data and its usage. These are data visibility, access and injection.

3.5.1.1 Data Visibility/Identification Once data is gathered, it is sent to the cloud where it is stored and mined to make inferences. Since the cloud would be used by multiple entities with different standards of security protocols and practices, it is extremely important that data stored in the cloud is encrypted so as not to allow its exposure to unwanted entities. Any data stored in plain form would be a risk to not only user privacy but also company rapport.

3.5.1.2 Data access/Secondary Use Access control is also a major issue in smart city data. Most Smart City applications rely on the usage of data from different applications to provide smart services, thus resulting in the gathered data to be used by many different enterprises. To allow this to take place smoothly while preserving privacy, suitable access control schemes will need to be devised to allow for responsible access to users of this data. A well-defined hierarchy of data users will need to be developed and implemented to limit access only to intended authorized personnel. Information flow control should also be employed that can track data flow as access to it is made and to detect any violation of access or usage rules. Moreover, data mashups that will occur in the cloud where multiple entities merge their data to work on some common goal, should be carried out with proper oversight. Data anonymization may also be required in such a scenario where specific values may be converted in to a range perhaps (e.g., using k-anonymity [90]) and unwanted data be deleted as necessary. Blockchain has been suggested to be used for access control as well as access tracking of users in IoT applications where each user access to a service or an application ends up as a transaction to form the applications IoT trail [98].

3.5.1.3 Data Injection/Data Integrity Data injection refers to the injection of false information or modification information about a user in the system after gaining access. Since data is typically stored in databases, SQL injection involves an attacker inserting queries to modify data or make false data insertions into the database. This can have far reaching consequences for smart city applications such as patient record manipulation as well as manipulation or deletion of government records. SQL injection prevention schemes involve the validation of data before using it [99] such as positive pattern matching. It also involves limiting database access based on the user requirements and performing penetration testing.

3.5.2 Network Layer

Like other networks, the IoT in Smart Cities are also susceptible to network attacks such as Denial of service, Eavesdropping, Man in the middle attack, Side Channel and spoofing attacks [100]. We discuss these attacks and discuss remedial actions that should be taken to prevent them.

3.5.2.1 Man in the Middle Attack Man in the middle (MITM) attack refers to the interception of data on the network by faking the identity of a network node or device. This is carried out by appearing as the intended recipient to the sender and the original sender to the recipient by unauthorized actors. To prevent this, the exchange of data between two entities should employ cryptographic protocols which can ensure secure communication. Many public networks do not make use of encryption when exchanging data, this endangers user data and can give access to unwanted persons to user information [90]. Policies need to be devised to allow for suitable data communication standards for such networks while keeping the user as well as commercial interests in mind.

3.5.2.2 Eavesdropping/Sniffing Attack Eavesdropping refers to the listening of data on the network. In eavesdropping an unauthorized entity joins the network and can listen to the data that is being exchanged between the devices on the network. To avoid this, strategies include the use of authenticate always protocols which initiate authentication steps whenever devices need to communicate with each other. This will ensure that no unwanted users are allowed access to network traffic to prevent such attacks. Moreover, industry standard security protocols such as the Transport Layer Security (TLS), Wi-Fi Protected Access 2 (WPA2) should be used for authenticating remote access. Furthermore, remote sessions should have time-outs implemented to ensure that mistakes are not made by forgetful employees.

3.5.2.3 Side Channel Attack Side Channel Attacks refer to the extraction of information by observing operation characteristics of the implemented computer algorithms or systems such as power consumed, time taken, traffic analysis, fault analysis, acoustic analysis [101]. Side Channel attacks don't give nonpermitted parties access to the data within the network but may allow them to determine important information about the system, such as the protocol used, or allow them to drop packets so as to degrade performance of the network. One solution to counter network traffic side channels attacks is to saturate network bandwidth to prevent patterns from being observed. Another popular method of preventing side channel attacks is to make use of masking [102].

3.5.2.4 Denial of Service Attack Denial of Service (DOS) or Distributed DOS (DDOS) attacks involve an entity getting access to the network and using legitimate nodes within it to flood the target with unnecessary requests to consume network bandwidth and degrade quality of service. With a smart city depending on sensors to provide it with a 'view' of the city, DOS attacks can make the smart city system blind which can lead to loss of property and life. Countermeasures for DOS attacks involve anomaly detection by monitoring of network data to check for any irregular behavior. Artificial Intelligence has found applications in this area as noted by [103], for, e.g., it has been used to identify abnormal data in smart grids [104] as well as detecting attacks [105, 106].

3.5.2.5 Spoofing Attack In a spoofing attack, an attacker adds itself to the network by appearing to be a legitimate device on the network, thereby allowing them to send irregular or abnormal data to upset normal operation of a smart city system. Due to the varying nature and types of IoT devices having different levels of built-in security, spoofing is a particularly dangerous attack for IoT systems. Methods to prevent spoofing are the use of cryptography, hybrid encryption [107] as well as the use of blockchain [108] to validate data exchange and as well as authenticate devices.

3.5.3 Perception Layer

Perception layers attacks refer to physical attacks on an IoT based smart city system. This requires the physical presence of an attacker near the sensing elements in an IoT system. We cover these attacks to provide a holistic assessment of the security and privacy issues in IoT Smart Cities. Tempering and Radio Interference (jamming) are attacks which can affect the performance of smart city systems. Tempering may occur during the development or the manufacturing process while jamming can take place due to a generation of radio frequencies which interfere with the frequencies used by devices on the network to exchange data. To circumvent this, policies may be incorporated into software which alert businesses and other Smart City partners to missing or abnormal data from sensing devices.

3.5.4 System Wide Issues

3.5.4.1 Data Leakage Data leakage refers to the unintentional conveyance of information about subjects in smart cities. The many data sources in smart cities may contain information related to a user's identity, health, quality of life etc. Smart City application managers use this data to improve their services and provide a better user experience, however, it is possible that this data might be shared with third party entities. It is therefore necessary to anonymize data before such assignments are taken. Data leakage can also take place when devices within a network perform discovery tasks and may provide personal information to rogue nodes in the network. In Smart Grids, power consumption data should be anonymized by considering the data on a neighborhood level rather than on the individual level, moreover, systems could be installed in homes using batteries to modify the demand response signal. Data aggregation is an important tool in preserving individual privacy and preventing data leakage. Moreover, data should be encrypted when sending it over the network so that any unauthorized access is avoided. Another strategy would be to use data minimization. Typically, sensors used in smart cities will gather data of less 'interest' in addition to data of interest. For, e.g., navigation systems many times record location information even when not in use or video applications such as facial recognition systems typically record other activities apart from being activated whenever a face is observed in the video. This extra information increases the risk of data leakage. Data minimization can be employed in such cases to limit the data that is being gathered on the user.

3.5.4.2 Trustworthiness Apart from the technical steps that need to be taken to provide a secure usage experience of smart city applications and to encourage its use, it is important that smart city users be provided clear policy guidelines to how

Table 5. Security and Privacy issue for IoT Smart Cities

Layer	Issue	Countermeasure
Application Software Layers (Middleware, Application and Business)	Data visibility/Identification	- Use of encryption to store data
	Data access/Secondary use	- Access control schemes based on user hierarchy - Data anonymization be employed - Use of blockchain for tracking user access
	Data injection/Data integrity	- Use of data validation before usage - Limiting data access - Query parameterization - Penetration testing
Network Layer	Man in the middle attack	- Use of cryptographic protocols for data exchanges - Encrypting data on public networks
	Eavesdropping/Sniffing attack	- Use always authenticate protocols - Remote access should use industry accepted protocols such as TLS, WPA2 - Timeouts for remote sessions
	Side channel attacks	- Bandwidth saturation - Masking to prevent similar operational patterns
	Denial of Service	- Check irregular data requests (AI has been shown to be of use here)
	Spoofing attacks	- Use of cryptography - Use of hybrid encryption - Use blockchain to validate data exchange as well as authenticate devices
Perception Layer	Tempering and Jamming	- Software policies for missing data
System Wide	Data leakage	- Data anonymization - Data minimization - Data aggregation
	Trustworthiness	- Provide clear policy guidelines to users - Flexible policy development in consultation with users

companies providing them these services will manage their data. Transparency in this regard will help increase user trust and the feedback will enable companies to develop better data privacy mechanisms. Another way trust could be developed would be for the companies to provide customers who are vary of data collection, certain options in policies where they could choose which parts of the data collection are acceptable to them and which aren't.

4 SWOT Analysis

To complete this discussion, we perform a Strength Weaknesses Opportunities Threat (SWOT) analysis on the use of IoT for Smart Cities that discusses the strengths that IoT offers for Smart cities, the weaknesses in the current implementation scenario, the opportunities which exist for future work in this area as well as the threats that IoT application to smart cities faces, a summary of our discussion has been given in

Table 6. SWOT Analysis for IoT in Smart Cities

	Positive	**Negative**
	Strengths	**Weaknesses**
Internal	- Sustainable living - Improved quality of life - Efficient city operations - Well suited for big data algorithms - Scalability of applications - Real-time/fast response due to distributed IoT structure - Reduced costs - Robustness - Enable heterogenous system connectivity	- Lack of data control policies - Laws need to be developed - Interoperability of networks - Incompatible sensor standards - Myriad of different application frameworks
	Opportunities	**Threats**
External	- Development of new sensor technologies. - Development of low power and higher speed communication schemes - Development of Encryption techniques for storage and data exchange - Development of Data processing for privacy preservation techniques - Development of new city services - Development of scalable, explainable AI	- Trustworthiness issues among users - Network attacks - Data theft - Data leakage

Table 6.

4.1 Strengths

IoT smart city strengths are the fact that they provide an improved quality of life for a city's population along with reduction of costs in terms of operation and also enable cities to be sustainable. IoT enables sensors and devices to be deployed throughout a city to give an overview of the state of the city's main functions such as transportation systems, electric, water and gas distribution as well as crime monitoring to name a few. This real time information helps city administration, businesses and other stakeholders to provide better services to people, increase the effectiveness of those services and reduce the cost through efficient operation.

On the technical side, IoT data has made possible to use data analytics to gauge various aspects of the multitude of services which are being provided in the city as well as to determine interactions between them and utilize that information for better decision making to make life easier for citizens. Furthermore, the distributed nature of IoT systems and the flexible architectures which enable fluidity through movement of sensing units is easily scalable thereby requiring little additional cost to upgrade and expand currently deployed systems. Moreover, this distributed architecture also makes such systems very robust to faults thereby increasing reliability of deployments and offering self-healing in applications such as electricity systems.

4.2 Weaknesses

IoT in Smart cities do suffer from some weaknesses in terms of technology, for, e.g., the current deployment scenario has a myriad of different technologies relating to networks, hardware platforms and software frameworks which do not often work together very well as discussed. Different standards' bodies such as the IETF, European Telecommunications Standards Institute (ETSI), IEEE and other organizations have been contributing with standards for communication, network discovery, identification, management of devices etc. However, the sheer number of 'standards' with many of them not being compatible with each other has not fully solved the interoperability problem and this can cause hurdles for expansion of IoT systems without a significant overhaul of system components. Another problem currently facing IoT systems is the lack of data policies and legislation. The concern here is that data policies are not mature enough to regulate how data is handled in IoT systems, as has been discussed previously. This is a major problem given the growing issue with user data privacy in a connected world.

4.3 Opportunities

IoT in Smart cities presents many opportunities to researchers and businesses alike in lieu of mitigating the weaknesses and also in the provision of new city services. The data gathered by the sensors in IoT systems has the potential to provide a holistic overview of the city's state allowing for the use of big data algorithms to develop new applications and services. For researchers in the data analytics domain, this heterogenous data provides a wonderful opportunity for the development of new data science algorithms for service delivery. There is a large monetary value towards the development and usage of computationally cheap encryption techniques, efficient data storage methods and networking technologies to make IoT deployment easier and cheaper. Development of new sensor technologies is another opportunity for researchers in IoT for smart cities. The development of newer, efficient, low-cost sensors would aid to the creation of IoT services and enable even wider usage.

4.4 Threats

With a connected system, there are several threats that come with IoTs for Smart Cities involving trust issues among users, privacy concerns due to network attacks, potential data theft etc. Privacy and security are the most important concerns of IoT applications, with such a personalized interaction mechanism between people and devices as is the case of smart cities, the risks for privacy breaches, data theft and leakages are high and this is a constant concern for service users as well as providers. Numerous attacks on Smart City systems have exposed the vulnerability of this technology to cyber attacks and also demonstrated the consequences that it has on the population. Traditional security procedures and methods such as access authentication, routing and networking might not be enough or possible in many IoT deployments due to IoT devices typically not having sufficient computing capabilities, this has exacerbated the privacy and security concerns for IoT stakeholders. This can also feed to a lack of trust by customers to participate in smart city applications.

5 Conclusions

This chapter presented a broad coverage of the Internet of Things in Smart Cities. Providing a detailed discussion of Smart Cities and its different domains, IoT was presented as a vital enabler of smart city services and the various smart city architectures and the challenges that are faced in the deployment of smart city applications were deliberated upon. This was followed by a review of the sensing and networking technologies used for such applications Finally, the security and privacy issues faced by IoT based Smart Cities were discussed and a SWOT analysis is provided.

CHAPTER III

AI IN IOT SMART CITIES

1 Introduction

This chapter provides insight into different ways in which AI has been applied in the IoT for Smart Cities using the application of clustering, regression, classification etc. In addition, various applications, solutions and data used for implementing the overall framework of Smart Cities are discussed in detail. along with the types of deployment used by these proposed approaches.

2 Big Data Algorithms/Artificial Intelligence

The various sensors that make up the internet of things in a smart city relay information about the city's state to the cloud. However, measuring raw data is not enough, to utilize this data and to make the city 'smart', data analysis is key. Data analysis in smart cities has four layers, the first is Data Acquisition, which deals with the collection and storage of data, this is followed by the Preprocessing layer which performs operations (such as imputing missing values, scaling, removing erroneous data points etc.) on the data to ensure that data is of suitable quality to be used for the data analytics stage. The data analytics stage involves the application of data science techniques on the data to extract patterns and insights which would be used for policy making, planning and other actions in the Service layer. In this section, we focus our attention on the third stage of the data analysis process, i.e., data analytics. Data analytics in the Smart City based on the IoT involves the use of Deep Learning and Machine Learning on the gathered data. The discussion in this chapter considers the following aspects of the use of AI (ML/DL) in the IoT for smart cities:

1. The type of application: This refers to the aim of the application.
2. Algorithm/Network: This refers to the algorithm being used to perform the task and can be any of the ML/DL algorithms covered.
3. System Architecture: System architecture refers to the IoT architecture proposed for the covered work. This can be Cloud, Fog or Edge.
4. Task: This refers to the type of machine learning task being performed. This can be classification, clustering or regression. In this category, a brief description is also presented about what quantities/outcomes are being worked with.

5. Data Type: This refers to the type of data being used. Data can be of two types, heterogeneous or homogeneous. Heterogeneous refers to the use of data of different modalities whereas homogenous refers to a single modality being used. The aim of providing this information is to capture the complexity of the data involved in an application.

2.1 Machine Learning

Machine learning (ML) has been a crucial element of smart city application development [109], helping in prediction (classification), estimation (regression) and clustering tasks. Machine learning refers to the set of approaches through which computers can be used to learn from empirical data [110] and has been used in smart cities in various applications. Since there has been a lot of work in this area using ML algorithms, we focus on work in the last few years. It was found that most commonly used ML algorithms have been the Support Vector Machine (SVM), Random Forests (RF), Decision Tree (DT), Naive Bayes (NB), K-Means, K-Nearest Neighbor (K-NN) and Logistic Regression (LR).

2.2 Deep Learning

Deep learning is the use of successive layers of Artificial Neural Networks (ANNs) to learn patterns. The idea is that successive non-linear layers of interconnected artificial neurons can be used to learn patterns in data that simple machine learning algorithms might not be able to do. Deep learning architectures can process noisy data to provide output for classification and prediction tasks. This makes them very useful in the Smart City environment where the IoT enables the collection of heterogeneous sensor data which can be of varying nature. Data derived from sensors can be processed to extract features or can be fed directly to deep learning algorithms which can perform both feature extraction as well as classification/prediction. Deep Learning methods such as Recurrent Neural Networks (RNN)(Long Short-Term Machines (LSTM) and Gated Recurrent Units (GRU)), Convolutional Neural Networks (CNN), Deep Neural Networks (DNN) and Stacked Autoencoder Networks (SAE) were the preferred deep learning methodologies used for smart city applications and our discussion revolves around the utilization of these methods.

3 AI Applications for Smart Cities

In this section, the applications for of AI in smart cities have been discussed, we also mention the kind of deployment as well as the nature of data utilized to achieve their task.

3.1 Smart Agriculture

The major applications of AI in IoT for agriculture are crop monitoring/disease detection and data driven crop care and decision making. Considering the scarcity of water, the authors in [111, 112, 113] develop irrigation systems which monitor and control the amount of water being used for crops, all structured around a cloud computing system. This problem has been devised both as a classification as well as a regression problem as in [114], who develop a closed loop water irrigation system using support vector regression and K-Means clustering. The authors in [115, 116] propose cloud based greenhouse monitoring systems using images and a host of physical parameters from plants such as temperature, humidity and light using several machine and deep learning methods. Plant disease detection is also an important task within smart agriculture and has been worked on by the authors in [117, 118, 119, 120] who present schemes for disease detection for various crops including tomatoes and potatoes. The proliferation of sensing systems in agricultural fields has also provided an avenue for data driven decision making and planning for farmers. This involves predicting various physical parameters which can affect crop growth like solar radiance [121] and temperature, humidity, windspeed [122, 123, 124, 125, 126] to help in decision making in terms of plant care but also classification systems for recommending crops to be sown [127, 128]. It is important to note that all of these implementations are cloud based.

There have been some suggested methodologies for bringing fog processing for AI in smart agriculture, for, e.g., in [129] a deep learning entrusted to fog nodes (DLEFN) algorithm is described to support efficient use of resources and reduce cloud resource usage. However, as noted in [130], who use an edge system for temperature prediction using an LSTM, edge device performance still lacks that of similar cloud systems but the inclusion of DL capable hardware does provide opportunities for further innovations. Previous work by the same author [131], where they aimed to monitor crops for frost signs and trigger anti-frost measures, compared edge and cloud computing systems for outlier detection and determined that cloud implementations to provide much better performance. However, they do note the potential for edge systems to provide highly responsive data analytics in smart agriculture. More applications can be envisaged for AI deployment in smart agriculture, for, e.g., monitoring of crop growth, selection of the fertilizer and the timeline for it to be used as well as targeted application, pest detection and intelligent pesticide spraying so as to reduce harm to the environment, environmental monitoring to track the effects of climate change and more. Some of these applications have potential to be deployed as edge computing systems. A summary of the use of IoT based AI in Smart Agriculture is presented in Table 7.

3.2 Smart City Services

A popular component of smart city initiatives is the management of waste and involves having bins equipped with sensors and connected to the cloud to not only inform the relevant authorities of the need to empty them but also using AI to determine the best route to reduce fuel consumption. The use of IoT systems for waste management has been observed in the works of [133] who utilize IoT systems to help reduce energy wastage in waste collection by municipalities. Hussain et al. [134] develop a waste management system that not only determines if bins are full and need collecting

Table 7. AI use for Smart Agriculture

Application	Network	System Architecture	Task	Data type
Crop Monitoring /Plant care (Irrigation)	LR [111]	Cloud	Classification - Different states of crop [less water etc])	Heterogeneous (Temperature, Soil moisture, Air quality, Sunlight etc)
	DT [113]			
	CNN [112]		Classification - Different conditions of plants and soil [dry etc])	Homogeneous (Images)
	SVR + K-Means [114]		Regression - Predicting amount of moisture in soil	Heterogeneous (Soil moisture, Soil temperature, Air temperature, Ultraviolet (UV) light radiation, Relative humidity, Weather forecast data)
Crop Monitoring /Plant care (Monitoring and disease detection)	SVM [115]	Cloud	Regression - Forecasting temperature	Heterogeneous (Temperature, Humidity, Light, Soil moisture)
	SVM [118]		Regression - Daily crop growth (indirectly from measured data)	
	SVM [117]		Classification – Different crop conditions	Heterogeneous (Images, Gas)
	SVM + K-Means + CNN [116]		Classification - Different stages of tomato growth	Homogeneous (Images)
	SVM [119]		Classification - Recognizing and detecting disease	
	CNN [120]	Edge		
Data driven crop care and decision making (Predicting physical parameters)	CNN + RNN(GRU) [122]	Cloud	Regression - Prediction of Temperature, Humidity and Wind speed	Heterogeneous (Temperature, Humidity, Wind speed, Location of monitoring station, Time, Rainfall, Solar radiation)
	RFC[126]			
	RNN (LSTM) [123]			
	DNN [125]			
	RNN (GRU) [124]			
	DNN [121]		Regression - Prediction of solar radiance	
	RNN (LSTM) [132]	Edge /Cloud	Regression - Temperature forecasting	
Data driven crop care and decision making (Crop recommendation)	DT [127]	Cloud	Classification - Different crops	Homogeneous (Temperature)
	DT [128]		Classification - Soil fertility and type, Regression - Prediction of soil toxicity	Heterogeneous (Soil moisture, Temperature, Humidity, PH, Soil nutrient content/fertility)

(using data from various sensors placed in the bin) but also predicts the air quality around it using RNNs. The sensing modalities in each of these applications is pretty similar in that they indicate to whether a waste bin is full or not which is then used for route planning. Considering the requirements of such a system, in terms of implementation, all of these systems are cloud based. Sewer monitoring has been performed in [135] in a cloud based system, they use sewer water level and rain gauge data along with a RNN to perform sewer overflow monitoring. The RNN is used to predict sewer overflow ahead in time. Water quality has monitoring has been the focus of [136, 137, 138] where the authors use multiple sensors measuring pH value, chloride, nitrate content and hardness of water to determine whether it is fit for drinking or not. In [139], Liu et al. use data from water monitoring stations along the Yangtze river to predict water quality. Like the classification-based systems, they use multiple chemical measurements from the water such as oxygen, pH, turbidity etc. Apart from air quality, smart city monitoring systems are an important application within the smart city services domain. This includes urban noise, which has been the focus of researchers in [140, 141] as well as other more comprehensive monitoring systems as proposed in [142, 143]. All these systems are cloud based and use a combination of sensors for sound and/or image data for performing noise monitoring/detection and various smart city dashboard applications. In Table 8, we summarize the type of deployments, applications task and data for smart city services applications. It is noted that most of the applications relating to city services such as air quality monitoring and prediction, sewer monitoring, waste collection have been proposed as cloud systems as data needs to be collected from nodes at various points in a city. It is envisaged that due to the nature of the applications, many smart city services would still rely on cloud or fog architectures as the decision-making taking place in such situations isn't possible on only a local level. It is also observed that most of the applications required data from multiple sensors and therefore utilized heterogeneous data to carry out the task at hand.

3.3 Smart Energy

Load/energy consumption forecasting is an essential task for monitoring and control of electrical power supply in the electricity grid and ensure appropriate demand side management. It has been performed by the authors of [148] who use data collected from consumers in a smart grid to determine load for up to 24 h in advance. They treat this as primarily a clustering problem where they form clusters of similar load profiles and then use distance functions to determine energy consumption for the future. The authors in [149] also use a cloud based clustering approach, using historical power data, they use K-Means clustering to determine the closest historical records and then combine them to predict energy consumption 24 h in advance. The load forecasting problem has been dealt as a regression by [150] using a SVM and by [151] through an RNN using electricity power data. A regression approach is also followed by [152, 153] who use electricity consumption in addition to environmental data for load forecasting using deep learning methods (DNN and a combination of Autoencoders and RNNs (GRU)). Edge based systems have been suggested by the authors of [154, 155, 156]

Table 8. AI use for Smart city Services

Application	Network	System Architecture	Task	Data type
Air quality	K-NN [144]	Cloud	Classification - Differentiate between different air quality levels	Heterogenous (Gas, Light, Temperature, Humidity, Pressure,
	RFC[145]			Wind speed, Weather information, Images,
	RFC[146]			Traffic flow data, Visibility, information about
	RNN (LSTM) [147]		Regression - Prediction of air quality levels	types of buildings etc)
Water quality monitoring	NB [136]	Cloud	Classification - Determine if water is fit to drink or not	Heterogeneous (Chlorides, Nitrates, Total dissolved solids,
	SVM [137]			pH and Hardness, and other chemical properties)
	DNN [138]			
	RNN (LSTM) [139]		Regression - Prediction of water quality	
Sewer Overflow Monitoring	RNN (GRU,LSTM) [135]	Cloud	Regression - Prediction of when	Heterogeneous (Water level sensor data (ultrasonic) over drain holes as well as rain gauges)
Waste management	RNN (LSTM) for prediction of air quality [134] , K-NN for detection of waste bin being full	Cloud	Regression - Prediction of air pollutant levels, Classification - Bin full or not	Heterogeneous (Odor, Weight, Level sensing using
	RFC[133]		Classification - Empty bin or not	ultrasonic sensor, Gas sensor for air quality, Vibration)
Urban noise monitoring	CNN [140]	Cloud	Classification - Different types of sounds	Homogeneous (Sound)
	RNN (LSTM) [141]		Regression – Prediction of noise levels	
Management of Smart City	CNN [143]	Cloud	Application – Dashboard (object identification etc)	Heterogeneous (Various sensors, Urban video and sound data)
	CNN [142]			

for load forecasting for household consumers, [156] use federated learning to train a RNN. In addition to load forecasting, smart grid management/monitoring is also a necessary application in this domain. The authors in [157] use decision trees in a cloud based system to classify between different faults in a smart grid. In [158], the authors use power consumption data from consumers in China to determine electricity theft in a cloud based system. They use wide and deep convolutional neural networks to capture the periodic and nonperiodic components from electricity consumption data and show their network to be suitable for electricity theft detection. The authors in [159] present a framework for edge computing based monitoring of the smart grid. Edge computing in the smart grid has several advantages as it reduces delay and also it is secure in terms of data privacy. A summary of the use of IoT based AI in Smart Energy is provided in Table 9.

Table 9. AI use for Smart Energy

Application	Network	System Architecture	Task	Data type
Energy/Load consumption forecasting	K-Means [148]	Cloud	Clustering - Determine clusters of similar power consumption	Homogeneous (Electric power)
	K-NN [149]		Regression - Predict consumption of electricity ahead of time	
	SVM [150]			
	RNN (LSTM) [151]			
	DNN [152]			Heterogeneous (Electric power, Temperature, Humidity, Time, Holiday
	SAE + RNN (GRU) [153]			
	CNN [155]	Edge		
	RNN (GRU) [154]			Homogeneous (Electric power)
	RNN (LSTM) [156]			
Smart Grid line event classification (fault etc.)	DT [157]	Cloud	Classification - Different powerline events	Homogeneous (Electric power)
Electricity theft detection	CNN [158]	Cloud	Classification - Theft detection for abnormal patterns of consumption	Homogeneous (Electric power)

3.4 Smart Health

There are two major applications of IoT with AI in health, these are activity recognition/fall detection and disease diagnosis/health monitoring, a summary of the IoT based AI systems used in Smart Health is presented in Table 10. Activity recognition involves the use of movement sensors such as accelerometers, gyroscopes and magnetometers with the aim to help provide the user with feedback on their health. This can be in terms of them having enough physical exercise or not, used for sports therapy, fall detection and for monitoring of different diseases such as Parkinson's or other motor degenerative ailments. The most popular sensor for activity recognition are inertial sensors which have been used by [160, 161] in a cloud based setting using various deep and machine learning algorithms. In [162], Castro et al. include vital sign data in addition to movement information for human activity recognition in a cloud environment, they utilize the DT as their classifier. In [163],, the authors propose an edge-based system to perform activity recognition for people by recording their movements using the accelerometer and gyroscope present on the phone. They use a SVM as their classifier and differentiate between six different activities of daily living. Fall detection has been performed by [164, 165] in a fog and edge environment, respectively, using an accelerometer, Santos et al. [164] use a CNN while Yacchirema et al. [165] useRFCwith both approaches showing promising results.

Equipped with the power of AI in IoT, Smart Health systems facilitate the provision of telehealth services as well as real time monitoring of patients, giving doctors and patients feedback on their health. Health monitoring systems and real time disease diagnosis have been one of the most important applications of IoT technology. The authors in [166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176] develop cloud based health monitoring systems for detecting various types of diseases, such as heart (stroke [167], irregular sound [169], irregular rhythm [171, 173]), epileptic seizures [172], Parkinson seizure [176] and multiple disease diagnosis systems [174, 175]. In [174, 175], the authors formulate the problem of disease diagnosis as a classification problem and utilize medical data such as ECGs, EEG, heart rate, blood pressure, blood sugar, heart sound, blood glucose, liver health along with various machine and deep learning methods to achieve this task. Fog and edge based health monitoring/disease diagnosis systems have also been suggested by a number of researchers. The authors in [177] present a fog based system using a deep neural network to detect heart disease from a patients vital signs (blood oxygen, heart rate, respiration rate, EEG, ECG, EMG, blood pressure, glucose) and activity data. In [178], Devarajan & Ravi work on a fog computing based Parkinson detection system using a persons speech. Moreover, an edge computing system is presented in [179] which utilizes EEG signals to determine seizures in patients.

Table 10. AI use for Smart Health

Application	Network	System Architecture	Task	Data type
Human activity recognition/Fall detection	DT [162]	Cloud	Classification - Different activities, fall /non falls	Heterogeneous (Acceleration, Heart rate, Posture, ECG, Respiration rate)
	RFC[160]			Homogeneous (Accelerometer)
	CNN [161]			
	RNN (LSTM) [180]	Fog Edge		Heterogeneous (Accelerometer, Gyroscope, Magnetometer)
	CNN [164]	Fog		
	RFC[165]	Edge		Heterogeneous (Accelerometer and Gyroscope)
	SVM [163]			
Patient health monitoring	DT [166]	Cloud	Classification - Recommendation about diet etc	Heterogeneous (Heart rate, Sleep, Calories burned, Weight, Physical activity time, Water, Exercise etc)
	SVM [169]		Classification - Different emotions	Heterogeneous (Speech and Image)
	RNN(LSTM) [181]			Heterogeneous (ECG, BVP, GSR, SKT, EMG)
	CNN + SAE [172]		Classification - Abnormal/normal heart sounds	Homogeneous (EEG)
	RFC[168]		Classification - Epileptic Seizure detection	Homogeneous (Heart sounds)
	SVM [171]		Classification - ECG arrhythmias	Homogeneous (ECG)
Disease diagnosis	DT [173]	Cloud	Classification - Different heart diseases	Heterogeneous (Heart health information, Patient records and other health sensors)
	K-Means [174]		Classification - Kidney, Heart and Liver disease	Heterogeneous (Heart and Kidney health data)
	RFC[175]		Classification - Detection of various diseases	Heterogeneous (Diabetes, Heart, Liver, Dermatology etc data)
	DNN [177]	Fog	Classification - Presence of heart disease or not	Heterogeneous (Blood oxygen, Heart rate, Respiration rate, EEG, ECG, EMG, Blood Pressure, Glucose and Activity data)
Parkinson detection/	RFC[167]	Cloud	Classification - Parkinson detection/stroke has happened/seizure detection	Heterogeneous (Blood pressure, Sugar, Pulse rate)
Seizure monitoring	SVM [176]			Homogeneous (Speech)
	K-NN [178]	Fog		
	NB [179]	Edge		Homogeneous (EEG)

3.5 Smart Homes

Ambient assisted living is a huge component of Smart Homes. This is especially needed for the elderly and is typically achieved by the use of ambient sensors, Wi-Fi and radio frequency systems in smart homes. In this work, we include all monitoring methods that depend on sensors placed in the home/within the smart home domain. In [182], Pirzada et al. use a network of reed switches connected to the cloud to monitor the activities of elderly people as a clustering problem. They use the K-NN algorithm to determine anomalies in the daily activities which can then be used to send medical or other help requests to assist people. A similar setup for activity recognition for ambient assisted living has been presented in [183, 184, 185] where they use data from a number of different sensors including motion, presence, water float, temperature etc to determine various activities being performed in a home. In [186], a cloud based assisted living system for the deaf has been developed that performs haptic conversions for sounds detected in a home. An array of sensors are used to monitor environmental sound and the authors use RNNs for detecting the sound event before its passed on to the haptic vibration producer. Another task within in monitoring is localization of people, this part of smart homes is also applicable to smart infrastructure in that such systems are used in smart buildings as well. Applications of localization include security, i.e., detecting unauthorized presence and people monitoring in general (for, e.g., locating elderly people in homes) etc. The authors in [187] perform localization using a grid of Wi-Fi units that measure signal strength to determine peoples locations indoors for buildings. They formulate this both as a classification problem as well as a regression problem. The classification problem being formulated as coded locations (for, e.g., a given room no) while the regression case estimating the location of the user in a coordinate grid. Their system is cloud based and they use a deep neural network to perform this task and have found suitably good results. Occupancy detection has been performed by the authors of [188] making use of various ambient measurements (temperature, humidity, pressure etc.) and passing them to a cloud before using a deep neural network to classify between the various number of people present in an indoor environment. In [189], Zimmermann et al. also make use of ambient sensors for the occupancy problem and use a naïve bayes learner to determine both the presence of occupants as well as their number. Home automation is another application that the IoT finds application within the Smart Home domain. The integration of AI has helped develop smart home automation systems that aim to reduce energy consumption in homes as well as maintain user privacy, security. Chowdhry et al. [190] use a combination of visual data and motion sensing to perform home automation for security using a SVM. An interesting use of AI in home automation is presented in [191] who develop a cloud based home automation system, they take measurements from various ambient sensors and control appliances and use a Naïve Bayes classifier to determine which technician to call whenever sensor measurements appear aberrated. The problem of intelligent consumption of energy has been considered by [192] who develop an energy disaggregation system on the appliance level in smart homes using stacked denoising autoencoders. They achieve this using power data for individual appliances as well as the total power consumed in the home and send it to a local cloud. Data is then disaggregated for various appliances to provide feedback to the user. More work providing energy intelligence to consumers has been performed by [193, 194, 195]. Konstantakopoulos et al. [195] pose this as a regression problem, they propose a cloud-based system utilizing both ambient sensor data (lighting, temperature etc) and appliance power data to forecast resource usage for consumers using a RNN. They show a reduction in energy consumption for their users using this information.

Eventhough applications covered in smart homes have been cloud based systems, there have been recent proposals for frameworks that combine edge and cloud processing as in [196] who discuss a hierarchical control system for smart homes through a edge microgrid and a cloud power grid. Due to the nature of smart homes in that the sensing scheme is present within a finite space (within the home), edge and fog computing based systems are expected to be increasingly incorporated smart home applications. A summary of the use of IoT based AI in Smart Homes is presented in Table 11.

3.6 Smart Industry

One of the major applications of AI in the IoT powered smart industry is towards fault detection in products and anomaly detection in industrial processes. This has seen the use of both Machine Learning (SVM [197, 198],RFC[199, 200]) as well as Deep Learning (DNN [201, 202, 203], CNN [204]) methods using a cloud computing structure to perform anomaly detection/product inspection and monitoring using a variety of heterogeneous and homogenous data sources such as inertial sensors for machines, images for products and processes and other process specific variables. Other approaches suggested in [205, 206] propose a fog computing method along with edge computing systems suggested in [207, 208]. An edge computing system for anomaly detection is presented in [209] where edge devices collaboratively train a deep anomaly detection model. Production management is another application that has found usage of AI in IoT based smart Industry. For, e.g., the authors in [210, 211, 212, 213, 214] use cloud based data driven systems along with machine and deep learning algorithms to help with task dispatching, performance analysis as well as worker activity recognition utilizing a variety of sensing modalities. The work of [213, 214] are especially interesting as they aim is to not only perform production management but also propose data for various health related analysis to create a safer working environment on the factory floor. A fog system for production management has been presented in [215] who use activity data to determine resource allocation locations to contribute to management of a production operation. Furthermore, product inspection, which is a common application of instrumentation systems in a factory, has been performed by [216, 217] who utilize images and sensor data in a cloud based system to monitor product quality.

A factory has a multitude of machines and equipment working round the clock manufacturing goods. Maintenance is an important aspect of this operation where regular checks are performed on the equipment to ensure that no breakdown occurs during the production process, which might result in monetary loss or loss of life.

Table 11. AI use for Smart Homes

Application	Network	System Architecture	Task	Data type
Ambient Assisted living (Activity recognition/Fall detection)	K-NN [182]	Cloud	Clustering - Detect abnormal clusters	Homogeneous (Reed switches)
	RNN (LSTM) [185]		Classification - Different activities	Heterogeneous (Human motion, Water float, Reed switches, Temperature, Pressure, Luminance, Gas and other environmental sensors in a home)
	RNN (LSTM) [183]			
	SAE [184]			
	RNN (GRU) [186]		Classification - Different sounds	Homogeneous (Sound recordings from rooms in a house)
Ambient Assisted living (Localization and Occupancy detection)	DNN [187]	Cloud	Classification and Regression - Localization estimation	Homogeneous (Wi-Fi signal strength and identifiers)
	NB [189]		Classification - Presence of people or not, Regression-Number of occupants	Heterogeneous (Volatile organic compounds, CO, Temperature, Humidity)
	DNN [188]		Classification - Different number of people present	Heterogeneous (Temperature, Luminance, Humidity, Pressure, CO2, Motion, Magnetometer, Gyroscope, Accelerometer, Sound, Door and window open/close status)
Energy management (Automation, Power consumption profiling)	SVM [190]	Cloud	Classification - Intrusion detection	Heterogeneous (Images + Sound)
	SAE [192]		Regression - Disaggregation of appliance power data	Homogeneous (Appliance power consumption)
	RNN (LSTM)[195]		Regression – Forecasting occupant resource usage	Heterogeneous (Appliance power consumption, Luminance, Vibration, Temperature, Humidity, Accelerometer [fan])
	SAE for disaggregation and RNN(LSTM) for forecasting [194]		Classification - Energy disaggregation, Regression - Load forecasting	Heterogeneous (Temperature, Luminance, Humidity, Proximity switches, Ultraviolet light sensors, Power consumption)
	NB [193]		Classification - Determine appliances that are on	Homogeneous (Appliance power consumption)

However, with the data gathered by various sensors on these machines, it is often more beneficial to take an active approach rather than a passive one by using this data for predictive maintenance purposes. Predictive maintenance utilizes data from the daily operation of machines in an industry to optimize the manufacturing operation [218] and is one of the main uses for AI in the industry. In [219, 220], the authors suggest a predictive maintenance scheme using SVMs utilizing data from accelerometers measuring vibration in a crane motor and data from various sources in a semiconductor manufacturing process, respectively, both work in a cloud environment as evidenced from the architecture. Prediction of failure can also be a regression operation, as was demonstrated by [221] who use RNNs to predict future values of a physical parameters of a pump using a number of heterogeneous sensors used to monitor it. As with the other two systems, this system also had a cloud architecture. The authors in [170] also present a regression based health prognosis system for the industry using a CNN on machine data (Images, stress, temperature, vibration, position and electromagnetic signal measurements). The use of IoT based AI in Smart Industry has been presented in Table 12.

3.7 Smart Infrastructure

An application within smart infrastructures also involves monitoring of civil structures for structural health. The authors in [222] take a clustering approach to perform health monitoring of a bridge using vibration data in a cloud setting. They use clustering to determine clusters of abnormal behavior in accelerometer measurements from a bridge. In [223, 224, 225], accelerometer signals have been used where as [226] have used piezo electric transducers for performing structural health monitoring of bridges formulated as a classification problem between different damaged states of a bridge. The prime sensing modality for monitoring has been measuring vibration using accelerometers, however, other sensors such as fiber optic gratings can also be used to measure stress.

The second avenue for AI applications in IoT for smart infrastructure is the use of IoT devices for building environment control and energy management as well comfort aware control. This involves the prediction of building energy usage based on environmental data (such as temperature, humidity) and electrical power data. The authors in [227] and [228] use electric power data and heat flow information in a building to predict the energy requirements in the future so as to better manage energy consumption. Ambience control of a museum has been performed in [229] where the authors use deep learning algorithms to predict the CO2 and humidity levels for the care of exhibits. Comfort aware energy management has been performed in [230] where the authors use a CNN to regulate thermal comfort in a building using various physical quantities. It can be noted that all of these mentioned systems have been deployed in the cloud, this is due to the nature of the application. However, there have been efforts for developing fog/edge systems for smart infrastructure. The authors in [231] describe a framework for deploying edge and fog computing services in smart buildings and demonstrate their systems effectiveness for the case of energy management. Table 13 summarizes the use of IoT based AI in Smart Infrastructure.

3.8 Smart Transport

Major smart transportation applications involve smart parking and transportation management. Smart parking aims to solve the problem of helping users finding parking spots in order to save time as well as reduce gas emissions and is therefore a much-researched topic for AI deployment towards smart transportation. Solutions to this problem have been formulated both as a regression problem as well as a classification one, both utilizing imaging and/or other occupancy sensing modalities. Regression solutions [232, 233, 234] are typically used to predict a parking lots occupancy levels in the future whereas classification systems [235, 236, 237] involve guiding drivers according to the shortest distance as well as used for user localization purposes within such lots. In addition to cloud based approaches, edge computing systems for smart parking have also been devised as suggested in [238, 239] who deploy CNNs on edge

Table 12. AI use for Smart Industry

Application	Network	System Architecture	Task	Data type
Fault and anomaly detection	DNN [203]	Cloud	Classification - Different classes of abnormality labels	Heterogeneous (Multiple sensor and controls [button states etc] information)
	DNN [202]		Classification - Different damage stages of a 3D printer	Heterogeneous (Accelerometer, Gyroscope)
	RFC[199]		Classification - Normal and abnormal operation in wind turbines	Homogeneous (Accelerometer)
	SVM [197]		Classification - Different wind turbine health conditions	
	SVM [198]		Classification - Normal and mixed cement	Homogeneous (Images)
	RFC[200]		Classification - Different fault types in steel manufacturing	Heterogeneous (Various sensors, dimensional measurements)
	DNN [201]		Classification - Normal and arcing	Homogeneous (Current)
	CNN [204]		Classification - Defected product or not	Homogeneous (Images)
	CNN [205]	Fog	Classification - Different types of defects	Homogeneous (Images)
	CNN [206]			
	SVM [207]	Edge	Classification - Abnormal and normal pressure	Homogeneous (Water pressure)
	CNN + LSTM [209]		Classification - Abnormal and normal time power patterns	Homogeneous (Electrical power)
	RNN (LSTM) [208]		Classification - Faulty and normal state of a machine	Homogeneous (Accelerometer)
Production management	SVM [210]	Cloud	Regression - Prediction of the slotted coefficient in a hydraulic press	Heterogeneous (Various measurements from a hydraulic press)
	ConvLSTM + SAE [211]		Regression - Forecasting machine speed to make production more efficient	Homogeneous (Speed of machine [rotary])
	DNN [214]		Regression - Bottle neck prediction in time	Heterogeneous (RFID, movement sensors)
	CNN [212]		Classification - Different activities in an assembling factory	Heterogeneous (IMU, EMG)
	SVM [213]		Classification - Different activities in a meat processing plant evaluate worker performance	Heterogeneous (Accelerometer, Gyroscope)
	RFC[217]		Classification - Bad or good product quality	Heterogeneous (Various sensors from a production floor in a factory)
	CNN [216]		Classification - Prediction of temperature, Carbon content in steel	Homogeneous (Spectrogram Images)
	RFC[215]	Fog	Classification - Determine Room ID, used for system disruption	Heterogeneous (Activity data, Location)
Predictive maintenance	CNN [170]	Cloud	Regression - Predict health index for machines	Heterogeneous (Images, Temperature, Vibration, Position, Electromagnetic signal measurements, Strain gauge)
	SVM [219]		Classification - Abnormal or normal vibration data (from electric motor in a crane)	Homogeneous (Accelerometer)
	RFC+ SVM [220]		Classification - Failure prediction	Heterogeneous (Multiple sensors from SECOM dataset)
	RNN (LSTM) [221]		Regression - Predicting data from sensors	Heterogeneous (Different sensors [Pressure, Temperature, Vibration etc])

Table 13. AI use for Smart Infrastructure

Application	Network	System Architecture	Task	Data type
Structural health monitoring	K-Means [222]	Cloud	Clustering – Look for abnormality of building state	Homogeneous (Accelerometers)
	K-NN [226]		Classification – Different damage states	Homogeneous (Piezo electric sensors)
	DNN [223]			Homogeneous (Accelerometer)
	CNN + RNN (LSTM) [225]			
	SVM [224]			
Energy and Environment management	SVM [227]	Cloud	Regression - Forecasting electrical power usage	Heterogeneous (Power and environmental data)
	SAE [228]		Regression -Energy predictions for buildings	Homogeneous (Heat flow data in buildings)
	RNN (GRU, LSTM) [229]		Regression -Prediction of environmental variables (CO2, Humidity etc)	Heterogeneous (environmental data such as CO2, Humidity, Air velocity)
	CNN [230]		Regression - Comfort level	

devices for occupancy detection and user localization, respectively. Another application of AI IoT for smart transportation involves determining traffic flow as well as prediction of traffic flow for traffic light control and other management tasks such as accident detection. In this regard, video cameras are popular for detection of vehicle density on roads for traffic congestion determination. However, with most cars having a GPS device and the commonality of cellphones with every driver, many approaches use the data from the GPS along with weather and generic traffic flow information to determine traffic prediction. The nature of traffic flow prediction using sensing modalities such as GPS require systems to be operated as cloud-based systems as is the case in [240, 241, 242, 243]. Of these, Wangyang et al. [242] and Xiao et al. [243] use deep learning based sequential modeling approaches to predict traffic flow ahead of time where as Aung & Naing [240] and Yunxiang Liu & Wu [241] solve this through a classification formulation. A traffic management system for public buses has been proposed in [33, 244] where GPS data is used to predict bus arrival times for public transportation systems. Accident detection has been performed using car position and velocity information in a Vehicular Adhoc NETworks (VANET) environment by the authors of [245]. They do this through a cloud based system that can use this information to predict whether an accident has occurred or not. Apart from typical sensing modalities, with smart phones and user participation in social media, smart transportation systems are expected to increasingly include more sensing modalities [246] fused together for use in decision making for traffic management purposes. Mukherji et al. [247] use Wi-Fi signals to determine commuter traffic a subway station. They do this by using the measured signal strength of the Wi-Fi signals along with a Random Forest classifier in a cloud-based setting. Their system is able to determine if a person is on the train or on the platform which can be used to help with planning train times and routes. A summary of IoT based AI for Smart Transport has been given in Table 14.

Table 14. AI use for Smart Transport

Application	Network	System Architecture	Task	Data type
Smart Parking (Parking occupancy detection/Routing/Location prediction)	K-NN [237]	Cloud	Classification - Presence of a vehicle	Homogeneous (Images)
	K-Means [232]		Regression - Future occupancy prediction	Heterogeneous (Occupancy, Location, Time)
	RNN (LSTM) [234]			
	LR [233]			Homogeneous (RFID data from cars)
	DNN+ CNN [235]		Classification - Different positions based on beacons installed	Homogeneous (Radio frequency signal strength)
	DT [236]		Classification - Recommendation of parking lot based on distance	Heterogeneous (Parking information, Time)
	CNN [238]	Edge	Classification - Detection of empty parking space	Heterogeneous (LIDAR, Images)
	CNN [239]		Classification – Different user locations localization	Homogeneous (Bluetooth received signal strength)
Transport management (Public transport management)	K-Means [33]	Cloud	Regression – Transport delay prediction	Heterogeneous (GPS, Ticket information, Time, Arrival, Departure information etc)
	K-Means [244]		Regression – Arrival time prediction	
	RFC[247]		Classification – Localization, as on platform or train	Homogeneous (Wi-Fi signal parameters)
Transport management (Traffic flow)	NB [240]	Cloud	Classification - Different traffic states	Homogeneous (GPS data, current and historical)
	RFC[241]			Heterogeneous (Weather, Road data)
	RNN (LSTM) [243]		Regression - Traffic flow prediction	Homogeneous (Traffic flow data[vehicle speed count etc])
	RNN (LSTM) [248]			
	SAE + RNN (LSTM) [242]			
Transport management (Traffic Accident detection)	RFC [245]		Classification - Accident or not	Homogeneous (Velocity, Position)

4 Conclusion

This chapter provided a coverage of the usage of AI in terms of machine and deep learning for applications within smart cities. For each of the applications discussed for the various components, the type of deployment based on the technologies and architectures discussed in the previous chapter were identified. Figure 7 provides a summary of the AI discussion in this section. Moreover, a tabular version is provided in Table 15 for this section. It highlights the applications in which each AI domain finds use in for smart city components.

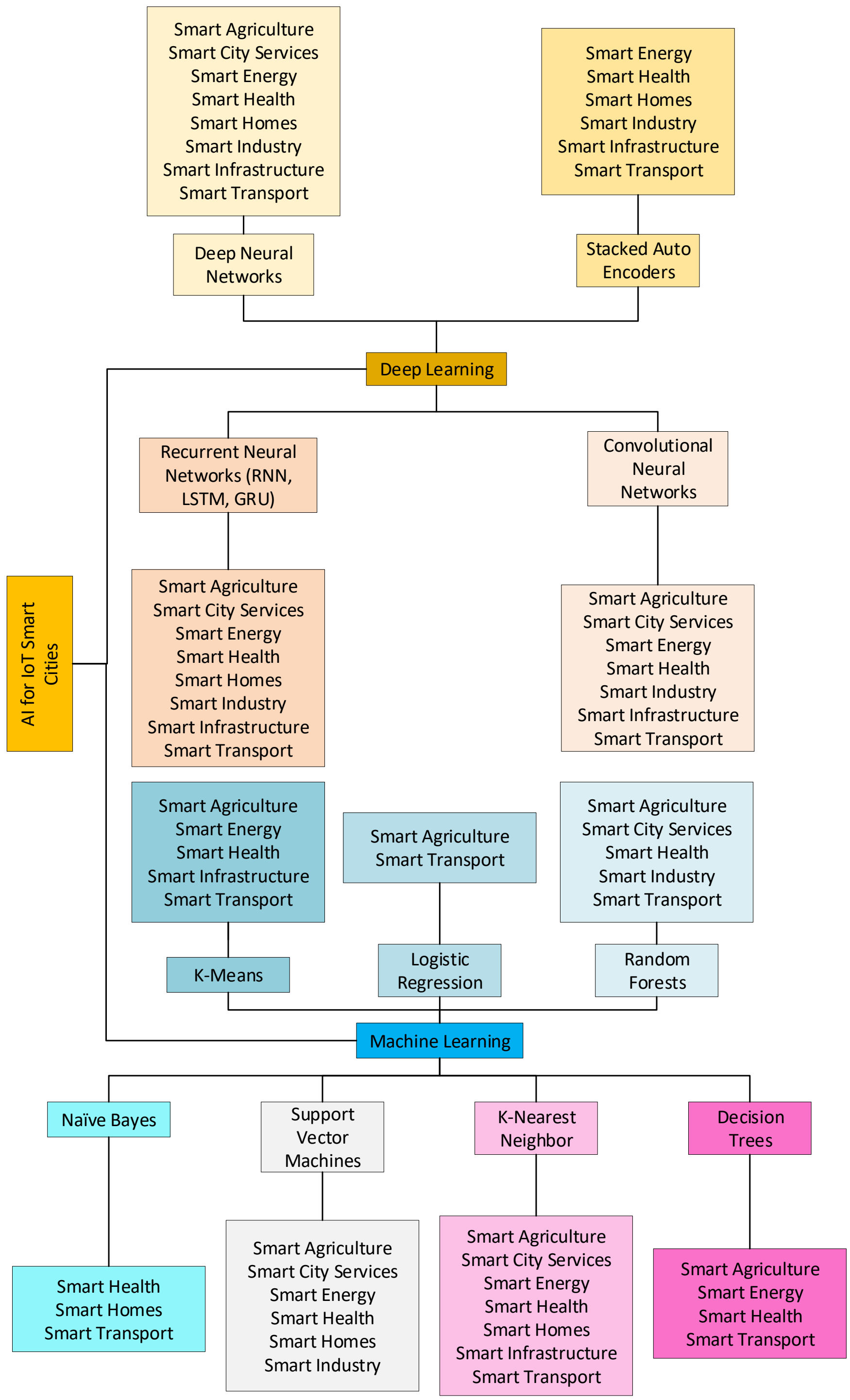

Figure 7. AI for IoT Smart Cities.

Table 15. AI applications for IoT Smart Cities

Smart City Component	Machine Learning	Deep Learning	Observations
Smart Agriculture	- Crop Monitoring /Plant care (Irrigation) - Crop Monitoring /Plant care (Monitoring and disease detection) - Data driven crop care and decision making (Predicting physical parameters) - Data driven crop care and decision making (Crop recommendation)	- Crop Monitoring /Plant care (Irrigation) - Crop Monitoring /Plant care (Monitoring and disease detection) - Data driven crop care and decision making (Predicting physical parameters)	
Smart City Services	- Air quality - Water quality monitoring - Waste management	- Air quality - Water quality monitoring - Waste management - Sewer Overflow Monitoring - Urban noise monitoring	For applications such as Smart Agriculture, Smart Energy, Smart Health, Smart Industry and Smart Transport, Deep Learning as well as Machine Learning algorithms have been deployed in Edge/Fog configurations.
Smart Energy	- Energy/Load consumption forecasting - Smart Grid line event classification	- Energy/Load consumption forecasting - Electricity theft detection	
Smart Health	- Human activity recognition/-Fall detection - Patient Health Monitoring - Disease diagnosis - Parkinson detection/Seizure monitoring	- Human activity recognition/-Fall detection - Patient Health Monitoring - Disease diagnosis - Parkinson detection/Seizure monitoring	
Smart Homes	- Ambient Assisted living (Activity recognition/Fall detection) - Ambient Assisted living (Localization and Occupancy detection) - Energy management (Automation, Power consumption profiling)	- Ambient Assisted living (Activity recognition/Fall detection) - Ambient Assisted living (Localization and Occupancy detection) - Energy management (Automation, Power consumption profiling)	
Smart Industry	- Fault and anomaly detection - Production management	- Fault and anomaly detection - Production management	The most popular machine learning algorithms were the SVM and RFC.
Smart Infrastructure	- Structural health monitoring - Energy and Environment management	- Structural health monitoring - Energy and Environment management	
Smart Transport	- Smart Parking (Parking occupancy detection/Routing/Location prediction) - Transport management (Public transport management) - Transport management (Traffic flow) - Transport management (Traffic Accident detection)	- Smart Parking (Parking occupancy detection/Routing/Location prediction) - Transport management (Traffic flow)	The most popular Deep Learning algorithms were RNNs and CNNs.

CHAPTER IV

OPTIMIZATION IN IOT SMART CITIES

1 Introduction

This chapter presents a coverage of combinatorial optimization in IoT based smart cities by deliberating on the most popular applications of optimization algorithms and providing an insight to how those problems are formulated and worked upon. It provides a mapped overview of the optimization landscape in the smart city domain while considering the IoT. Through this mapping, the common optimization problems across the various components of the IoT enabled smart city have been identified. For each problem discussed, the objective function used, the nature of the objective as well as the constraints considered have also been elaborated on.

2 Optimization and Heuristics in IoT Smart Cities

As has been observed by [249], combinatorial optimization problems in the real-world are known to be difficult to formulate and generally are hard to solve. Moreover, choosing the right algorithm is also a tedious task as each comes with a different set of characterizations. This is very applicable to the IoT bases smart city where applications in each component caters to a different environment and aspect of the city's operation and therefore requires intricate design of the problem. This chapter takes guidance from the work of [249]. They note that the most popular algorithms for use in combinatorial optimization problems are the Ant Colony Optimization (ACO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE) and Artificial Bee Colony (ABC). Moreover, following from their discussion, six factors are considered for each application identified. These are:

1. The type of application: This refers to the problem being optimized within the smart city domain.

2. Data Setup: For each application, we mention the data setup used. While doing so, we aim to capture the various ways in which researchers have sourced data for their proposed method.

3. Single-Parallel problems: Another thing to note in smart city optimization problems is whether a problem has been considered as a single problem or multiple sub-problems/parallel.

4. Objective direction, function and number of objectives: Maximization or minimization, lowest fitness function value desirable or higher fitness function value

is desirable. Since we list the objectives, we also covere the number of objectives inherently. Single objective, where a single fitness is optimized for its best value or multi-objective where multiple objective functions need to be considered at the same time. The latter is a complex process as some objectives may have conflicts and thus requires the need to perform trade-offs with what's acceptable.

5. Constraints: Constraints are a set of restrictions or prerequisites that may sometimes be necessary to determine if a solution is considered valid or not. Soft constraints are desirable but not necessary whereas hard constraints are mandatory to be met. Constraints are put on the fitness function according to application being considered. Covering this aspect is particularly important as constraints are determined by the application being worked on.

3 Meheuristic Algorithms

Metaheuristic algorithms are widely used to solve combinatorial optimziation problems in the real-world [249]. The aim of these algorithms is to determine the minima or maxima of an objective function which translates an optimization objective in to one or more mathematical equations. Five algorithms have been considered in this chapter, these are the Ant Colony Algorithm, Genetic Algorithm, Particle Swarm Optimization Algorithm, Differential Evolution and Artificial Bee Colony. As mentioned earlier, these have been chosen as these were the most commonly used optimzation algorithms identified by [249]. In this section, we provide a brief intuitive description for each of them.

3.1 Ant Colony Optimization

Ant colony optimization is derived from the behavior of ants searching for food [250]. It was observed that during this process, each ant deposits a chemical called pheromone on the path which it takes towards the food. The more the ants take a given path, the more the pheromone is deposited on it. Subsequent ants that want to go towards the food use the amount of pheromone on a given path or sub-path to decide which path to take so as to determine the optimal route to the food. In the artificial ant colony optimization algorithm, the points on the path that form the sub-paths are encoded on a graph where each ant can only visit a given vertex (travel on the same sub-path) once. Each iteration starts with a number of artificial ants, an ant would choose the next vertex to visit based on the level of pheromone on the possible sub-paths available to it as well as whether that path has been used before. At the end of each iteration, the pheromone levels are updated so as to prioritize the use of the most used path for ants in the next iteration.

3.2 Genetic Algorithm

A genetic algorithm [251] is based on evolutionary science. The idea behind the genetic algorithm is that starting from a given population set of organisms, subsequent reproduction will result in fitter organisms being produced given that the organisms chosen for reproduction have some level of fitness. To solve an optimization problem, a genetic algorithm begins with a given population size of strings also called a chromosome. Each 'chromosome' consists of a 'gene' which may represent points in the population. The sequence in which the genes are present would represent a solution to the problem. For e.g. for a route scheduling problem, this may be the coordinates of the stops. The 'goodness' of a chromosome is determined by a fitness function that quantifies how appropriate a given chromosome is as a solution for the problem. Choosing the right fitness function is an important consideration in genetic algorithms as it needs to gauge the kind of optimization that is to be performed. The optimization process starts with an initial population of a given size. Once a fitness function has been defined, in each iteration, two or more chromosomes (parents) are taken at random from the population and one or more offsprings are generated. The random selection takes the selected parents fitness function value in to account, however, its necessary that not all parents chosen are the fittest of the population as otherwise, diversity will be compromised, and the algorithm may get stuck in a local minimum. The method by which these offspring are generated is called the Crossover function and the number of parents mated to form offsprings from them depends on the crossover rate. The Crossover function defines the way the genes within the chromosomes are exchanged. Usually, the crossover rate has a high value. Moreover, depending on some mutation rate, a given fraction of all offsprings are mutated. Mutation depends on the mutation function used and involves members of the offspring being swapped in some manner. When one iteration of the offspring creation from the parents in the entire population is completed, the offsprings replace members of the original population (typically the weakest) and one generation of the GA is said to be completed. In order to converge to a sufficiently good result, GA's need to run for many generations.

3.3 Particle Swarm Optimization

Particle Swarm Optimization is modeled on the social behavior of animals rather than the evolutionary biological behavior on which Genetic Algorithms are based. PSO [252] is particularly based on the behavior of a flock of birds searching for food. Each bird in the flock searches for food and can communicate with other birds in the flock as soon as it finds food or a good direction for it. Therefore, each bird has two food 'directions' that it needs to consider, first is on an individual level which is called the personal best and the second is on the flock level which is the global best. Using these two values the bird will proceed towards that path and will inform other birds in the flock too. The idea here is that after enough time and with all the individuals working together, the swarm will find the place with the highest concentration of food. In terms of using PSO for optimization tasks, individual birds are equivalent to a solution as in a GA and each is considered as a point or a particle, collectively they make up the swarm. This swarm population may be initialized randomly or based on some domain knowledge about the problem. The highest concentration of food represents the optimal solution for the whole swarm where as a good direction represents the local optimal solution for each case. The aim here, like birds in a flock is to determine the global optimal solution using information from each individual particle. Each particle has three aspects to it, its position, its velocity and a value of its current position's 'goodness'. This 'goodness' as in the GA is defined by a fitness function. Like birds, each particle has a personal best fitness value and is also aware of the global best fitness value. For any particle, the new direction of movement is decided by a combination of the personal best and the global best as well as the particle's intention to maintain its current movement. Furthermore, several different topologies can be utilized, and a neighborhood can also be set for each particle to convey positions when limiting the global best parameter to the local best scheme rather than the whole swarm. The algorithm may be stopped till the best solution is reached or no more improvement is observed.

3.4 Differential Evolution

Differential evolution [253] is a stochastic evolutionary algorithm which is used for optimization problems for continuous functions. The DE algorithm does not expect gradient information and thus it doesn't need to be differentiable. Like other evolutionary algorithms, a solution is searched for in the design space by maintaining a set of individual candidate solutions. In each iteration, the solutions with the best fitness are combined together and retained for the next iteration. The aim is to improve upon the fitness value and this process is repeated until a pre-decided condition for termination of this process is satisfied. Differential Evolution is very similar to Genetic Algorithms in that both are evolutionary in nature, however, the difference is that the DE uses real numbers in the chromosome and also that the mutation operation consists of the difference between two or more chromosomes of the population with the possible addition of some type of noise to create a new individual. Like GA, DE also suffer from getting stuck in local minima. The DE algorithm also has three control parameters, the population size, the mutation factor and the crossover probability.

3.5 Artificial Bee Colony

Article Bee Colony [254] is a swarm intelligence algorithm based on the behavior of bees. Within bees, there are different social classes who work together to complete tasks such as harvesting pollen and nesting through the use of smell, 'swing dance' and other methods. Bee colonies consist of three types of bees, queen, male and the worker bees which work on activities such as searching for food, gathering and storage of honey. After gathering honey, the worker bee comes to the nest and informs other bees about the location of the honey source site through a dance. The ABC algorithm simulates the behavior of bees by considering three elements, the food source, employed bees and unemployed bees. The food source is represented as revenue considering its distance and quality, the higher the revenue, the better is the value. In computational optimization terms, the food source is a potential solution to the objective formulation of the considered problem and the quality or value of the food source represents the fitness value of that particular solution. At the start, all bees are used as scouts, searching for food sources. When a food source is found, for a high value food source, bees who find those food sources become hire bees and collect the food source. If the food source is of intermediate value, the bees become follow bees and if the food source value becomes low, the bees become scout bees to look for better food sources. Hire bees, the bees with a food source location having a high value, lead the follow bees to develop solutions in their neighborhood in order to come up with more solutions. A subset of the highest-ranking solutions are then considered before this process is repeated again until the end conditions are met.

4 Optimization Applications in Smart Cities

Several tasks in smart city operations require the use of metheuristics to be solved, the aim being to optimize the resources present in the city. This section presents the different uses of optimization techniques for IoT enabled Smart Cities. This task is performed for all eight components, Smart Agriculture, Smart City Services, Smart Grid, Smart Health, Smart Homes, Smart Homes, Smart Industry, Smart Infrastructure and Smart Transportation.

4.1 Smart Agriculture

Smart Agriculture involves the use of digital technologies such as sensors and intelligent systems to improve agricultural productivity. The sustenance of agriculture depends on water, and it is central to the agricultural production of food items around the world. However, water is becoming an ever-scarce resource due to the increasing demand caused by human population growth, increased economic activity by industries and also due to climate change. It therefore is necessary to utilize this precious resource wisely so as make use of it in the best manner possible. One approach towards ensuring that water and land is used appropriately is by introducing irrigation management schemes such as irrigation scheduling and water allocation in the farming process. A summary of the optimization problems, objectives, constraints in smart agriculture and the use of IoT is illustrated in Figure 8.

Measurements in water irrigation systems are typically performed by sensors placed at different points in the canals and rivers to determine water flow, volume and speed. This information regarding water movement can be combined with economic information such as yeild, crop profit to optimize water distribution. Irrigation management through scheduling has been performed by the authors of [255, 256, 257, 258] to maximize net return on crop profits, water use efficiency and to minimize leakage losses. In [259] Fuqiang et al. aim to optimize water delivery through canals while also performing scheduling. They do this using a genetic algorithm and work with two objectives, minimizing the difference between the time of water delivery and water demand and the fluctuation in water discharge of the main canal. Their model was applied to a district in China.

The authors in [260, 261] work on optimal allocation of water. Of these, Ikudayisi et al. [260] use the differential evolution algorithm to minimize the water allocated to

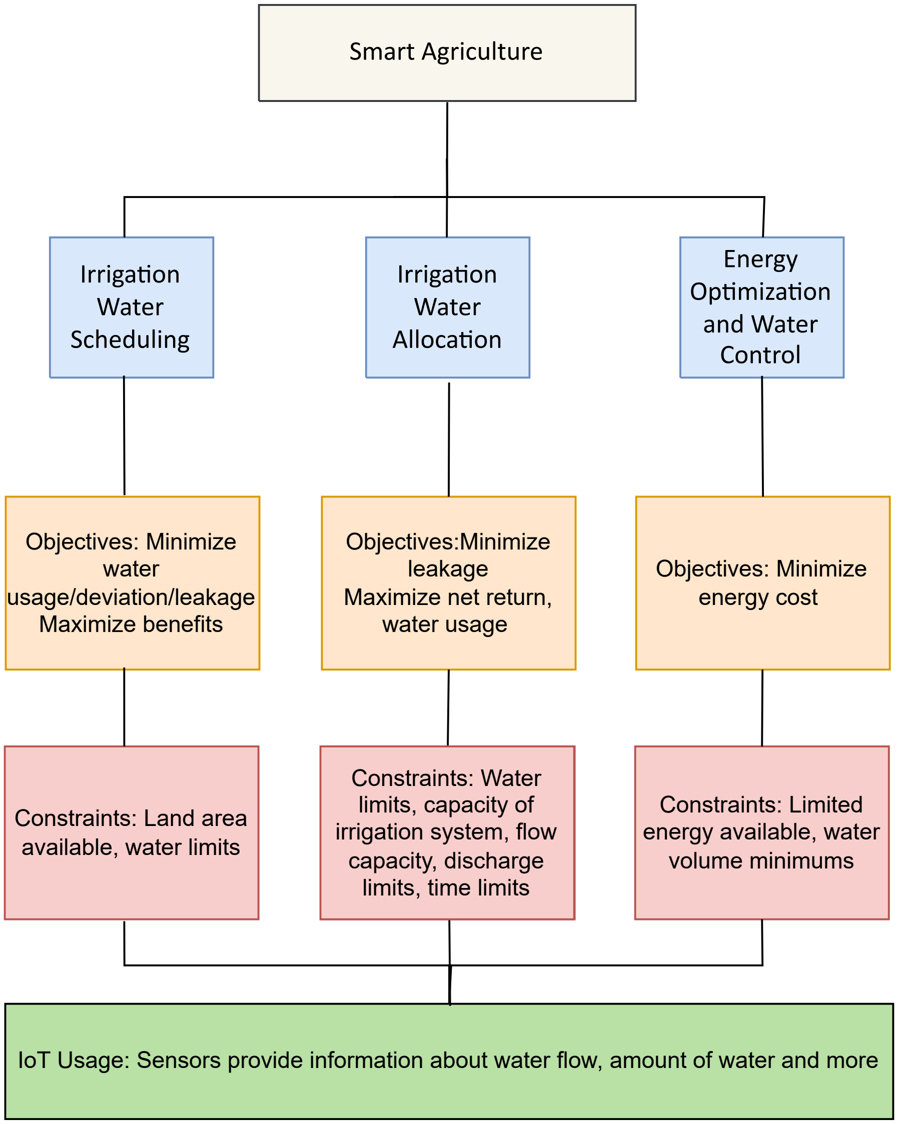

Figure 8. Optimization applications in Smart Agriculture.

farms in South Africa while maximizing the benefits in terms of job creation, ensuring food security and the yield of crops. Wu et al. [261] approach this as problem of reducing deviation between different channels and minimizing water seepage to ensure a more consistent supply to all water channels. An approach presented by Ocampo et al. [262] tackles this problem not as a task of allocation directly but considers the scenario of providing sufficient water to a smart farm while controlling two water pumps. The objective function is formulated to minimize the energy cost of the water pumps while maintaining sufficient supply of water to the plants on the farms. Constraints considered for such applications include the limited quantities of water being worked with, the time for which the water was available and the area of land which was to be considered. Another allocation based scheme is presented by Zhuo et al. [263] who use a genetic algorithm for an irrigation system based on a reservoir and two pumping stations. Their aim is to ensure that there is no water shortage. The objective function used by them is the minimization of the annual sum of squared water shortage. Other work in [264] also minimizes use of groundwater and increase benefit to the regional water supply through reduction of water deficits in the Dujiangyan region of China.

A precision agriculture scheme is presented by Roy et al. [265] who combine an IoT system consisting of sensors on plants using a GA based rainfall predictor. Combining predicted rainfall information along with the sensed moisture level in the crops, they control plant watering. Lin et al. [266] propose a framework for the management of fertigation and irrigation in precision agriculture. They perform short term management and long-term management. Soil and crop growth data is gathered from IoT based sensor systems. Long term planning refers to the allocation of water and fertilizer resources to crops in terms of quantity so as to meet demands whereas short term refers to when and how to use them. They use a genetic algorithm for long term planning considering the allocation of water and fertilizer for crops to maintain pH value and electrical conductivity. This information has been presented in Table 16 while a summary of the identified data sources used by each considered work has been provided in Table 17.

4.2 Smart City Services

According to the world bank, the amount of annual solid waste generated is set to be 3.40 billion tons [268] in 2050. Managing this waste and its collection in an efficient manner is imperative for health and climate reaons. The most common application towards smart city services optimization is waste management as illustrated in Figure 9 which summarizes the objectives, constraints and the use of IoT.

Smart waste collection systems include sensors attached to trash cans which can inform the municipal authority about the status of the garbage amount present in them. Once the trash cans are close to being full, it is the responsibility of the municipal corporation to perform garbage collection in an efficient manner. In this respect, data provided by the sensors on garbage cans can be used to determine an optimized route for garbage collection to construct the Vehicle Routing Problem (VRP) in the Smart City Services domain. As such, this problem has been performed keeping in view various goals. The minimization of the route distance taken by a garbage tuck has been performed by the authors in [269, 270, 271, 272, 273]. The aim in this case is to determine a route for garbage collection vehicles that minimizes the total distance traveled by the them. Zhang et al. [273] consider multi-vehicle allocation while considering the single objective of minimizing route distance. Wei et al. [274] use the Artificial Bee Colony algorithm to determine garbage collection routes resulting in the minimum emission of CO2. Another optimization objective in route optimization for waste management has been the minimization of the total travel time, such a target is described by the authors of [275, 276, 277, 278] who aim to reduce travel time while considering emptying of waste bins. Another optimization consideration in route optimization for waste management is to reduce cost. This has been carried out by Tirkolaee et al. [279] who formulate a multi-objective function of travel cost and total useage cost of vehicles and determine the route with the minimum

Table 16. Optimization in Smart Agriculture

Application	Algorithm	Single/ Parallel problems	Objectives	Constraints
Irrigation Management (Irrigation Water Scheduling)	ACO [255] PSO [256]	Single	Maximizing net return on crop	Constraint on water availability Capacity of irrigation system Water savings should be more than deficiency
	GA [259]	Single	Minimize water fluctuations and difference between the time of water demand and need	Finite canal capacity
	GA [257]	Parallel	Maximize yield, global and local water use efficiencies	Maximum rotation time limitation Constraint on irrigation interval Minimum and max irrigation amount
	GA [258]	Parallel	Minimize leakage loss both individually and overall	Flow capacity limited by maximum Irrigation time constraint Net discharge constraint Total flow rate should be sum of individual flowrates
Irrigation Management (Irrigation Water Allocation)	DE [260]	Single	Minimize irrigation water allocated and maximizes net benefits	Constraint on the land area available Minimum and max planting areas for crops Limited water available for the farm
	PSO [261]	Parallel	Minimize deviation in the different channels, water seepage in the distribution channels	Time Water quantity constraints
	GA [264]	Parallel	Maximize benefit to regional water supply, minimize water deficit groundwater exploitation in regions	Water supply quantity constraints for annual water, ground water
Irrigation Management (Energy Optimization)	GA [262]	Parallel	Minimize energy cost while maintaining water supply for plants	Limited energy available Water volume maintained in storage tank, fish pond
	GA [263]	Single	Minimize sum of squared water shortage	Annual water availability in reservoir Water rights of replenishment pumping station Water rights of the irrigation pumping station Operational rule constraints
Irrigation Management (Water Control)	GA [267]	Single	Maximize yield	Minimum and maximum water depth limits Min and max soil moisture
Irrigation and Fertilizer Management	GA [266]	Single	Maximize economic profits and environmental benefits	Limits on the demand of water for each crop Total water does not exceed available Total fertilizer doesn't exceed availability Water allocation should be positive

Table 17. Data setup used for Smart Agriculture Optimization

Data Type	Papers
Self-collected / Presented	[256, 258, 259, 262, 267, 266]
Government and private agencies	[260, 256, 255, 267, 259, 257, 258, 261]

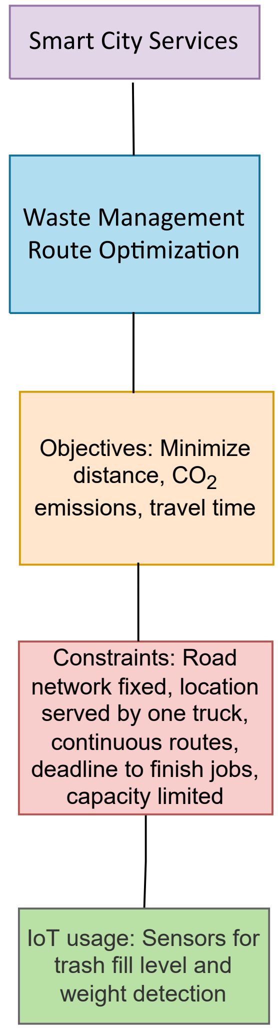

Figure 9. Optimization applications in Smart City Services.

costs using the ACO. Constraints considered in such applications are related to a fixed road network which depends on the locality for which the optimization is being performed, the continuity in the determined route as well as fulfillment of capacity restrictions. The useage of optimization algorithms in smart city services is provided in Table 18 and the data sources used are provided in Table 19.

4.3 Smart Grid

The electricity grid has been a major beneficiary of smart city technologies. The increasing demand for energy by consumers along with the environmental impact that fossil fuel-based energy production has on the planet has forced utility companies to introduce renewable energy sources within the electricity distribution system and make their energy production and distribution systems more efficient through planning and design improvements. Optimization algorithms find applications within the smart grid domain in terms of power management and planning. A summary of

Table 18. Optimization in Smart City Services

Application	Algorithm	Single/ Parallel problems	Objectives	Constraints
Waste Management Route Optimization	ACO [269]	Single	Minimization of distance	Road Network is fixed
	GA [270, 271, 273]			Each dumpster served by one truck only
				Trucks leave depot to go to landfill
	PSO [272]			Routes are continuous
	ABC [274]	Single	Minimize Co2 emissions	Capacity constraint for bins as well as trucks
	ACO [275]	Single	Minimize total travel time	Trucks leave a depot empty
	GA [276, 277]			Bins needs to be fully emptied by trucks
				Vehicle start depot and end at landfill
	PSO [278]			Demand should not exceed capacity
	ACO [279]	Single	Minimize travel cost and total usage cost of vehicles	Subtour elimination
				Jobs should finish within a given deadline

Table 19. Data setup used for Smart City Services Optimization

Data Type	Papers
Self-collected / Presented/ Generated	[269, 270, 271, 275, 279, 277, 273]
Government Agency	[276, 278]
Dataset	Capacitated VRP datasets [280] by [272], Capacitated VRP Instances [281] by [274]

the applications, objectives, constraints and IoT usage for optimization algorithms in Smart Grids has been illustrated in Figure 10.

An increasing population has led to an increasing demand for electricity around the world. This burdening of the electricity grid has led to measures for increasing the performance of the electricity distribution system by reducing loss, prevent overload and reduce cost. The authors in [282, 283, 284, 285, 286, 287, 288] work on the improvement of grid performance by minimizing cost and reducing power losses. Power loss minimization is specifically targeted by [282, 286, 284]. Of these, Ettappan et al. [282] aim for the reduction of power losses, voltage deviation and increasing voltage stability. Atteya et al. [286] also address this problem by considering network redistribution to minimize losses in the grid whereas Sakr et al. [284] focuses on minimizing transfer losses in the smart grid to accomplish this task. Nguyen and Mohammadi [285] attempt the reduction of power losses and line congestion by determining the location of thyristor controlled series compensator devices. The problem is formulated as a multi-objective problem aiming to minimize loadability of the lines, active power loss and the reactance of the transmission line. A cost reduction-based approach to improve grid performance is followed by Das et al. [283] who aim to reduce cost of maintaining electrical stability and also the cost of management of the distribution network. They do this by considering changing the location of energy storage systems within the grid. Kanwar et al. [287] take maximizing profits and

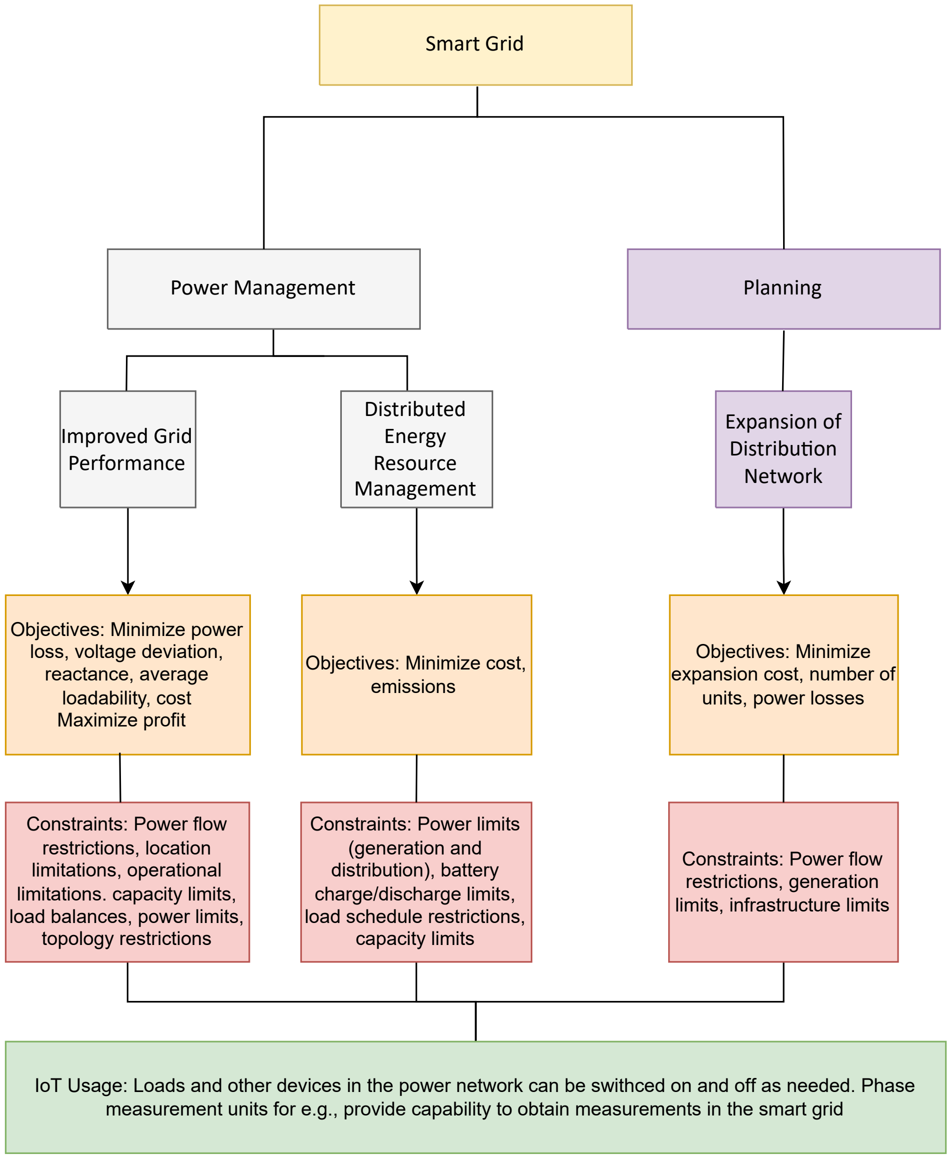

Figure 10. Optimization applications in Smart Grid.

minimization of power losses while considering sizing of a distributed energy resource generation system.

Distributed energy resource (DER) management is another area where optimization algorithms are used in Smart Grids. The management of distributed energy sources within smart grids is dependent on the interconnectivity provided by IoT in the smart grid system. Smart meters within the smart grid provide real-time information relating to power consumption which can be used for controlling DER electricity. Moreover, IoT devices allow for switching loads and generation sources as required. This assists in creating a virtual power plant to aggregate all energy sources in a DER scenario. With global warming and a changing climate, utilities around the world are increasingly incorporating various renewable energy sources within their grid which often times are an economically convenient option as well. However, many of these sources such as wind and solar (photo voltaic [PV]) do not offer a consistent supply of power throughout the day. In this regard, systems such as batteries as well as conventional generation plants need to be used together along with renewable energy sources. For utility companies, it is necessary to optimize power production so that the maximum amount of energy is utilized from these renewable sources so as to reduce cost to the user while also maintaining the quality of service. The authors in [289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302] provide a management scheme for DERs to minimize cost. In this regard, the authors in [292, 293, 294, 297, 289, 290, 300, 301, 302] all formulate the problem of distributed energy resource management as a single objective problem where the cost incurred is minimized. On the other hand, the authors in [298, 295, 291] and [299] formulate this as a multi-objective problem. Azaza and Wallin [299] not only target reduction of electricity production cost but also maximize reliability of the system while reducing the environmental impact of the distribution system. It is interesting to note that the improvement of system reliability is formulated as a minimization problem as well. Similarly, Das et, al. [291] consider the reduction of both the total cost as well as the environmental cost of the system. Their considered scenario also consists of a PV, Wind Turbine and battery. The constraints considered were constraints regarding power flow, limitations on power and voltage values, power balance constraint and power generation constraints on the renewable energy sources. In [289, 302], a DER management system is developed for a microgrid which consists of a controllable collection of energy storage and generation sources powered by IoT devices.

Planning in distribution networks has been considered by the work of [303] and [304]. Mahdavi et al. [303] work on expanding transmission lines utilizing the artificial bee colony algorithm to minimize cost of network expansion, power losses in load and generation. On the other hand, Maji and Acharjee [304] aim to determine the minimum number of Phase Measurement Units (PMUs) to make the distribution network observable. The constraints used were power flow and balance of power as well as limits on the number of transmission lines available. The internet of things also finds usefulness in terms of the use of Phase Measurement Units that provide voltage and current measurement capabilities within smart grids to perform maintenance and monitoring operations. A summary of this discussion has been provided in Table 20 and the data setups used by the covered research work is presented in Table 21.

4.4 Smart Health

Smart health refers to the use of technology to provide better healthcare to patients. This can be in the form of developing tools for better diagnosis of diseases or the use of algorithms for better planning and healthcare delivery. The deployment of timely emergency vehicles to a person in need is imperative towards providing healthcare services to people. Two applications of optimization problems within Smart Health are emergency vehicle routing and their allocation and relocation as shown in Figure 11. It also summarizes the objectives uses, constraints considered and role of IoT.

Late arrival of ambulances and other emergency vehicles to people in need may result in irreversible damage to life and property. Studies have shown that delayed

Table 20. Optimization in Smart Grid

Application	Algorithm	Single/ Parallel problems	Objectives	Constraints
Power Management (Improve Grid Performance)	ABC [282]	Single	Minimize active power loss, volage deviation and voltage stability index (L-index)	Power flow constraints
	GA [284]			Restriction on power source installations and other components related to power structure
	PSO [286, 288]	Single	Minimize power loss	Generation and other component operations within limits
	GA [285]	Single	Minimize average percentage of loadability of the lines, active power loss, reactance of transmission line	Limitation on values of bus voltage
				Transmission line capacity, generator active and reactive power.
	ABC [283]	Single	Minimize cost for maintaining thermal and voltage stability and lower asset management of distribution networks	Active and reactive power must be balanced
				Limits on voltage and load maximum ESS max charging and discharging constraints
	PSO [287]	Parallel	Maximize annual profit by reducing charges for annual energy losses, peak power loses etc	Constraint on the node voltage (soft)
			Minimize power loss for the network reconfiguration	Power injected by DER and SG within limit Power generated at a given node has a limit For reconfiguration: Radial topology, Node voltages has a max hard constraint
Power Management (Distributed Energy Resource Management)	ABC [292, 293, 294]	Single	Minimize total cost	Power generation by renewables within limits
	DE [295, 297, 296]			Battery charge and discharge limits and system reliability
	GA [289, 290]			Power balance constraint (generated equal to consumed)
	PSO [300, 301, 302]			Specific loads are interruptible
				Constraints on the efficiencies of the sources
	DE [298]	Single	Minimize cost and emission	
	ABC [305]	Single	Minimize cost and power imported from outside microgrid	Power flow constraints for the DER
	GA [291]	Single	Minimization of cost of energy and life cycle emissions (CO2 and energy stored in batteries or converted by renewable sources during process of satisfying load requirements)	Constraints on battery capacity
				System reliability constraint Energy produced equal or greater sthan required
	PSO [299]	Single	Minimize reliability cost, cost of electricity production and operation environmental impact ()using renewable factor)	
Expansion of distribution network	ABC [303]	Single	Minimize cost of network expansion, active losses and loss of load and generation	Power flow and active power balanced
				Power generation limits Number of transmission line limits
	PSO [304]	Single	Minimize number of PMUs	SG Network should be observable

Table 21. Data setup used for Smart Grid

Data Type	Papers
Self-collected/ Presented/ Generated	25 Bus network s[295, 297, 298, 291, 299, 300, 301, 302] [305]
Government Agency/ other research work	[292, 293, 294, 290, 299, 303]
Dataset/ Standard Network	IEEE 14 Bus [284, 304]
	IEEE 30 Bus [282, 284, 285]
	IEEE 33 Bus [286, 283, 287, 296]
	IEEE 37 Bus [289]
	IEEE 57 Bus [282, 284, 304]
	IEEE 69 Bus [287]
	119 Node system of [306, 288]

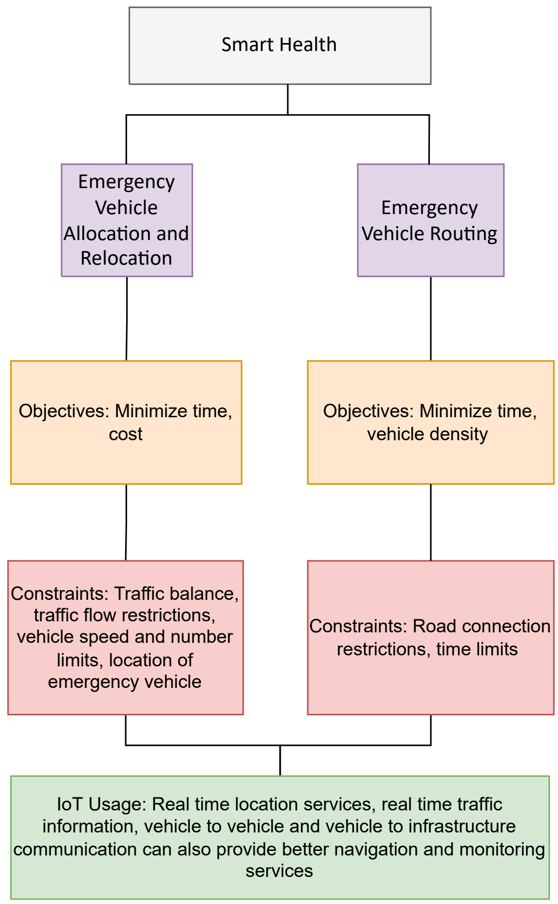

Figure 11. Optimization applications in Smart Health.

ambulance dispatch increases mortality [307], moreover, economically speaking, a one-minute delay in response time for cardiac patients found that the mortality increases by 1% and adds annual costs of USD 7 billion in healthcare expenditure [308]. Keeping this in mind ambulance deployment and location determination have been of considerable interest in the area of optimization for smart health. These two problems are specific cases of the Vehicle Routing Problem [309] and Maximum coverage problem [310] sometimes called the Ambulance Routing Problem [311] and Ambulance Location Problem [312]. The authors in [313] work on the optimal allocation determination based on fixed sites and a finite number of ambulances while minimizing lateness of ambulance arrival using the Ant Colony Optimization. Later on, in their work in [314], they do a comparison with using GAs and find that GAs provide better performance. Kochetov and Shamray [315] attempt localization of ambulance fleet at base stations with the aim to minimize the average waiting time for arrival of ambulances. An interesting approach to this problem is presented in Yan et al. [316] who work on this problem from a scheduling perspective where they control scheduling of emergency vehicles to reduce the total cost in terms of money and time using a Genetic Algorithm. Another approach for sequencing vehicles to ensure emergency vehicles reach their destination in time is presented by Lu et al. [317] who aim to prioritize emergency vehicle thoroughfare on traffic intersections. They do this by minimizing the entrance time of the vehicle by manipulating vehicle order. Constraints used for these problems include constraints on the speed of the ambulances, the flow of vehicles on the road, specific road connections present as well as time constraints. The internet of things serves a pivotal role in enabling the allocation and routing of emergency vehicles. The connectivity provided by IoT through vehicle-to-vehicle communication as well as vehicle to infrastructure communication facilitates providing a real-time indication of the vehicle's location as well as the condition of traffic in a given area. This information can then be used to determine an optimal route for emergency vehicles as well as for their optimal deployment to serve people in need. Information about optimization methods for smart health has been presented in Table 22 and the data setups used in these approaches in Table 23.

Table 22. Optimization in Smart Health

Application	Algorithm	Single/ Parallel problems	Objectives	Constraints
Emergency Vehicle Allocation and Relocation	ACO [313]	Single	Minimize lateness	Ambulance from nearest hospital is dispatched Speed of ambulance Total number of ambulance limits
	GA [314]			
	GA [315]	Single	Minimize average waiting time of ambulances	Balance constraints on exit and entry volumes Flow conservation constraints
	GA [316]	Single	Minimize total cost in money and time	
Emergency Vehicle Routing	PSO [318]	Single	Minimize travel time, road length traveled, density of vehicles on the road	Road connections are specific
	GA [317]	Single	Minimize the entrance time of emergency vehicle by changing the order of vehicles going through intersections	Constraint on the difference between arrival times of current and previous vehicles and on the entrance time of the vehicle

Table 23. Data setup used for Smart Health

Data Type	Papers
Self-collected/ Presented/ Generated	[313, 314, 315, 316, 318, 317]
Government Agency/ other research work	[314, 315, 316]

4.5 Smart Homes

Home energy management has been the prime application of optimization in smart homes, a summary of the objectives, constraints and the use of IoT has been shown in Figure 12.

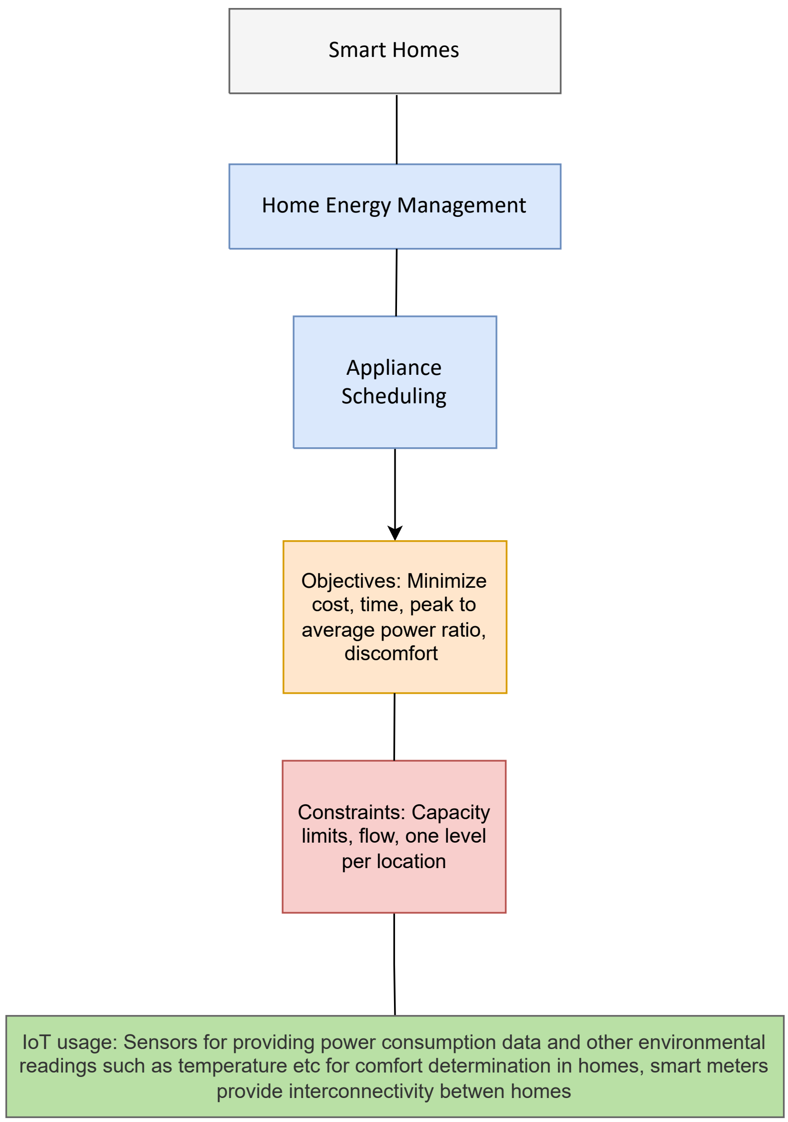

Figure 12. Optimization applications in Smart Homes.

Home energy management refers to the development of demand side management schemes that aim to reduce the electricity cost billed to a customer or maintain comfort for the user. One way this is performed is by appropriate appliance scheduling. The idea here is to schedule the usage of appliances in such a way that the most power-hungry devices are turned on during off peak hours when electricity costs might be lower. The combination of the Smart Grid and Smart Homes facilitates the development of optimization schemes that not only benefit the customer (in terms of reduced electricity costs and maintaining comfort) but also be useful for the utilities in ensuring that load profiles (though minimizing the peak to average ratio) are more consistent thereby allowing better planning of the power generation mix used by them. The authors of [319] perform appliance scheduling for the purpose of minimizing electricity cost and the waiting time for appliance usage. Interestingly, they incorporate comfort maintenance by adding it as a constraint. A similar approach has been followed by Bui et al [320] and Makhadmeh et al. [321] who aim to minimize the cost of electricity usage with a constraint for maintenance of comfort. Makhadmeh et al. [321] also include the reduction of waiting time rate for appliances by the user and the reduction of the peak to average ratio of the power consumed as constraints. The authors in [322, 323, 324, 325] perform appliance scheduling while considering electricity cost and peak to average ratio which need to be minimized. All of the authors present a multi-objective function for this purpose combining the objectives of minimizing the cost and the peak to average power ratio. Azimi et al. [326] combine the problem of reducing cost and power together as a single objective by considering the minimization of the ratio of operating cost and load factor in a battery supported system. The works of [327, 328, 329, 330, 331] also consider user comfort as part of the objective. In [327], Essiet and Wang form a multi-objective minimization problem of electricity cost, peak to average ratio for power and discomfort of users in a smart home supported by a renewable energy system consisting of a battery and PV system. In Chanra et al. [332], the authors aim to reduce electricity cost by appliance scheduling in such a manner so as to make as much use of onsite energy units as possible so as to reduce usage of utility provided electricity. The energy units they consider are a diesel generator, renewables and battery. Another approach that aims to reduce cost of consumed electricity is presented by Faia et al. [333] who formulate it as a problem of minimizing the energy bill and the cost associated with curtailment of power in a system with a battery and a photovoltaic system. Work in [334, 330, 335, 336] also perform appliance scheduling to reduce cost of electricity. Appliance scheduling for smart homes has also been performed by Fatima et al. [323] and Abid et al. [322] considering a microgrid for homes where instead of optimizing data from single homes, the authors used data from connected smart meters to determine an optimized control scheme for appliances across the grid. The constraints used for optimization in smart homes are on the comfort needing to be maintained, constraints on the powerflow, time of operation, the maximum power that is present or used and which appliances are switchable appliances. Appliance scheduling is based on smart meters as well as individual control and monitoring of appliances using IoT systems. IoT devices enable the microgrid which is used to gather data as well as control the switching on and off of sources from the houses electricty supply. The information gathered from these IoT units can be processed to optimize energy consumption patterns to reduce cost to the customer as well as increase comfort. The use of the considered optimization schemes for smart homes has been presented in Table 24 with the data setups presented in Table 25.

4.6 Smart Industry

One of the biggest enablers of the Industry 4.0 concept has been the use of AI techniques to improve the efficiency of the manufacturing and production process. This has led to the development of cyber physical systems aiming to assist in activity recognition [339], machine health prediction [340] and production management in terms of bottleneck prediction [341]. Apart from conventional AI applications of anomaly detection, classification and regression, computational optimization also finds numerous applications as it fits well with the objective of efficient and streamlined manufacturing. The major applications for the use of computational optimization have been in the area of routing and location for logistics and are variations of the vehicle routing problem and are typically represented as Multidepot Vehicle Routing Problem (MVRP), Vehicle Routing Problem Pick-up and Delivery with Time Windows (VRPPDTW) or Large-scale Dynamic Vehicle Routing Problem (LSDVRP). Figure 13 summarizes the objectives utilized, constraints and the role of IoT in optimization for Smart Industry.

The authors in [342] and [343] use the ABC and the GA respectively to determine the best location of service sites for logistic operations. Both these approaches use multi-objective formulations aiming to reduce cost of operations, transportation as well as the establishment of the centers. The authors in Su et al. [344] use ACO, Alinaghia et al. [345] PSO and Utama et al. [346] use ABC to address the problem of determining the best route for logistics operations. The routing and coverage problem for logistics involves determining the best route for either a single or multiple vehicles at a depot which have to visit every customer. The works of [344, 345, 346] focus on reducing the cost incurred in the routing for vehicles in logistics as a single objective formulation. On the other hand, the authors of [347, 348] and [349] all work on the minimization of distance as their objective in determining the optimal route for delivery vehicles trying to serve multiple locations. Mounia and Bachir [348] address routing in logistics as a multi-objective problem where they not only aim to minimize the distance traveled by the vehicles but also aim to reduce CO2 emissions and the number of vehicles used. A time based optimization approach is presented by the authors of [350] and [351] with [350] also factoring in reduction of fuel consumption in their objective function formulation. Constraints used for the routing and location determination problem are related to time, capacity constraints for the vehicles, each customer being served only once, constraints related to the route. The determination of the location and the route for vehicles is dependent on real time information concerning the traffic in the area as well as the loads to be collected from each site in addition to other information which can be provided by IoT units. The usage of optimization algorithms for smart industry has been presented in Table 26 with data setups presented in Table 27.

Table 24. Optimization in Smart Homes

Application	Algorithm	Single/ Parallel problems	Objectives	Constraints
Home Energy Management	ACO [319]	Single	Minimize cost and waiting time	Comfort needs to be maintained
	ACO [322]	Parallel	Minimize cost and peak to average ratio	Power flow constraints
	ACO [323]	Single	Minimize cost and peak to average ratio	Maximum energy capacity constraint
	DE [324]			Device counted that can be shifted is positive
	PSO [325]			Number of devices shifted at any time should not be more than the available number of controllable devices
	GA [337]	Single	Minimize peak to average ratio for load shaping	Load shaping, redistribution of load in a flexible manner
	GA [326]	Single	Minimize ratio of operating cost and load factor	Charging and discharging of batteries
				Complete load transfer and load clipping limits
	DE [327]	Single	Minimize electricity cost, peak to average ratio of power and discomfort minimization of users	Constraints on PV supply limits
	ACO [328]			State of charge and rate of discharge of battery
	DE [329]	Single	Minimize electricity cost and discomfort	Time of operation within specified limits
	PSO [331]			Temperature, air quality, illumination and energy should be within maximum limits
	GA [330, 338, 305]	Parallel		A given appliance must be on for specified times of the day
				Power limits to be followed
	ABC [320]	Single	Minimize cost of electricity	Appliances for comfort have fixed times
	DE [332, 334]			Some appliances cannot be delayed
	GA [335]			Power balance constraints
	PSO [321, 336]			Surplus solar power sold back to distribution system
				Maintain zero net energy in building
				Time constraints
				Load safety factor
				Load phases of appliances fulfill energy requirements
				Comfort needs to be maintained
				Peak to average power ratio balancing
	PSO [333]	Single	Minimize energy bill and cost associated with KWH curtailment	Power values within limits, battery charge and discharge limits

Table 25. Data setup used for Smart Homes

Data Type	Papers
Self-collected/ Presented/ Generated	[322, 319, 323, 324, 325, 326, 327, 329, 305, 338, 330, 320, 334, 335, 321, 336]
Government Agency/ other research work	[322, 323, 337, 327, 338, 331, 332, 321, 333]

Table 26. Optimization in Smart Industry

Application	Algorithm	Single/ Parallel problems	Objectives	Constraints
Location determination for sites	ABC [342]	Single	Minimize transportation and hub establishment cost	Single allocation for each demand node A given number of hubs are established Covering radius constraint Time reliability constraint
	GA [343]	Parallel	Minimize distribution cost and maximize profit	Load capacity meets needs of customers A delivery vehicle can only be delivered when it receives a task Capacity constraints
Routing for Logistics	ABC [348]	Parallel	Minimize distance travelled, CO2 emissions, number of vehicles used	Every customer visited only once Every vehicle visiting a location must leave it too Ensure route continuity Demands of any route must not exceed capacity Edges satisfying time window constraint are allowed.
	ABC [349] GA[347]	Single	Minimize total transportation distance	Each customer served only once Route should start and end at the same depot Served demand of each vehicle does not exceed capacity limit
	ACO [344] PSO[345] ABC [346]	Single	Minimizing total cost	Each customer served only once Dispatched vehicles not more than available Vehicle routes don't contain disconnected routes Customer demand shouldn't be larger than vehicle capacity
	ABC [351]	Single	Minimize travelling time	Vehicle load constraint Subtours not allowed Speed, time and distance Maximum number of vehicles on a route Each customer served by one vehicle Vehicle number max limit
	PSO [350]	Parallel	Minimize fuel consumption and travel time	Each customer serviced by only one vehicle Continuity in route Vehicle load conservation between nodes, First in first out proper when traveling time is computed Time taken for customers as stated, Maximum time for servicing Vehicle capacity constraint Depot is the first and final destination of each vehicle

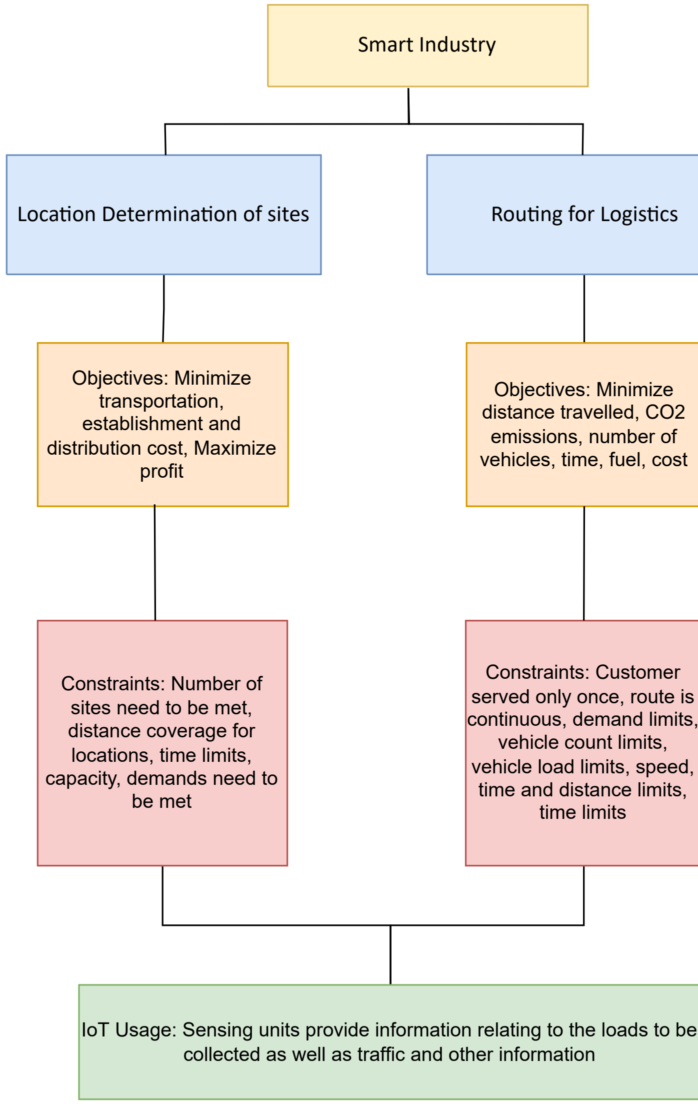

Figure 13. Optimization applications in Smart Industry.

4.7 Smart Infrastructure

Within the infrastructure domain, the most common optimization problem is the area of health monitoring of structures. Structural Health Monitoring (SHM) is a necessary application within the smart infrastructure domain as it makes for safe usage of different structures of public use. These structures can be buildings as well as transport structures such as bridges, tunnels. Structural health monitoring typically involves the use of sensors attached to a structure at several points that can gauge some type of physical variable (vibration, strain, acceleration, temperature, tilt etc) from the structure. Data gathered from these connected sensors is then used to determine if any structural damage has taken place or not. Within the domain of SHM, optimization algorithms find application towards the Optimal Sensor

Table 27. Data setup for Smart Industry

Data Type	Papers
Self-collected / Presented/ Generated	[342, 343, 347, 351]
Government Agency/ other research work	[348, 349, 344, 346, 351]
Dataset/ Standard Network	Test instances in [352] used by [345] and [350]

Placement (OSP) Problem as illustrated in Figure 14. Figure 14 summarizes the objectives used, constraint and the use of IoT.

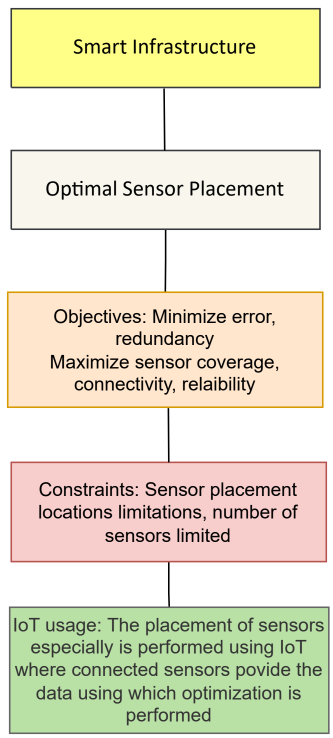

Figure 14. Optimization applications in Smart Infrastructure.

For the optimal sensor placement problem, the aim is to determine the best number and placement of sensors over a structure so as to reduce the number of sensors used as well as improve the measurement process, both these aims result in increased reliability of the SHM system as well as potentially lower the cost of the system too. The authors in [353, 354, 355, 356, 357] work on the placement of sensors for structural health monitoring focusing on improving the effectiveness of the deployed system. In this regard, [353] and [356] use the genetic algorithm to solve a multi-objective problem aiming to minimize the measurement error and cost. Yang et al. in [355] formulate OSP as single objective minimization where they aim to reduce the ratio of sensor placement performance to the redundancy of information resulting from each tested placement. Another approach that works on the error is presented by [354] who use the Particle Swarm Optimization to maximize the reconstruction

Table 28. Optimization in Smart Infrastructure

Application	Algorithm	Single/ Parallel problems	Objectives	Constraints
Sensor placement	GA [353]	Single	Minimize measurement error and measurement cost	
	PSO [354]	Single	Maximize reconstruction accuracy and robustness of transfer relationship between deformation displacement and surface strain (formulated as a minimization problem for negated accuracy and robustness)	Sensor placements within predefined range and angles
	GA [355]	Single	Minimize the ratio of sensor placement performance to redundancy information	Sensor placement is permitted on chosen location
	GA [356]	Single	Minimize the MAE between the system and the estimated response (global error) and minimize the maximum difference between the system and its estimated response (local error)	Sensor locations are from a set of predefined locations
	DE [357]	Single	Maximize quality of coverage, lifetime, connectivity uniformity of sensor nodes and cluster heads and reliability	Constraint on the number of cluster heads associated with each sensor node and cluster head
	GA [358]	Single	Minimize cross correlation of the sensing network	Sensor placement is permitted on chosen location

Table 29. Data types for Smart Infrastructure

Data Type	Papers
Self-collected/ Presented/ Generated	[353, 354, 355, 356, 358, 358, 359, 360]

accuracy and robust transfer relationship between the deformation and surface strain with different sensor placements. It must be noted that the objective function is formulated as minimization of negated accuracy and negated robustness measurement. Optimized structural health monitoring for aircraft monitoring has been targeted in [358]. In their setup consisting of vibration sensors, the authors optimize sensor placement by minimizing the cross correlation of the vibration waves in the sensing network. The most common constraint for sensor placement is restrictions on the places where sensors can be placed. This information has been provided in Table 28 and the data setups are presented in 29.

4.8 Smart Transportation

One of the most popular optimization applications within smart cities are within the smart transport domain. These include parking system routing, traffic signal control and scheduling. A summary of the applications, their objectives, constraints and the role of IoT is illustrated in Figure 15.

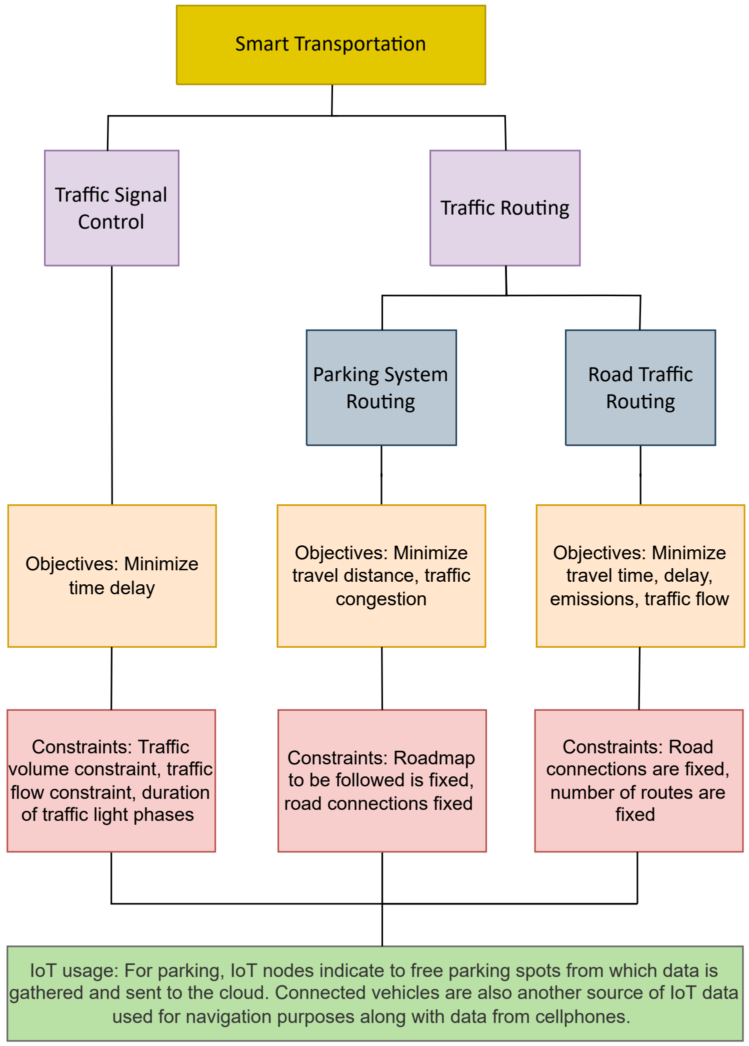

Figure 15. Optimization applications in Smart Transportation

Smart transport systems consist of sensors along roads and traffic intersections to measure relevant parameters while also providing communication services between vehicles and infrastructure. This allows for measurement of the current state of roads in terms of traffic congestion and usage thereby allowing for the use of optimization techniques to improve trip experiences for users and make the transportation system more efficient. The authors in [361, 362, 363, 364, 365] work on the minimization of time (wait and travel) in traffic signal control. The aim of such systems is to reduce traffic build up on signal intersections. Of these, the work in [361, 362] and [363] use the artificial bee colony and the genetic algorithm respectively for a single objective function of minimizing delay time. An interesting approach for this problem is presented by Li et al. [365] who use a multi objective formulation targeting the minimization of the average travel time both overall and individually for all vehiclesl. Another multi-objective approach in traffic signal control is presented by Chen and Yuan [366] who form a mixed problem of minimizing vehicle emissions and travel time together. Korkmaz [367] work on the estimation of delays in traffic signals using a genetic algorithm, they use it to minimize the difference between the estimated and simulated values. Tang et al. [364] carry out distributed optimization in a fog and cloud hierarchy. First, fog nodes optimize phase timings within a single cycle and if the number of vehicles exceeds a threshold, the results are sent to the central controller to further optimize over different cycles so that a traffic jam is avoided or alleviated. Zhang et al. [368] attempt traffic signal optimization using multi objective optimization functions of reducing time delay and increasing traffic capacity. Constraints used for traffic signal control are timing constraints on the phase durations, flow rate of vehicles and on the travel time.

Traffic routing is also another important aspect in smart transportation. This typically involves the determination of the best route to the destination keeping in view various criteria such as reduction of distance, time, cost etc. The problem of traffic routing is addressed by the works of [369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379]. The authors in [369] and [370] use the ant colony optimization and genetic algorithm to minimize the travel distance in parking system routing. They aim to minimize distance traveled by a driver looking to find a free parking spot, using the algorithm, an optimized route is determined for the parking spot. In, [371, 373] and [372] the ant colony optimization algorithm is used to determine the best route in a generic traffic scenario where cars can communicate with road side units in a VANET architecture. Routing for public transport is performed by [380] and [375] in a connected vehicle scenario aiming to minimize travel time. An economic objective approach to traffic routing is taken by the authors of [376, 378] and [379] who minimize the total cost of the trip. Mao [379] also include traffic congestion and travel time as well in their computation. Hassoune et al. [381] work on a parking guidance using the ant colony optimization algorithm to reduce traffic congestion and minimize distance. Constraints for traffic routing are related to the road network allowing travel in specific directions, signaling and travel time. Within smart transportation, IoT nodes are used to determine occupied parking spaces and this data is used for routing applications in parking. Traffic routing is based on vehicle to vehicle and vehicle to infrastructure communication provided by VANETs within the IoT framework. These systems enable cars to exchange data with each other and also with fixed infrastructure on the roads. This discussion is also presented in Table 30 and the data setups for the covered work are presented in Table 31.

5 Conclusion

This chapter provides coverage of the application of five popular computational algorithms in the IoT enabled Smart City. It provides a mapping of the various applications to the specific smart city domain as well as highlights the different formulations of the objective function used to solve the considered problem. This coverage is provided in terms of the number of objectives as well as whether the problem was solved as a single objective, in a hierarchical manner or otherwise. It also highlights the constraints used by the researchers in solving the problem which is an important aspect as constraints are governed by the application at hand. An overview of the mapping of various smart city optimization applications derived from this work is provided in Figure 16.

Table 30. Optimization in Smart Transportation

Application	Algorithm	Single/ Parallel problems	Objectives	Constraints
Traffic signal control	ABC [380]	Single	Minimize travel time	Interval of feasible green time length values
	ABC [374]			Interval of feasible offset time length values Constraints on cycle lengths
	ABC [361, 362]	Single	Minimize time delay	Only one active stage
	GA [363, 364]			Flow dynamic constraint
	GA [365]	Parallel	Minimize time delay and also achieve traffic network equilibrium	Link volume constraint Constraints on duration of green/red phases Offset phase duration Minimize average travel time. Relationship between route and link flows need to be maintained as defined
	GA [366]	Single	Minimize vehicle emissions and travel time for vehicles	Sum of green time of each phase is equal to total available green time Green time is set by a lower bound
	GA [368]	Parallel	Minimize delay, and exhaust emission and maximize traffic capacity (formulated as minimization problem)	Cycle length of signals has minimum and maximum limits
Traffic Routing (Parking System)	ACO [369]	Parallel	Minimize distance with bend straightening and turn reduction	Bend straightening and turn reduction
	ACO [381]	Parallel	Reduce traffic flow and shortest distance towards parking	
	GA [370]	Single	Minimize distance	Specific prefixed routes possible for free parking
Traffic Routing (Road Traffic)	ACO [371, 373]	Single	Minimize distance, minimize congestion	Follow roadmap
	ACO [372]	Single	Maximize flow	
	ACO [375]	Single	Minimize travel time	Constraint on relationship between green time lengths cycle length, offset on the network calculation
	GA [377]			Interval of feasible green time length values Interval of feasible offset time length values Specific road segments Connected constraints on the values of time taken for vehicles
	DE [376]	Single	Minimize travelling cost and rental cost	Each bus has one employee Employees can be assigned when stop is available Bus stop assigned when bus is in use Constraint on distance of bus stop from employee home and more
	DE [378]	Single	Minimize total cost	Road network connections followed Solutions contains correct number of routes
	ACO [379]	Single	Minimize transit time, travel distance, road congestion and traffic expenses	Variable value constraints

Table 31. Data types for Smart Transportation

Data Type	Papers
Self-collected / Presented/ Generated	[374, 362, 361, 363, 364, 365, 366, 368, 369, 381, 370, 371, 373, 372, 380, 377]
GovernmentAgency/ other research work	[361, 365, 366, 370, 374, 375, 377, 376, 378, 363]

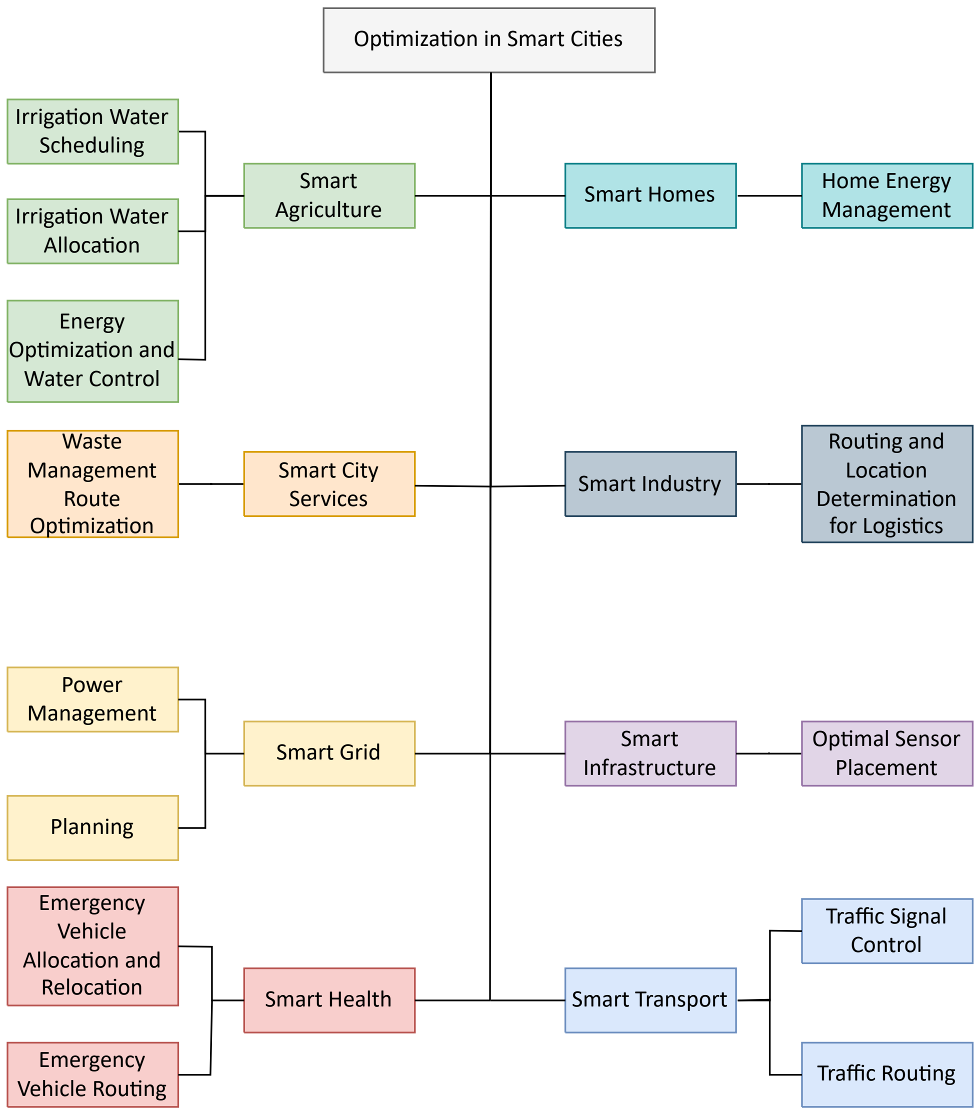

Figure 16. Optimization applications in IoT based Smart Cities.

CHAPTER V

CASE STUDY - SMART HEALTH

1 Introduction

As discussed earlier, IoT finds usage in many different Smart City domains, both for use in classification/regression tasks as well as for optimization applications. One of the most important applications of IoT within Smart Cities is within the Smart Health domain wherein IoT devices enable the monitoring of patients for different diseases such as heart disease etc. While there are a multitude of IoT within Smart Health, in order to develop algorithms on IoT sensor data, the case of ambient assisted living was chosen, in particular, this work considers the case of fall detection.

2 IoT for Fall Detection

The United Nations estimates that the number of old people (aged 60 and above) will be 2.1 billion by 2050 and 3.1 billion by 2100 [382] leading to an increased old age population. With people above 65 years suffering the greatest number of fatal falls among adults aged 60 and over [383] healthcare costs related to fall related injuries run in to billions of dollars. These falls can result in injuries of moderate to severe nature in the people experiencing the falls and may lead to decreased mobility [384], especially for the elderly [385]. Furthermore, following the initial fall, the likelihood of experiencing additional falls increases [386] and can lead to mental stress in the form of post-fall syndrome [387].

The Internet of Things has spearheaded the development of cyber-physical systems that facilitate the recognition of activities/events being performed by people in their daily life. One aspect of this application scenario is fall detection. Falls can occur due to a variety of reasons [388] and Fall Detection Systems (FDS) are used for people with different of health related ailments such as in Parkinsons disease [389], epilepsy [390], arthritis, people suffering from cardiovascular diseases and various neuro-degenerative diseases [391]. In any scenario, it is pertinent that when a person undergoes a fall, they be provided care as quickly as possible. Providing timely care after a fall may improve post-fall life quality of a patient. Fall detection Systems can play a vital role in contributing to the provision of timely care [389, 392] by alerting healthcare professionals. As noted by [393] fall detection systems are necessary for old people with cognitive impairments who may not be able to get up after a fall for long durations of time which may result in pressure sores and other complications and due to vulnerability to injuries when experiencing a fall, it may sometimes result in death [394].

A fall is an unintended and sudden change of posture resulting in one resting on the ground or some other lower level elevation. The aim of a FDS is to monitor the movement of a person and determine when a fall has taken place with the aim to alert healthcare personnel or other caregivers. Over the years different methodologies have been suggested to address this problem, however, before discussing the different methods utilize for fall detection purposes, it is important to mention what researchers enumerate to be the traits of a 'good' FDS [395].

2.1 Challenges/Requirements of a FDS

The following are the typical typical traits/requirements of a good fall detection system.

2.1.1 Non-intrusiveness

An important requirement in fall detection systems is to be non-intrusive. Any FDS should not be an impediment in the execution of daily activities of a person as it may restrict movement or inconvenience their lifestyle.

2.1.2 Low Latency

An FDS needs to be able to detect a fall with low latency. Latency refers to the time duration between the occurring of a fall and its detection. FDS need to detect falls as quickly as possible to ensure that caregivers can be notified right away so that apt care is provided to the person. Low latency depends on not only the algorithm being used to detect falls but also parameters like network speed and reliability. Sensor data transmission to the main system might also be included in this calculation depending on the FDS proposed.

2.1.3 Low power consumption

Power consumption is an important issue in FDS, especially for ones which are battery powered. Many FDS are expected to be used by elderly people who might not be as regular in charging the FDS, therefore FDS should consume as little power as possible.

2.1.4 Allow mobility

A FDS should not hamper mobility as it might be preventive in adoption of a FDS too. FDS which require people to remain infront of a camera at a certain viewpoint etc can be difficult to deploy and use.

2.1.5 Differentiate between Falls and Activities

A FDS should have low false positives and false negatives. In other words it should be able to confidently detect falls and be able to differentiate them from activities of daily living such as walking, sitting, jumping, running etc. Any false alarms will result in wastage of valuable healthcare resources, on the contrary, any false negatives might cause the patient/person having undergone the fall to not get appropriate medical care in time and lead to potential death or trauma.

2.1.6 Notify caregivers

A complete fall detection solution should have some mechanism of informing caregivers whenever a fall is detected. Typically this is carried out in terms of email, notifications via some system or messages. Moreover, this process should be quick so as to ensure the dispatch of any needed help as quickly as possible.

2.1.7 Track history

A complete fall detection solution should be able to provide the history about a patient in terms of falls that a person may have suffered from as this will help monitor the patients health and determine likelihood to suffer from falls in the future. FDS may sometimes also look at other biological parameters during falls so as to help investigate the causes of falls using the conditions prevalent at that time. Biological parameters that could be measured include heart rate, perhaps an ECG or others as deemed necessary based on a patients history.

2.2 Types of Fall Detection Systems

Fall detection systems can be categorized in to three types based on the sensing mechanisms used and their placement. These are Ambient Sensor based systems, Visual based systems and Wearable Sensor based systems.

2.2.1 Ambient Sensor based systems

Ambient sensor based systems are also sometimes referred to as environmental sensor based systems since these types of FDS rely on the use of a number of wirelessly connected ambient sensors placed in a given area for the detection of falls. These FDS make use of various sensors including Passive InfraRed (PIR) Sensors [396], Acoustic sensors such as microphones, thermal sensors as well as sensors measuring Wi-Fi signal strength etc. For e.g., the authors in [397] develop a fall detection system based on two PIR sensors and a vibration sensor to detect vibrations on the floor when a person falls down. Using the PIR sensor to detect a persons motion. feature extraction is performed on vibration measurements and an SVM classifier is used to determine whether a fall has taken place or not. Another approach presented in [398] utilize sounds in a home to determine falls in the elderly. They do this by extracting features in sound signals and like [397] use a Support Vector Machine classifier to determine falls. The authors in [399] make use of Wi-Fi channel state information for fall detection. They achieve this by feature extraction in the frequency domain followed by an SVM classifier.

Ambient fall detection systems offer the advantage of not requiring people to wear anything on their bodies and are thus nonrestrictive in that sense. However, such systems are prone to suffer high false positives due to the large number of external factors such as other heat emitting devices, multiple people moving around at the same time etc. Another issue that comes with ambient fall detection systems is that deployment could be cumbersome to cover a large space, therefore they are typically deployable in small places such as a home or a small building but are very difficult to use outdoors due to the nature of the setting as well as the possible sources of external interference.

2.2.2 Vision based systems

Vision based systems rely on the processing of video frames or other visual information such as depth and thermal images or their combination [400] to detect falls. Typically, the recorded data is sent to a server or central computer which provides processing capabilities. Moreover, recently, with the use of deep learning algorithms, such systems have found to provide improved performance. For e.g. [401] use Convolution Neural Networks on velocity information in video frames. Another approach that uses CNNs is given in [402] who model human motion from video frames using CNNs and then use a logistic regression classifier to determine falls. Other approaches with CNNs include the works in [403] and [404], both of which achieve very high detection accuracy. Another DL network which has been useful in fall detection systems based on cameras is Long Short Term Machine, a recent example of that is the use of a CNN + LSTM combination on 360 degree video recordings of an indoor facility to determine falls. The LSTM allows for capturing the changing characteristics between video frames to provide fall recognition.

Eventhough vision based FDS have been able to provide improved fall detection performance recently, like ambient sensor based systems, these systems suffer from some drawbacks. They are affected negatively in terms of performance and deployability from external factors such as occlusion, restricted mobility of the people being monitored, typically usable in environments over a small area, have high computational costs and are expensive. One other issue with vision based systems is privacy, vision systems which use normal cameras pose risk of violation to the privacy of people being captured on video. The question of privacy can be circumvented by the use of depth cameras for e.g. a kinect [405, 406] or using heatmaps using infrared cameras/sensors [407].

2.2.3 Wearable Sensor based systems

Wearable fall detection systems involve the use of a device attached to the subject. The wearable device monitors user activity by means of sensors and can determine when a fall has occurred. Information about the fall is then conveyed to a doctor or other medical professionals. Various types of sensors can be used in these systems including inertial measurement sensors such as accelerometers, gyroscopes, magnetometers [408] or health sensors such as ECGs, oxygen level and pulse rate sensors as well. The devices that contain these sensors can be worn on the hand, wrist, arm, chest, waist, legs, thigh or even put inside shoes and are usually battery powered. Wearable fall detection systems have the advantage that they do not restrict movement of the person using them and consume less power than the other two approaches, however, a disadvantage to this is the need to 'carry' the device all the time. Another disadvantage is the need to charge the device regularly which might be difficult to do for older people, a way this has been addressed in some approaches is to make use of energy harvesting [409].

2.3 Types of Wearable FDS

There are two types of wearable systems in use for fall detection purposes.

2.3.1 Threshold based systems

The first of type of wearable FDS are threshold-based fall detection systems where authors have used various thresholding techniques on sensor values measuring motion, particularly the accelerometer. The major advantage of threshold based fall detection systems is their relative low computational cost and use of sensor fusion techniques to improve fall detection accuracy , however, the big disadvantage of such systems is that threshold based systems do not generalize well across subjects since people may have different gaits, this is especially true for people of different ages. As a result, threshold based FDS may suffer from having many false positives as well as false negatives since the FDS needs to differentiate between activities of daily living and falls, which many times may appear to be fairly similar to each other in terms of sensor readings.

2.3.2 Machine/Deep Learning based systems

Since threshold-based systems are rigid in terms of their usage, ML/DL methods provide a flexible method for fall detection. ML/DL algorithms can learn complex patterns from sensor data which indicate to a fall while monitoring a persons activities and therefore are useful in fall detection systems. As opposed to threshold based systems, their generalizing capability allows them to be used across subjects. A typical ML framework for fall detection will involve data acquisition, prepossessing, feature extraction and then usage of a suitable ML algorithm to make the inference for a fall or no fall. A deep learning framework based FDS might have a similar methodology, but depending on the DL algorithm used, feature extraction might be omitted.

2.4 Sensors used in Wearable FDS

As mentioned before, the sensors used in wearable FDS are accelerometers, gyroscope, magnetometer, ECG, Pulse Rate and oxygen saturation levels. In this section, we will discuss briefly the role of these sensors in wearable FDS and parameters of interest when using them in FDS.

2.4.1 Accelerometers

Accelerometers, together with gyroscopes are the most widely used sensors in wearable FDS. One reason for this is that both these sensors are typically present in smartphones, which have been used to either collect data for and/or deploy fall detection systems. An accelerometer provides a measure of absolute acceleration and are used to detect vibrations, force in a variety of applications involving monitoring of machines, planes, civil architectures and more. In an FDS, an accelerometer is able to capture the movement patterns of individuals which can be used to determine falls.

2.4.2 Gryroscope

Gyroscopes are used to measure tilt/orientation. Gyroscopes find applications in various areas where orientation measurement is required, for e.g. in airplanes and submarines where they are used in stabilization systems, smartphones and game remotes for interactive gameplay, among other things. Since orientation is an important characteristic of a fall, gyroscopes are one of the most widely used sensors in FDS along with acceleratorometers.

2.4.3 Magnetometer

A magnetometer is a device that measures the direction and strength of magnetism relative to the earths magnetic north. In addition to other applications, magnetometers are used in aircrafts for direction referencing. These sensors are not very popular in FDS but have been proposed in some approaches [410, 411] to be used for fall detection in addition to accelerometers and/or gyroscopes.

2.4.4 Various Medical Sensors

Wearable fall detection systems may also incorporate different types of medical sensors such as ElectroCardioGraphs, ElectroMyoGraphs, Pulse Rate or Oxygen Level measurement in the FDS solution. Even though there have been some approaches which have used medical sensors exclusively for fall detection (such as EMG[412]), however, like magnetometers, these sensors are usually used along with accelerometers and/or gyroscopes. Adding these sensors to fall detection systems has the added advantage of assessing falls from a health standpoint too as this information can be used to determine risk of falls [413].

2.5 Design considerations for Wearable FDS

Wearable fall detection systems utilize sensor measurements to ascertain the occurrence of a fall. Since wearable FDS units are always attached to a persons body, they continuously provide measurements of a persons movements and activities. In order to use these measurements in a FDS, there are some design considerations involved. We discuss them in this section.

2.5.1 Sampling frequency

The sampling frequency dictates the number of sensor measurements recorded per second. The sampling frequency used in an FDS should be high enough to capture fall motion but not too high so as to increase data processing, storage requirements as well as allow for energy efficient operation. In fact, depending on the position of the sensors, significant reductions can be made in the sampling frequency used for sensors with little or no change in performance as demonstrated by [414]. Such reductions could result in low power consumption by the wearable module thereby enabling longevity of operation.

2.5.2 Windowing

Since wearable fall detection systems provide a continuous stream of sensor data, the data needs to be windowed for fall detection to be performed. Windowing refers to the extraction of a subset of the sensor data stream in a sliding manner or otherwise and is typically specified in terms of time duration for its size. It is typical for wearable FDS designers to test different window sizes for their fall detection algorithms. The size of the window used would dictate memory and computational requirements of the unit used for deploying the FDS system. Another important factor in windowing sensor signals is whether overlapping is used during the windowing process or not. This might be important to allow for smooth transitions between different windows.

2.5.3 Feature Extraction

Feature extraction is the process of determining quantities from data which characterizes it for a desired task appropriately. Feature extraction not only aims to capture important aspects of the data being worked on but also to present its content in a reduced size, thereby, making use of the data easier. Feature extraction is a typical step of the a signal processing and machine learning workflow and follows the windowing step. There are different types of features which have been computed for fall detection applications, common feature computations include the determination of time domain features, frequency domain features, statistical features etc. or some combination from sensor data in FDS. Feature extraction is followed by either a thresholding algorithm to sensor values or a machine/deep learning model. For deep learning, there have been some end to end deep learning approaches which skip the step of manual feature extraction. In such approaches, deep learning networks such as CNN may be used to extract features followed by other networks such as LSTM to determine falls, an example is proposed in [415].

2.6 Literature Review

As discussed previously, wearable fall detection systems present several advantages when used for fall detection purposes such as non-intrusiveness, ease of mobility, small size of deployment, low cost and that many implementations are typically standalone systems. This added with the fact that sensors and other components which are used in such systems have been becoming cheaper and cheaper to produce has resulted in a lot of interest in the development of wearable FDS by researchers and technologists alike. Therefore, wearable FDS methodologies have gained considerable interest in fall detection research. We discuss some of the popular methodologies proposed in this regard.

The authors in Mrozek et al. [416] present a complete design of a wearable fall detection system for an Internet of Things scenario. They conduct two experiments with their system, in one experiment, data is collected from a gyroscope and an accelerometer and sent to the cloud for fall detection (as a web service) while in the second experiment, fall detection is performed on the edge node (smart phone). Through their experiments they conclude that performing fall detection on the edge node results in less network traffic and storage requirements for the cloud. Algorithmically speaking, they extract 3 second overlapping windows from measurements of both sensors, perform feature extraction and pass it on to a boosted decision tree classifier for determining whether a fall has taken place or not. Moreover, the boosted decision tree classifier was trained using the SisFall [417] dataset. Another IoT based system is proposed by Marquez et al. [418] who develop a fall detection system for IoT on the edge. They first gather data from an accelerometer and a gyroscope placed on the waist of multiple subjects and train a support vector machine classifier on a combination of raw sensor values and its standard deviation. They are able to deploy their system successfully and achieve satisfactory results. Moreover, feature selection is also performed before the classification stage. A fog based fall detection model has been proposed by Sarabia-J'acome et al. [419] who utilize a Long Short Term Memory Recurrent Neural Network in their method. Edge nodes collect Inertial Measurement Unit (IMU) measurements and relay them to a fog node which processes the data to determine if a fall has taken place or not. One motivation for them to develop RNN models over a Convolutional Neural Network one is the requirement of less parameters and thus less computational requirements. Their final model consists of a 1D CNN layer followed by two LSTM layers and a fully connected layer for classification and is trained using raw data (window of 15s) from the SisFall dataset.

Zurbuchen et al. in [420] provide a comparison of various machine learning algorithms for fall detection while using data from the SisFall dataset. They perform experiments using Decision Trees, Random Forest Classifier, Gradient Boosting tree, K-Nearest Neighbor and Support Vector Machines. Segments of 10 seconds are extracted from the trials in the SisFall dataset and various time and frequency domain features are computed for those segments. These are then passed on to the classifiers. Through their experiments, they find that Random Forest and Gradient Boosting trees perform the best among the considered methods. Another comparison of various ML algorithms is provided by Chelli and Pätzold in [421] who compare the performance of a number of machine learning algorithms on two datasets, the Cogent Labs dataset [422] and the one provided in [423]. The algorithms they consider are an artificial neural network, K-NN, an ensemble bagged tree (EBT) and Quadratic Support Vector Machine (QSVM). They compute several time and frequency domain features from accelerometer and gyroscope signals and pass them on to the classifiers. Their best performing classifiers were QSVM and EBT. Kerdjidj et al. in [424] use compressive sensing to explore the use of different modalities for fall detection purposes. The performance of accelerometer only and an accelerometer - gyroscope sensor combination in a fall detection scenario is compared. A Shimmer device [425] is used to capture data from 17 volunteers asked to perform activities of daily living as well as fall like movements. They first extract segments of length 2.56 seconds from their recordings before performing feature extraction and passing it on to four different machine learning classifiers, K-Nearest Neighbor, Support Vector Machines, Decision Trees and an Ensemble Classifier (EC). They conduct two such experiments, one with compressive sensing and the other without. Their results indicate that the EC classifier and SVM perform the best with the help of compressive sensing included in the pipeline. A 'transfer learning' approach towards fall detection is provided by Fanez et al. in [426] and they test it on the UMAFall [427] dataset and the FallOVI dataset (created by them). Using a finite state machine which captures windows of accelerometer measurements based on peak values, the authors convert the windowed segments into string representations using symbolic aggregate approximation (SAX) [428]. During training each user performs normal ADLs and falls for a short period of time. Peak values are determined from sensor measurements and used to create bag of words using SAX. After this, normal operation of the system starts and information retrieval (term frequency - inverse term frequency) values are used to determine what label to give to new SAX words derived from sensor values. If a determined word is not similar (using Manhattan distance) to any word in the bag of words (K-NN clustering is used to group similar words), a fall event is suggested to have occurred. 'Transfer learning' involves the use of clusters formed by other users (performing activities only) as the starting point of the training. Their experiment compared an SVM classifier with their information retrieval based scheme. They are able to detect falls all the time with the UMAfall dataset but have a high number of false positives for the FallOVI dataset. An noticeable aspect of this approach is the use of clustering and 'feature extraction' as strings. In Giuffrida et al. [429] the authors use data from the SisFall dataset and an SVM classifier to differentiate between falls and no-fall samples. They extract slices (of 1 second) from the SisFall trials and label each slice as containing a fall or not. A number of features were then computed for each slice before feature selection was performed and an SVM classifier was trained to determine the output class.

Deep Learning methods have also been widely used for fall detection purposes. Casilari et al. in [430] explore the contribution of the gyroscope sensor for use in combined accelerometer-gyroscope based fall detection systems. To do this, they make use of measurements from the SisFall dataset, extract observation windows of 5 seconds around peaks of accelerometer sensor signal values over a trace and pass them on to a Convolutional Neural Network with four convolutional layers, three max pooling layers followed by one fully connected layer for classification. Training is performed in two different sets of experiments, one including gyroscope data and the other without it. They note that the results they get while using only accelerometer signal measurements are better than that when using data from both sensors. A CNN is used for fall detection purposes by Santos et al. in [431] from accelerometer measurements. Their network consists of six convolutional layers, two max pooling layers in between followed by a fully connected layer for classification. They use data from the URFD dataset [432] and the Smart Watch and Notch datset [433] in their experiments. They note that their model performs best when they use data augmentation for both the datasets. However, this was achieved by two different variants of the same model. This highlights the point of a single model not necessarily performing well on all datasets. Another interesting aspect of their work is the use of the Mathews Correlation Coefficient to evaluate performance of their algorithms. A modified AlexNet [434] has been used in [435] by Alarifi et. al. They collect tri-axial data from inertial measurement sensors consisting of accelerometer, gyroscope and magnetometer at six different positions on a subjects body. A total of 16 activities of daily living and 20 falls were recorded by them. Feature extraction is then performed in terms of various statistical measurements as well as frequency analysis. This is followed by principal component analysis and then passed on to the classification stage consisting of an optimized AlexNet ConvNet. Waheed et. al. in [436] develop a FDS using a Bi-Directional Long Short-Term Memory (Bi-LSTM) network. They consider the binary case of fall and no fall and perform experiments using the SisFall dataset as well as the UP-Fall dataset [437]. Their network consists of eight layers in total. Two Bi-LSTM layers and two fully connected layers with dropouts being used for regularization. Training is performed with creating missing values in the data to introduce noise tolerancy. Casiliri et al. in [438] provide a performance comparison for a CNN network on multiple public fall detection datasets. The aim of their work is to test the cross-application of a similar CNN network trained on different datasets. They set their experiments up by training a four layer CNN on each dataset separately for fall detection on 14 datasets (using similar positions of sensor placement) on windows of 5 seconds for accelerometer signals. Their results indicate a very good performance of the network for the SisFall dataset, however, it doesn't perform very well for most of the other datasets, in some cases, performing less than random chance, which is quite surprising. Their experiments do however, highlight an issue with the well accepted notion that an algorithm developed on some benchmark dataset will necessarily work similarly well with other datasets. The authors suggest that the erratic performance of their method could be due to the difference in the nature of the data in terms of sampling frequency used, range of the sensor and the varying type of movements. Another cross-dataset approach for fall detection has been proposed in [439] where the authors use a combination of a CNN and Long Short Term Memory network to extract features followed by a K-Nearest Neighbor classifer to detect falls as well as identify the subjects within four fall datasets, DFNAPAS [440], SisFall, UniMiB-SHAR [441] and ASLH [442]. The network is trained using the DFNAPAS dataset, before training the network however, data augmentation is carried out for the fall class. Their best results are achieved for a value of K = 3 and they achieve good results for all experiments. They observe that using a deep learning architecture for feature extraction purposes along with a machine learning classifier performs better than a using a fully connected layer at the classification stage.

Post fall intelligence is an important research area in the field of fall detection as it can be useful in determining various post fall injuries [443] and serve as an intelligence parameter [444] for doctors. Jung et al. in [445] also target detection of pre-impact falls for wearable airbag deployment. Their method involves the application of a threshold on determined features from accelerometer and gyroscope measurement measurements. The thresholds are then applied to a dataset collected by them and also on the SisFall dataset. The threshold is determined by performing a grid search on the extracted features from their dataset to maximize specificity and sensitivity. They do note that some activties like sitting down quickly on a chair or a mattress triggered false positives. Moreover, when applied to SisFall, their methods performance is not as high as achieved by methods based on ML/DL. Koo et. al. [446] present experiments for post fall detection from a combination of self collected data and the SisFall dataset. They conduct tests using sliding windows as well as discrete windows from these signals and compute statistical features from them. After feature extraction, two different classifiers, the artificial neural network and support vector machine are tested with the computed features as well as raw sensor values. They find that both ANN and SVM are suitable for use in post fall detection scenarios. Another approach looking at the different phases of a fall has been presented in [447] where Hsieh et. al. use accelerometer sensor data to differentiate between five phases of a fall, pre-fall, free-fall, impact, resting and recovery and the initial and end static phases. To do this, they compute various time domain and statistical features and test five classifiers, SVM, K-Nearest Neighbors, Naive Bayes, Decision Trees and Adaptive Boosting (AdaBoost). For their experimental setup, the best results were achieved using the K-NN classifier. In Musci et al. [448], the authors propose a RNN based method to differentiate between falls, pre-impact falls (state where a person is in a dangerous state of transition which may result in a falls) and normal activities. Their motivation for including preimpact falls is to enable timely deployment of a fall protection system. First, they label data from the SisFall dataset to form three classes. They then extract windows of duration 1.28 seconds with an overlap of 50% from the sensor signals and pass them on to their network which consists of two fully connected layers and two-layer LSTM layers. Moreover, due to an imbalanced training set, they define a new balancing loss function. Their method results in fall detection good results for the three classes considered. A CNN-LSTM approach has been proposed in [449] by Yu et al. for detection of pre-impact fall and falls in the SisFall dataset. They provide a comparison of standalone CNN, LSTM and a combined CNN-LSTM architecture for this task and also implement them on a Jetson Nano. They define a pre-impact fall as the time interval where a subject transitions from a controlled state to a state which may lead to a fall. First, they label data in the SisFall dataset as described in [448] to form a three class problem. They achieved their best results for their CNN+LSTM model having four CNN layers followed by two LSTM layers and a fully connected layer for classification. Their approach highlights the combined capability of CNN-LSTM for feature extraction. An aspect of fall intelligence is direction, in [450] Lee et al. use the velocity vector from the acceleration sensor in a smartphone for fall detection with direction and then later on in [451], they use the standard deviation of the accemerometer and gyroscope sensor measurements from a smart phone to determine falls and fall direction. They perform a small experiment and are able to differentiate between left, right, front and backward falls. More work on fall detection with direction was performed by the authors in [452] who use an accelerometer and gyroscope combination along with a kalman filter for tilt estimation. Fall detection is then performed using an SVM classifier. Falls with direction detection has been attempted in [453, 454] by the same authors. They collect accelerometer data from various subjects and extract three features from the recordings for each axis, the mean, standard deviation and principal components (using Principal Component Analysis). These are then passed on to a SVM classifier to differentiate between a forward, backward, left and right fall and ADLs.

While direction aware fall detection is an important determination in terms of post fall intelligence, fall detection with severity is necessary since it could help provide indications to falls with immediate recovery or otherwise, as falls without immediate recovery would be more detrimental to health than a fall with immediate recovery as has been suggested in [455]. In [456] the authors attempt to classify between different falls (direction: forward, backward, right, left and intensity: hard, soft [fall on knees first, then on floor]) and five different ADLs using accelerometer data. They collected data for their experiments using a triaxial accelerometer mounted on the chest. Feature extraction consists of first concatenating acceleration values in each axis and then using a Debauchies-2 level-3 wavelet which are then sent to the classification stage. In the classification stage five different classifiers, an ANN, a Radial Basis Function (RBF) Network, Probabilistic Principal Component Analysis (PPCA) and Linear Discriminant Analysis (LDA) are used through a voting machine to determine falls. A voting machine (VM) involves multiple classifiers giving a vote towards any of the multiple classes and the sum of the vote is compared against a vote threshold to determine the event that has taken place. In their work, the authors train individual VMs for all the activity and fall types in their dataset by structuring them all in parallel and adding a comparator function at the end of the pipeline. Moreover, a K-NN multiclass classifier also feeds in to the comparator and is trained while training of the other classifier, it is provided the true value of the activity being input to the classification stage. The authors show that their dataset works very well. A valuable insight about their work is the ensembling mechanism the authors have employed to determine fall directions. In [457], Hussain et al. propose a fall detection system that can first determine falls and then the type of fall using data from the SisFall dataset. They accomplish this in a hierarchical setup where their system first considers fall detection as a binary problem, whether a fall has taken place or not, and if a fall has been detected, it classifies between the various falls in the dataset. Their system is designed to work with 10 second non-overlapping windows of accelerometer and gyroscope signals. Data from each record is first low pass filtered before two different types of feature sets, consisting of various time domain and statistical features, are computed on the data. This is then followed by the machine learning stage where three different classifiers are tested, K-NN, SVM and RFC. In the fall detection stage, statistical features are computed from ADL and fall signals and sent to the three classifiers for the preliminary binary classification. After a fall has been determined to have happened, numerous other statistical and time domain features are then computed on the data before being sent to the next stage to determine the type of fall activity taking place. In their experiments, the authors find that K-NN is most effective in differentiating between falls and ADLs where as RFC performs the best when the different fall activities need to be determined. This work highlights the usefulness of a hierarchical approach towards non-binary fall detection. One proposed method to perform combined activity recognition and fall detection has been presented in [458]. In this, Li et. al. use multi-modal sensor fusion and a Bi-LSTM classification network to differentiate between five activities of daily living and a fall. The sensors they use are an inertial sensor placed on the wrist, waist and ankle as well as a radar sensor. After pre-processing both the inertial measurement and radar signals, various statistical and moment computations were performed to be used as features. These were passed on to the multilayer Bi-LSTM network after feature selection to determine the output class. One thing to note is that both both [458] and [421] consider fall as a single category rather than considering falls as a detailed problem (direction and/or severity) in itself. More recent work by We et al. [459] also considers activity recognition and fall detection together. They use inertial measurement sensor data from two datasets, the MobiAct dataset [460] and the SmartFall dataset [433]. The MobiAct dataset contains data from four falls and nine activities of daily living whereas the SmartFall dataset has non-fall and fall recordings. In their experimentation, they compare the performance of different machine learning models and several deep learning models, including a CNN, LSTM, CNN-LSTM combination and Gated Recurrent Units. The machine learning models are trained by computing time and frequency domain features whereas the deep learning models are trained using raw sensor data. They find that the GRU designed by them consisting of two GRU units followed by a softmax classification layer is the best performing model. Another deep learning approach utilizing sequential modeling for a fall detection system that also considers ADLs has been presented by Sengül et al. [461]. They collect their own data for two types of falls and four activities of daily living. After data augmentation on the minority classes, they use a Bi-LSTM for classification. Le et al. [462] propose a non-binary fall detection scheme utilizing a collection of time, frequency frequency domain features in addition to the three Hjorth parameters of activity, mobility and complexity. They use data from two datasets, the MobiAct dataset and the UP-Fall dataset. They are able to achieve good results on both datasets using all three feature types with a random forest classifier.

The use of attention based models has also been finding increasing usage in fall detection research. Yhdego [463] use an attention model to perform binary fall detection. The fall dataset is collected by them and after windowing, the authors use time2vec positional encoding on the data. Their network consists of three attention blocks each consisting of multiheaded self attention followed by normalization and a fully connected network. After the attention blocks follow two fully connected layers. They find that their network incorporating attention performs well in differentiating between fall and non-fall samples. Another interesting approach using attention is presented by Liu et al. [464]. Inorder to apply attention on both temporal (step wise) and spatial (channel wise) aspects of the signals, a gated scheme is suggested. A fully connected layer is used to determine the embeddings before positional information is added to the data. Positional encoding is only used for stepwise attention as channel wise positions are not useful. Self attention is performed in the step-wise part of the network and is carried out for the individual channels in a pairwise fashion across all time steps while the channel wise attention is performed across the different channels across all time-steps. The outputs of these two units are combined through a fully connected network which weighs them before concatenation and classification.

2.7 Fall detection Datasets

There are several datasets published for use in algorithm development for fall detection as has been noted in [438]. Of these, for this case study, we choose two datasets, these are the SisFall and the K-Fall dataset. The SisFall [417] dataset is chosen as it contains recordings of elderly people which are most at risk from injury due to falls and therefore should provide a good representation of activities of the elderly. It should be noted that its one of the most used dataset for fall detection purposes [438]. The other dataset used is the K-Fall [465] dataset. It was released on the pattern of the SisFall dataset and includes more activities compared to SisFall. A summary of the datasets is presented in the next section.

2.7.1 SisFall Dataset

The SisFall dataset consists of accelerometer and gyroscope recordings of 19 types of activities of daily living and falls. Two accelerometers and one gyroscope were placed on the waist and used for making the measurements. This location was chosen so as to ensure that all body movements while performing the activities/falls were captured by the sensory system. Furthermore, the data recordings involved 23 young people between the ages of 19-30 years old and 15 elderly people between the ages of 60-75 years. The dataset contains annotated sensor measurements of each of these ADLs and falls recorded as well as video recordings of sample experiments. There are four main types of falls present, six of them are forward falls , three backward falls, four lateral falls and two vertical falls. The types of falls performed were directed through a survey taken from elderly people living independently as well as the ones living in retirement homes and include scenarios of slipping and tripping. Moreover, they are preceded by various types of ADLs being performed to make the recordings as close to a real world scenario as possible. The ADLs recorded were approved by medical personnel and were chosen so as to be similar to falls. The ADLs include high mobility activities such as walking up/down stairs, jogging as well as activities which can be confused with falls such as quickly sitting in a chair, bending at the knee and stumble while walking etc. In total, there were 2706 ADL recordings and 1798 fall recordings. The SisFall data is utilized in this study as it has been the the dataset of choice for several previous research approaches addressing the subject of fall detection [466, 467, 468, 436, 469] and also because the volunteer makeup consists of both men and women, young adults and the elderly. The ADLs and Falls in the SisFall dataset have been presented in Table 32 with a brief description of the activities and falls provided as well.

Table 32. ADL and Falls present in the SisFall dataset.

Activity/ Fall Code	Brief Description
D01	Walking slowly
D02	Walking quickly
D03	Jogging slowly
D04	Jogging quickly
D05	Walking upstairs and downstairs slowly
D06	Walking upstairs and downstairs quickly
D07	Slowly sit in a half height chair, wait a moment, and up slowly
D08	Quickly sit in a half height chair, wait a moment, and up quickly
D09	Slowly sit in a low height chair, wait a moment, and up slowly
D10	Quickly sit in a low height chair, wait a moment, and up quickly
D11	Sitting a moment, trying to get up, and collapse into a chair
D12	Sitting a moment, lying slowly, wait a moment, and sit again
D13	Sitting a moment, lying quickly, wait a moment, and sit again
D14	Being on one's back change to lateral position, wait a moment, and change to one's back
D15	Standing, slowly bending at knees, and getting up
D16	Standing, slowly bending without bending knees, and getting up
D17	Standing, get into a car, remain seated and get out of the car
D18	Stumble while walking
D19	Gently jump without falling (trying to reach a high object)
F01	Fall forward while walking caused by a slip
F02	Fall backward while walking caused by a slip
F03	Lateral fall while walking caused by a slip
F04	Fall forward while walking caused by a trip
F05	Fall forward while jogging caused by a trip
F06	Vertical fall while walking caused by fainting
F07	Fall while walking, with use of hands in a table to dampen fall, caused by fainting
F08	Fall forward when trying to get up
F09	Lateral fall when trying to get up
F10	Fall forward when trying to sit down
F11	Fall backward when trying to sit down
F12	Lateral fall when trying to sit down
F13	Fall forward while sitting, caused by fainting or falling asleep
F14	Fall backward while sitting, caused by fainting or falling asleep
F15	Lateral fall while sitting, caused by fainting or falling asleep

2.7.2 K-Fall Dataset

The K-Fall dataset has been developed based on the SisFall dataset and contains 15 fall types and 21 ADLs. The falls in K-Fall are the same as those in SisFall. However, in the case of ADLs, they remove the activity of sitting in and getting out of a car and combine some of the activities while adding static activities of sitting on a chair, sitting on a sofa and lying down. A total of 2729 ADL recordings and 2346 fall recordings are present in the dataset. This dataset is included in the study to test cross-dataset fall detection. The details of the K-Fall dataset are presented in Table 33.

2.8 Experiments

This section discusses the experiments conducted for the purposes of fall detection. Four experiments have been performed in this regard, first considering fall data only

Table 33. ADL and Falls present in the K-Fall dataset.

Activity/ Fall Code	**Brief Description**
D01	Stand for 30 seconds
D02	Stand, slowly bend the back with or without bending at knees, tie shoe lace, and get up
D03	Pick up an object from the floor
D04	Gently jump (try to reach an object)
D05	Stand, sit to the ground, wait a moment, and get up with normal speed
D06	Walk normally with turn (4m)
D07	Walk quickly with turn (4m)
D08	Jog normally with turn (4m)
D09	Jog quickly with turn (4m)
D10	Stumble while walking
D11	Sit on a chair for 30 seconds
D12	Sit on the sofa (back is inclined to the support) for 30 seconds
D13	Sit down to a chair normally, and get up from a chair normally
D14	Sit down to a chair quickly, and get up from a chair quickly
D15	Sit a moment, trying to get up, and collapse into a chair
D16	Stand, sit on the sofa (back is inclined to the support), and get up normally
D17	Lie on the bed for 30 seconds
D18	Sit a moment, lie down to the bed normally, and get up normally
D19	Sit a moment, lie down to the bed quickly, and get up quickly
D20	Walk upstairs and downstairs normally (5 steps)
D21	Walk upstairs and downstairs quickly (5 steps)
F01	Forward fall when trying to sit down
F02	Backward fall when trying to sit down
F03	Lateral fall when trying to sit down
F04	Forward fall when trying to get up
F05	Lateral fall when trying to get up
F06	Forward fall while sitting, caused by fainting
F07	Lateral fall while sitting, caused by fainting
F08	Backward fall while sitting, caused by fainting
F09	Vertical (forward) fall while walking caused by fainting
F10	Fall while walking, use of hands to dampen fall, caused by fainting
F11	Forward fall while walking caused by a trip
F12	Forward fall while jogging caused by a trip
F13	Forward fall while walking caused by a slip
F14	Lateral fall while walking caused by a slip
F15	Backward fall while walking caused by a slip

and aiming to determine falls with direction and severity. The second, third and four experiments involved the inclusion of activities of daily living classification in addition to the fall classes considered in the first experiment. Moreover, experiment four presents results for a cross-dataset fall detection scenario.

2.8.1 Fall Detection with Severity and Direction consideration

This section describes the work on fall detection considering severity and direction while only considering sensor data for falls. The experiment has been performed as two tasks, one for determining falls considering direction only and the other while considering both direction as well as severity.

2.8.1.1 Data Labeling

To perform fall detection that is direction and severity aware, we only considered fall data from the SisFall dataset. As can be seen from Table 32, most of the falls in the dataset have been labelled as either being in the forward, backward or lateral direction. However, two of the falls (*F06* and *F07*) are not labeled in terms of direction. For the considerations of this research work, these were assigned the labels of *Forward* and *Lateral* respectively using the videos of sample trials provided by the dataset authors.

Concerning fall severity, while the original labels from the dataset contained information for most falls for direction, the approach followed by Gibson et al. [456] was used to determine the severity of falls. According to the practice followed by them, all falls where in some support was used to soften the impact of the fall were considered as soft falls whereas all falls where the subject fell directly were classified as hard falls. This resulted in six classes for fall types with hard and soft for imapact and forward, backward and lateral for direction. These are Forward Soft Falls (FSF), Forward Hard Falls (FHF), Backward Soft Falls (BSF), Backward Hard Falls (BHF), Lateral Soft Falls (LSF) and Lateral Hard Falls (LHF). These labels have been summarized in Table 34.

Table 34. Labeling used for Fall only classification for the SisFall Dataset

Experiment Name	**Fall Code**	**Assigned Fall Name**	**Assigned Fall Label**
Direction	F01	Forward Fall	FF
	F02	Backward Fall	BF
	F03	Lateral Fall	LF
	F04	Forward Fall	FF
	F05	Forward Fall	FF
	F06	Forward Fall	FF
	F07	Lateral Fall	LF
	F08	Forward Fall	FF
	F09	Lateral Fall	LF
	F10	Forward Fall	FF
	F11	Backward Fall	BF
	F12	Lateral Fall	LF
	F13	Forward Fall	FF
	F14	Backward Fall	BF
	F15	Lateral Fall	LF
Direction + Severity	F01	Forward Hard Fall	FHF
	F02	Backward Hard Fall	BHF
	F03	Lateral Hard Fall	LHF
	F04	Forward Hard Fall	FHF
	F05	Forward Hard Fall	FHF
	F06	Forward Soft Fall	FSF
	F07	Lateral Soft Fall	LSF
	F08	Forward Soft Fall	FSF
	F09	Lateral Soft Fall	LSF
	F10	Forward Soft Fall	FSF
	F11	Backward Soft Fall	BSF
	F12	Lateral Soft Fall	LSF
	F13	Forward Soft Fall	FSF
	F14	Backward Soft Fall	BSF
	F15	Lateral Soft Fall	LSF

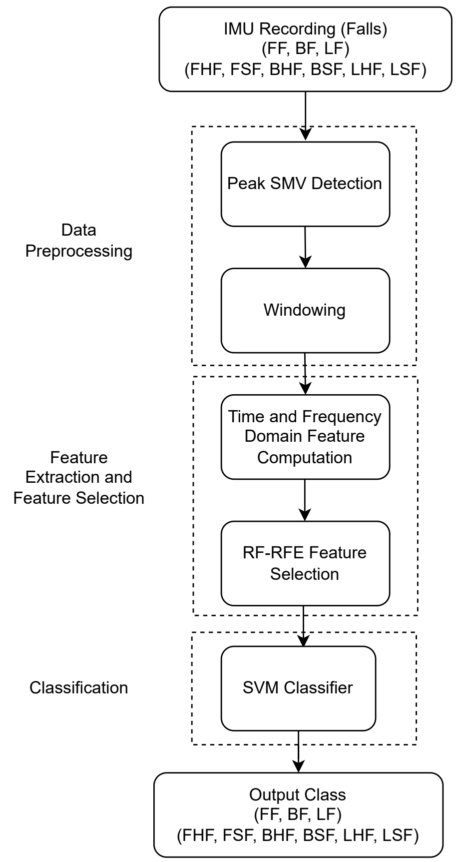

Figure 17. Fall Detection with Direction and Severity.

2.8.1.2 Methodology The methodology used for this experiment follows a typical machine learning pipeline as shown in Figure 17. First, we pre-process the data which involves extraction of sensor segments representing falls. This is followed by feature extraction that aims to extract useful representations from the accelerometer and gyroscope sensor data. Lastly, feature selection is carried out where we aim to reduce the number of features used in the last stage that is classification.

Data Pre-processing

Recordings in the SisFall dataset vary in length between 12 and 100 seconds. The trial recordings for falls in the SisFall dataset consist of subjects performing various activities before experiencing the fall. For a sound feature extraction process, we extract equal duration segments of sensor readings from these trials that represent the fall taking place along with some part of the pre-fall activity being performed. To do this, for each record we first calculate the acceleration magnitude (also called the Signal Magnitude Vector (SMV)) [430] for all sensor value samples within a trial recording. The SMV can be calculated as given in Eq.1.

$$SMV_j = \sqrt{\left|A_{x_j}\right|^2 + \left|A_{y_j}\right|^2 + \left|A_{z_j}\right|^2} \tag{1}$$

where SMV_j represents the SMV value for a sample j in a given trial. The SMV is calculated for all samples and the sample location for the SMV peak value is determined within the trial. This is then used as the midpoint for extracting a 2.5 second segment from the trial. A segment length of 2.5 seconds was chosen as it was visually found to capture all the falls as well as some part of the pre-fall activity being performed. Using this scheme, segments were extracted for all the fall trials in the dataset.

There are a total of 1798 fall samples in the dataset, after the labeling used, there were a total of 838 samples for the forward fall category, 360 samples for the backward fall category and 600 samples for the lateral fall category. Sample waveforms for the three directions of falls and three severity levels have been shown in Appendix A.

Feature Extraction

Feature extraction is used to convert inputs in to more useful representations. In this experiment, we take cue from the work of [420, 470] who use various time and frequency domain features successfully for fall detection and recognition of activities of daily living. Table 35 lists the features computed for each extracted segment. Each of these features have been computed for every axis of the considered accelerometer and gyroscope sensor measurements.

Table 35. Features computed for each fall segment

Domain	Features Computed
Time	25th Percentile 75th Percentile Delta Interquartile range Kurtosis Mean Median Maximum Minimum Skewness Standard Deviation Variance
Frequency	Power Spectral Entropy Power Spectral Density Mean Power Spectral Density Median Power Spectral Density RMS

Percentiles (25th Percentile, 75th Percentile) and Interquartile Range

For a set of numerical values X ordered in arranged ascending order, the i^{th} percentile is defined as the number n below which i percent of the total numbers of X fall below it. Therefore, the 25$^{\text{th}}$ Percentile (also called the First Quartile) is the number in X below which exactly 25% of the values fall. Similarly, 75$^{\text{th}}$ Percentile (also called the Third Quartile) is the number in X below which exactly 75% of the values fall. Another important quantity concerning percentiles is the Interquartile Range (IQ

Range), also called the Midspread. The IQ Range is the difference between the 75th Percentiles and the 25th Percentiles.

Delta The Delta represents the difference between the minimum and maximum value of a set of numeric values X.

Kurtosis It's a metric for how much a distribution's tails diverge from that of a normal distribution. A large Kurtosis values indicates to larger extremity of the divergences which can be thought of as outliers. The Kurtosis for a set of numerical values X can be calculated as given in Eq. 2.

$$Kurtosis(X) = \frac{1}{N\sigma^4}\sum_{i=1}^{N}(x_i - \mu)^4 \tag{2}$$

Mean For a set of values X, the arithmetic mean or the average returns the center value of X. Mathematically it is given in Eq. 3.

$$Mean(X) = \frac{\sum_{i=1}^{N} x_i}{N} \tag{3}$$

Median For a set of values X, the median indicates to the central tendency of X. It divides a set of values in to two equal parts. For a set X of size N arranged in ascending order, the median can be calculated as in Eq. 4.

$$Median(X) = X[\frac{N+1}{2}] \tag{4}$$

Maximum For a set of values X, the maximum value represents the largest value in the set X.

Minimum For a set of values X, the minimum value represents the smallest value in the set X.

Skewness The Skewness measures the lack of symmetry in the probability distribution of data around its mean. It is calculated as in Eq. 5.

$$Skewness(X) = \frac{1}{N\sigma^3}\sum_{i=1}^{N}(x_i - \mu)^3 \tag{5}$$

Standard Deviation and Variance The Standard Deviation measures the variation of a set of numerical values around its mean. For a set of numerical values X, the standard deviation can be calculated as in Eq. 6.

$$Std(X) = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}} \tag{6}$$

where μ is the Mean of X. The Variance of X is the square of the Standard Deviation.

Power Spectral Density The Power Spectral Density (PSD) is an indication to the power content of a signal with respect to its frequency. It therefore helps to understand the distribution of power in the signal for the different frequencies that comprise it. The PSD has been computed using the Welch method [471]. Once the PSD has been computed, the mean, median and Root Mean Squared (RMS) values of the PSD are computed.

Power Spectral Entropy The Power Spectral Entropy (PSE) is a measure of the entropy in the power spectrum of a signal and indicates to the complexity present therein. It is computed as an entropy calculation on the normalized PSD estimate. Similar to the PSD computation, the Welch method was used to estimate the PSD before PSE calculations were performed.

The aforementioned quantities were computed for all windowed samples of the fall recordings. In total, this resulted in 96 features being computed for each segment. Before these features are passed on to the next stage, normalization was performed feature wise for each sensor.

Feature Selection

Feature selection is an important step in the machine learning pipeline. It not only helps to reduce the feature set size used in an application which helps computationally but may also result in performance improvement as observed by [472]. Feature selection is performed using Random Forest Recursive Feature Elimination (RF-RFE) [473] which involves determining the importance of features by eliminating features iteratively and looking at classifier performance. It starts by developing a model using all the available features within the dataset, in every iteration, the feature that has the 'least importance/contribution' is discarded and a new model is fitted with the remaining features. This process is carried out until a pre-decided number of features remain. Since in this work, our goal is to reduce the number of features from the original count, we perform manual tests by establishing a baseline for the performance of our models. The baseline is formed by using all features for each of our classification tasks and using the highest weighted F1-score achieved as the performance goal cut-off for the reduction of features. This step will ensure that we atleast get the same performance as the original feature set.

Classification

Classification refers to the usage of a classification algorithm, also called a classifier to determine the output 'label' or category given a set of input values called features. In the current setting, the classification problem is a supervised one wherein the classifier is provided inputs and their corresponding output labels which allow it to learn the relationship between the input and the output. Once this mapping has been sufficiently learned (a process called training) the trained classifier can be provided unlabeled inputs for it to predict an output label value or class. The efficacy of the classifier in predicting the correct output is measured through various metrics such as accuracy, precision, recall, F1-score etc. This experiment consists of two classification experiments being performed, one is the direction only experiment where the classifiers used are trained to distinguish between the three direction of falls given the input feature set and another experiment where the classifiers are trained to determine fall directions and severity together.

Four different classifiers were tested to perform these tasks, these are the Random Forest Classifier, Support Vector Machines, Decision Trees and Extreme Gradient Boost (XGBoost). All of the classifiers were trained using five fold cross validation with a stratified split and a parameter search being performed for tuning purposes.

Support Vector Machines Support Vector Machine is a supervised learning algorithm that, in its simplest form tries to fit a line in between data samples of two different classes to separate them. The criteria used by the SVM to fit this line is the maximization of the distance between the line and the (data) points closest to it. This concept of using a line to separate two dimensional data can be extended to two dimensional planes being used for separating three-dimensional data and hyperplanes for higher dimensional data. SVMs are quite flexible in that they can not only be used to work with linearly separable data but also with data that is not linearly separable. This is achieved using Kernel functions which map the nonlinearly distributed data in to a higher dimensional space to make it linearly separable (as much as possible). There are several kernel functions available to be used for this purpose such as the hyperbolic tangent, radial basis, sigmoid, polynomial etc. Optimizing the SVM requires the tuning of the cost parameter C, a small value of C indicates an underfit whereas a very large value of C indicates an overfit.

Even though, the current experiment considers using SVMs for a classification taks, it should also be mentioned that SVM's are not limited to application towards classification problems and they have also been used for regression applications too.

Decision Tree Decision Trees are one of the most widely used classification algorithms due to their interpretability. The aim of the algorithm is to come up with as simple a solution as possible while successively dividing the data in to simpler and simpler classification tasks based on the values of the features. For a classification problem, given a set of input features of the training set, decision trees use the values of the features to divide or split the dataset in to more simpler sub problems. The aim being to get the largest information gain when choosing the feature/s for the split. Each split point in a decision tree is called a node and each split is called a branch. Branches from a given node represent all the possible outcomes from that node and may contain subsequent decision nodes as well which are called child nodes. Each branch ends at a node called the leaf node where data points are assigned a class label.

Depending on the number of input features and the output classes, decision trees can be built to classify the dataset in a 'complete' manner due to its iterative partitioning scheme. However, this is undesirable since that would lead to overfitting of the model and also because such trees would lead to a complicated model, which is not desirable since that would impact interpretability. The process of removing branches and replacing them by leaf nodes is called pruning. Pruning aims to increase the models generalizations capabilities and increase robustness. Two paramters which are used for this purpose are tree depth limits the number of child nodes present in every branch and the number of features used at each split.

Random Forest Classifier Random forest is a bagging approach towards decision trees where instead of deciding on where to split for each node, for each tree a random subset of features is used and the feature/threshold to split the data is determined within that subset of features to separate the data in the best manner possible. This process is repeated multiple times which results in various trees producing possibly different predictions. The predictions from each tree are combined to form an ensemble either through majority voting or by averaging. The implementation of Random Forests used in this work uses the averaging approach. There are several parameters that are important in the tuning of Random Forests are the number of trees created before the averaging process starts, the number of levels in each tree and more.

Extreme Gradient Boost Extreme Gradient Boost is another ensemble based tree algorithm, however compared to the RFC, XGBoost uses boosting instead of bagging. In boosting, the input samples are uniformly weighted and first a 'weak' learner (classifier) is created for a given task that doesnt perform well but is sufficient in the sense that it is better than random guessing. In subsequent iterations of the learner, input samples which werent classified correctly by the learner are reweighted by assigning larger weights to them while the weights for correctly classified inputs is reduced. This step is repeated for a number of times with different weak learners being trained. The decision of the weak learners are combined through majority voting to determine the output class. Parameters used to tune XGBoost are the maximum depth allowed in each tree, the weight settings etc.

2.8.1.3 Results

As mentioned previously, two separate experiments were conducted, first was to differentiate between fall directions only and the other to differentiate between fall direction as well as severity. The results from those experiments are presented here. The metrics considered for judging model quality is the F1-score. The F1-score is the harmonic mean of the precision and recall and is given as in Eq. 7.

$$F1\ Score = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{7}$$

where Precision is the ratio of the True positive samples divided by the sum of the True positive and False positive samples as given in Eq. 8. It is also called positive predictive value as it indicates to the capability of the classifier to identify correctly the samples of a given class.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{8}$$

and recall indicates to the models capability to identify the samples of a given class over the whole dataset. It is also called the sensitivity and is the ratio of the True positives to the sum of the True positives and False Negatives as shown in Eq. 9.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{9}$$

We report on the F1-scores for each class when discussing classification performance and deciding on the best classifier. Moreover, for the best classifier identified the precision and recall have also been provided.

Fall direction

In this experiment, fall segments labeled for direction only were used as input to the classifiers. To establish a baseline to be used as a reference in the feature selection process, all computed features (96 in total) were passed to the four classifiers. The best performing model was found to be the SVM with a weighted F1-score of 90.4%. Feature selection was then performed keeping in view baseline performance of the SVM classifier so as to retain or improve this performance baseline when choosing a given number of 'top' features. Following this rule, feature selection resulted in 90 features. The results are given in Table 36. It can be observed from the table that the SVM has outperformed the other algorithms in detecting forward and lateral falls whereas it achieves a slightly lower F1-score for the backward fall class. Overall the SVM has shown to be better at discriminating falls with direction.

Table 36. Fall F1-scores (Fall direction only)

Fall	**Classifier (F1-score %)**			
	SVM	**DT**	**RFC**	**XGBoost**
BF	96.05	92.22	96.59	**96.63**
FF	**92.67**	83.60	90.10	89.49
LF	**88.67**	71.78	85.63	84.35
Average	*92.46*	*82.53*	*90.77*	*90.15*

Fall direction and severity

In this experiment, fall segments from the dataset were labeled for direction and severity based on the labeling shown in Table 34 and passed to each of the classifiers after feature selection. Like before, to establish a baseline, all computed features (96 in total) were passed to the four classifiers. The best weighted F1-score of 78.44% was achieved by the SVM classifier. We used this value as the baseline we aim to achieve after feature selection. With this condition the number of features were reduced to 93. The performance achieved for each of the four tested classifers is summarized in Table 37. It can be observed that the SVM is the best performing classifier for all fall types bar *FHF* where the random forest classifer achieves a slightly higher F1-score. The highest F1-score overall is achieved for the class *BSF* where as the lowest has been achieved for *LHF*. When looking at fall direction, falls in the forward direction can be seen to be the hardest to identify followed by lateral falls. No such pattern can be noted for soft/hard falls.

Table 37. Fall F1-scores (Fall severity and direction)

Fall	**Classifier (F1 Score %)**			
[]	**SVM**	**DT**	**RFC**	**XGBoost**
BHF	**87.27**	50.00	69.39	66.67
BSF	**95.08**	77.88	88.33	85.47
FHF	72.43	54.02	**75.56**	67.04
FSF	**73.25**	50.60	67.57	58.93
LHF	**66.67**	29.85	35.90	50.00
LSF	**84.65**	62.24	75.17	75.69
Average	*79.89*	*54.10*	*68.65*	*67.30*

2.8.1.4 Discussion

To summarize the results from both experiments, we present the individual F1-scores, Precision and Recall for the best performing classifier in each case. These have been presented in Table 38 and Table 39.

Table 38. Best Results: Fall direction

Fall	**Precision (%)**	**Recall (%)**	**F1-score (%)**
BF	94.44	97.70	96.05
FF	93.33	92.02	92.67
LF	88.67	88.67	88.67
Average	*92.16*	*92.80*	*92.46*

In both cases the best performing classifier was the SVM. This is inline with the findings of other research work as found from the literature review [424, 429]. A surprising outcome during this work was the unsatisfactory performance of tree based algorithms contrary to [420], we attribute this to the difference in the data segmentation scheme and also the fact that we follow different labelling for the data

Table 39. Best Results: Fall direction and Severity

Fall	Precision (%)	Recall (%)	F1-score (%)
BHF	80.00	96.00	87.27
BSF	96.67	93.55	95.08
FHF	74.44	70.53	72.43
FSF	74.17	72.36	73.25
LHF	60.00	75.00	66.67
LSF	85.00	84.30	84.65
Average	*78.38*	*81.96*	*79.89*

in terms of a direction and severity aware system. Moreover, the final weighted F1-scores for the fall direction experiment was 92% and for the fall direction and severity was 79.53% which resulted in an improvement of just above 1% in both cases with a small reduction in the number of features utilized, 90 and 93 respectively compared to 96 in the original feature set. Commenting on the direction, the hardest to predict fall was in the lateral direction and the classifier incorrectly predicted forward falls and lateral falls in both the scenarios tested. This highlights the difficulty in capturing the difference between these two fall directions using the features considered.

2.8.2 Fall Detection with Severity and Direction along with ADL consideration using Wavelet Pooling and K-NN

This section provides a framework for a fall and activity recognition system. It aims to determine an appropriate sensor modality to use and the window size to be used for the task. The framework does this as a problem of differentiating between various activities of daily living as well as various types of falls with regard to fall datection being direction and severity aware. To do this, data from the SisFall dataset is used and after suitable pre-processing and feature extraction, machine learning algorithms are utilized to differentiate between different activities of daily living and falls

2.8.2.1 Data Labeling

Since we aim to perform activity recognition and fall detection with direction and severity, the ADL labeling of the original dataset has been modified. This labeling has been shown in Table 40. As can be observed from Table 40, the activities Walking (W), Jogging (J), Sitting (S) and Standing (SB) have been considered for this work which are typical activities in ADL detection problems. Each of these labels includes data from multiple original activities, for e.g. activities with original labels of walking upstairs and downstairs, walking slowly and walking quickly have been considered as walking in this work. A similar scheme has been used for the other three activity labels as well. Some of the activities such as being on one's back change to lateral position, wait a moment, and change to one's back (D14), getting in and out of the car (D17), stumble while walking (D18), and gently jumping without falling while trying to reach a high object (D19) have not been considered. The reason for this is that they have very few samples to be considered as standalone activities (only one type of sub-activity and also because most of these are not considered in typical

ADL detection scenarios). For falls, the labeling same as that for the direction and severity experiment has been retained.

Table 40. ADL Labels used for SisFall Recordings.

Activity Code	Assigned Activity Name	Assigned Activity Label
D01	Walking	W
D02	Walking	W
D03	Jogging	J
D04	Jogging	J
D05	Walking	W
D06	Walking	W
D07	Sit	S
D08	Sit	S
D09	Sit	S
D10	Sit	S
D11	Sit	S
D12	Sit	S
D13	Sit	S
D14	-	-
D15	Standing	SB
D16	Standing	SB
D17	-	-
D18	-	-
D19	-	-

2.8.2.2 Methodology

Figure 18 shows the methodology for this work with individual parts being elaborated upon in the proceeding subsections. As can be seen, the first stage consists of data preprocessing, followed by feature extraction and then evaluation or classification.

Data Preprocessing
Data preprocessing for this experiment was performed in a similar manner to section 2.8.1 for most recordings. The SMV was computed for the complete recording and windowed segments of a fixed duration were extracted. Windowed segments are extracted in this manner from all considered activities in the SisFall dataset except the activities of *D01, D02, D03* and *D04* which consist of a single trial per subject of duration 100s. In such cases, continuous windowed segments of duration n seconds are extracted from the recordings. It is also pertinent to mention here that since both accelerometers are placed at the same position, we only consider one of the accelerometers along with the gyroscope readings present in the recorded trials. To determine the value of n as well as the appropriate sensor modality to use for the final system, experiments were performed on the developed framework and the results have been discussed in subsequent sections.

Feature extraction
In this experiment, feature extraction consists of two steps. The first is the use of

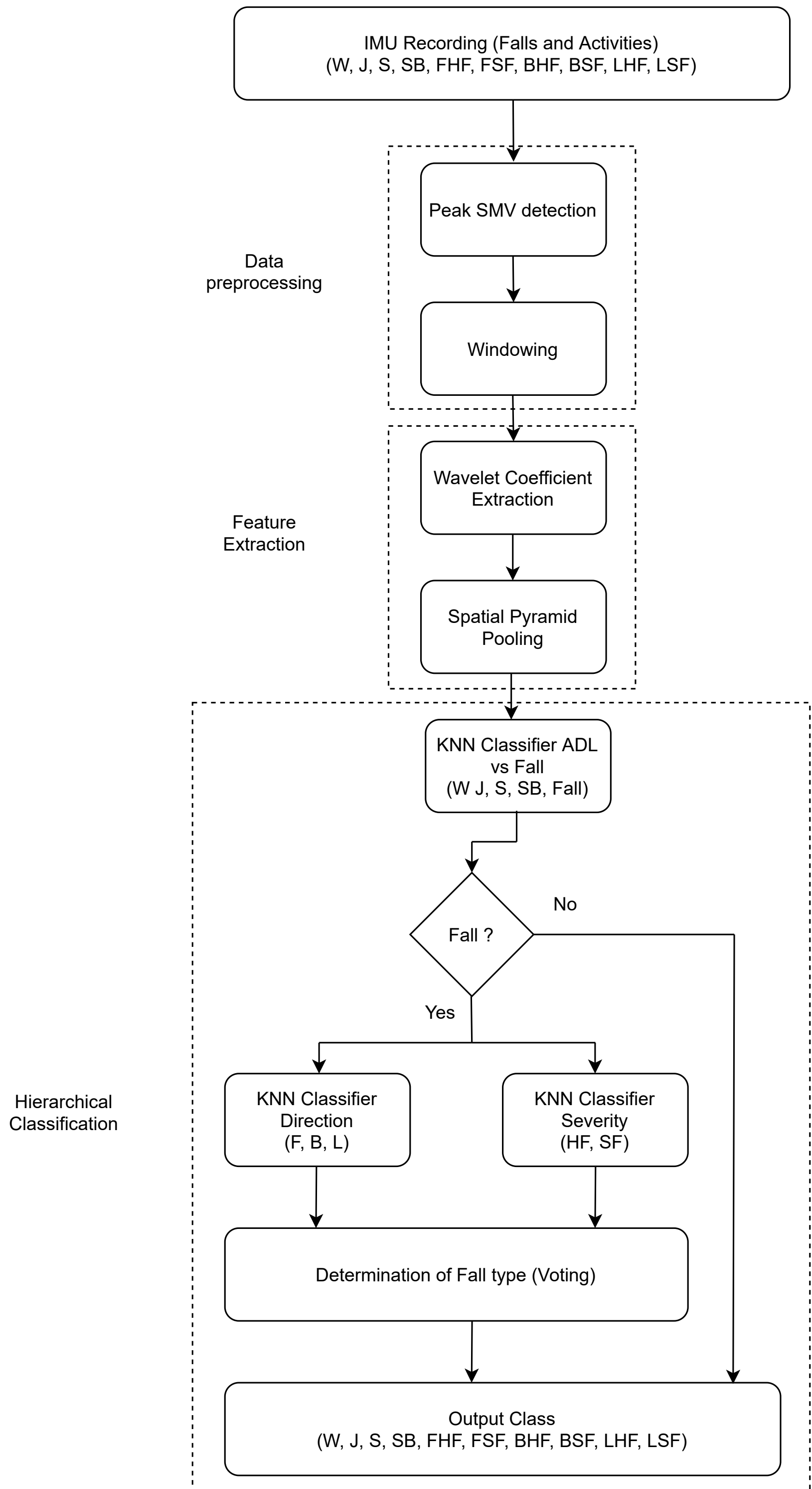

Figure 18. Hierarchical classification scheme for ADL and Fall detection.

wavelets [474] for performing wavelet decomposition and then performing pooling, the inspiration for this type of feature extraction came from the work of [475].

Wavelet Decomposition Wavelets can be used for a variety of applications such as spectral analysis where they are used to analye a signal in the time-frequency domain, denoising, compression and signal enhancement. A wavelet is a function of finite duration and has an average value of zero, an example of a mexican hat wavelet is shown in Figure. 19. The width of the wavelet is called its scale and it is analogous to frequency in an inverse manner. A large value of the scale (wider window) allows for capturing of low frequency components of the signal whereas a small scale value captures low frequency components.

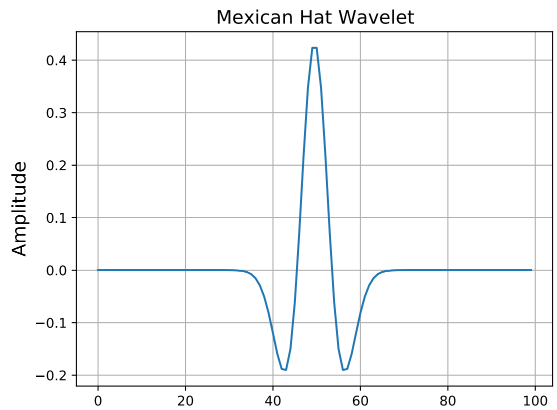

Figure 19. Sample Wavelet (Mexican Hat).

Wavelets are convolved with an input signal to produce two sets of coefficients. These are called the detail and approximation coefficients. The detail component represents the high frequency component captured by the wavelet whereas the approximation component represents the low frequency part. Since the wavelet has only been convolved once with the signal, these coefficients are said to be level 1 coefficients. At each level, resulting coefficients are downsampled by half to result in a lower resolution signal. Subsequent approximation components can be convolved with the wavelet to result in approximation and detail components for the next level thus forming what is called a wavelet decomposition tree.

To perform wavelet decomposition, in this experiment, we make use of the Haar wavelet [476] shown in Figure. 20. Level 4 haar wavelets were computed for each of the extracted segments of the signals.

Spatial Pooling The results from the decomposition provide a large number of features which would pose a problem for use in the proceeding classification stage. In order to reduce the feature dimension size, Spatial Pyramid Pooling [477] was used. Spatial Pyramid Pooling is an adaptive pooling method which was developed to address the issue of fluctuating input sizes in CNNs for image-based applications, and it entails converting varying-size convolutional feature maps into fixed-length summarizations. These summarizations, having uniform length can then be passed on to the fully connected parts of the CNN where a fixed length input is necessary.

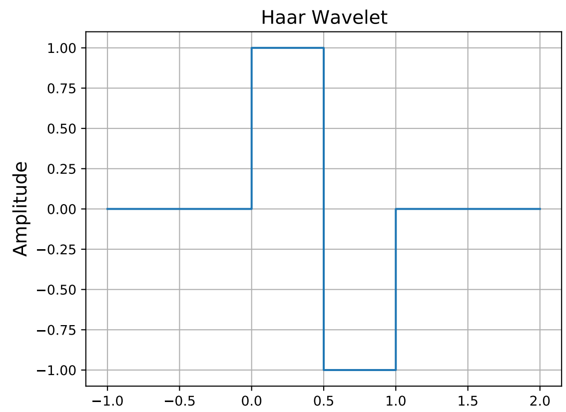

Figure 20. Haar Wavelet.

Given a pooling size *pxp*, adaptive pooling works by dividing the input in to *pxp* pieces while computing the size of each piece automatically and performing any necessary padding. Once these pieces are created, a pooling operation is typically performed (max pooling or average pooling for e.g.) on each of these pieces to summarize the input into an output of fixed size *pxp*. This results in a fixed output length for any size of the input. In this experiment, 4-2-1 1D Spatial Pyramid pooling was used on the detail and approximation coefficients of the wavelet decomposition of activity and fall data. This process has been shown in Figure 21. Each coefficient set was divided in to four and two parts and then max pooling was used to determine the maximum value in these divided parts and the coefficient set as a whole. These maximum values were then concatenated together to form the seven valued output from that coefficient set. Furthermore, the results for each coefficient set within each axis were also concatenated to form the feature vector for a sensor axis measurement. This operation was performed for each axis of accelerometer and groscope sensor data with the final feature vector of 210 values consisting of the concatenations of the individual vectors for each axis. It is hypothesized that this way local as well as global information at each level of the wavelet coefficients can be captured. For the wavelet decomposition, tests were performed with level values of 2, 3, 4, 5 and 7 and it was determined that level-4 produced the best results.

Classification

A hierarchical classification approach is employed to discriminate between the various activities and falls considered from the SisFall dataset. Hierarchical classification involves the division of a complex taxonomic classification problem in to a set of subsets that are potentially easier to differentiate as the task becomes more localized. Hierarchical classifiers have been used in multiple different applications [478] where they have been found to improve upon the performance of many flat classification schemes. The classification framework used in this work combines hierarchical classification with a vote based system. The classification problem is divided into three parts, each with its own classifier to indicate to the subclass of the output. The clas-

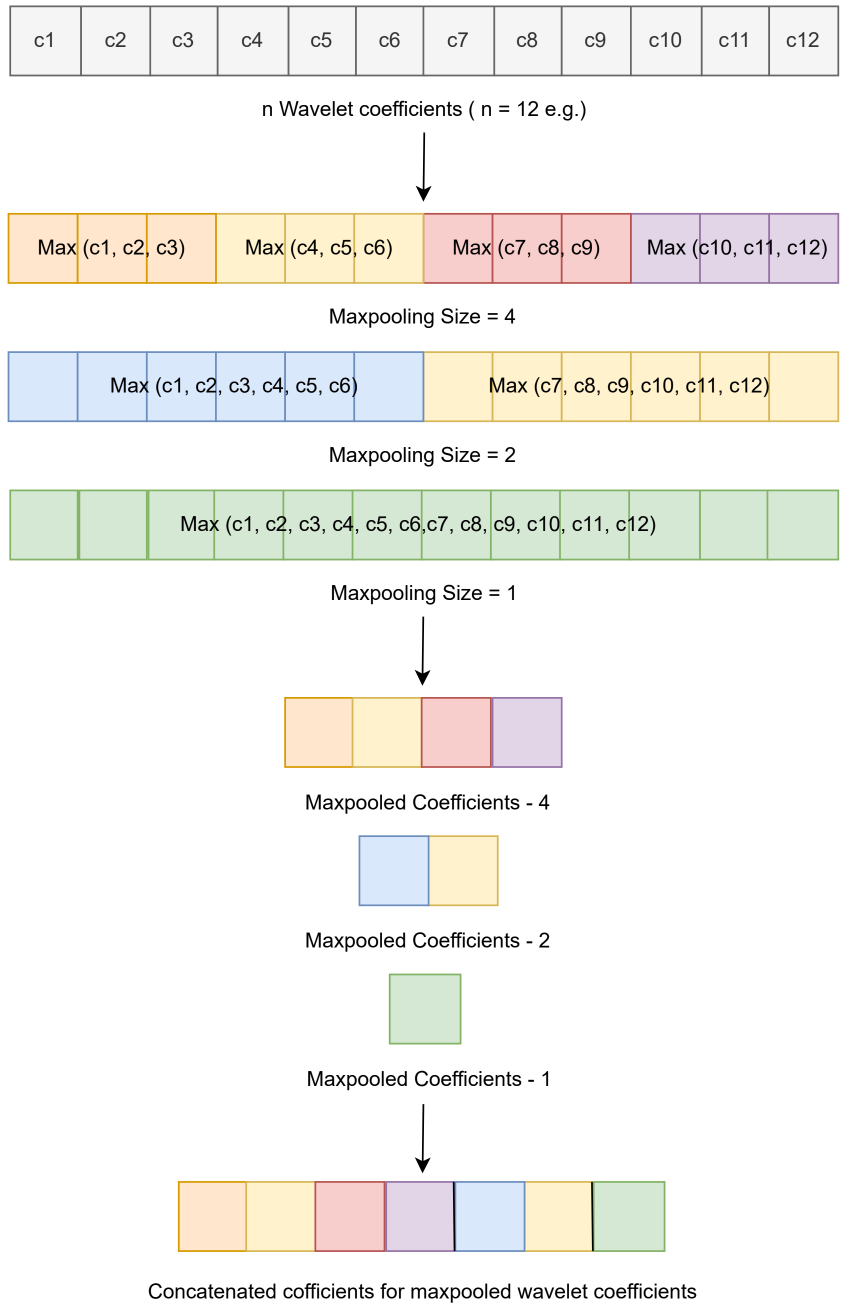

Figure 21. Example: 4-2-1 1-D Spatial Pyramid Pooling.

sifier in part one consists of differentiating whether a given recording is a fall or one of the four considered ADLs. In order to train this stage, the activities of Standing, Walking, Sitting and Jogging along with all falls combined in to one class are passed to the classifier. This dilutes the original ten-class problems in to a five-class sub problem. The output of this stage is the determination of whether a given recording is either one of the four ADLs (Standing, Walking, Sitting or Jogging) or a fall. If a recording has been detected to be a fall, it is sent to the second and third stages.

The second and third stages work in parallel on samples detected as falls from the first stage in the form of a voting machine. These two stages vote individually on the direction and severity of the detected fall samples. In order to train them, fall samples were relabeled to represent direction and severity only and are fed to the classifiers. For the direction, the classification problem is formulated as a three-class problem of determining fall directions as being Forward, Backward or Lateral. For the severity classifier, the classification problem is formulated as a two-class problem of a fall being either Soft or Hard. After a signal has passed through all necessary stages, the outputs of the individual stages are combined to indicate to the activity or type of fall being fed at the input.

Four classifiers were tested for each part of the hierarchical scheme, the classifiers considered were K-Nearest Neighbors, Support Vector Machines, Random Forests and eXtreme Gradient Boosting. Parameter tuning was performed using gradient search for each classifier over a range of values for each parameter.

K-Nearest Neighbor Compared to the SVM which attempts to create a decision boundary between classes on a global level, the K-NN algorithm operates locally. Starting from a data point, the 1 Nearest Neigbhor algorithm will assign each new sample the class of its neighbor. However, using a single neighborhood point to assign new classes may lead to erroneous classes being assigned. Therefore, typically multiple neighboring points are used to determine the label for a given point. In K-NN, 'K' represents the number of points used in the neighborhood to assign labels to data samples. Each considered sample is assigned the majority label within a neighborhood of K points around it. In this manner, the K-NN algorithm is successively able to create a decision boundary separating the data in to the different output classes.

2.8.2.3 Results

Three experiments were performed in this section. The first was to choose the best classifier to use, once that was determined. experiments were conducted to determine the observation window size and different combination of sensor modalities. We used the weighted F1-score as our training metric due to the imbalance in the samples of the different classes in the data. Moreover, for evaluation purposes, we report on the individual F1-scores for each output class and provide discussions as necessary. Furthermore, for the best performing results, the Precision and Recall and Specificity were also specified. Specificity can be calculated as given in Eq. 10. It is also called the True Negative Rate and indicates to a classifiers capability to correctly detect samples which don't belong to a given class. These metrics have been computed on a one versus all basis.

$$Specificity = \frac{True\ Negative}{True\ Negative + False\ Positive} \tag{10}$$

The data after the feature extraction stage was split in to a train/test partition based on a 75/25 ratio with a parameter grid being searched through to obtain tuning parameters for maximizing the weighted F1-score while using five-fold cross validation.

Classifier Selection

This test was conducted to determine the best classifier to use for the proceeding experiments of observation window selection and sensor modality selection. Here, experiments were performed for the considered window durations for each activity and the classifier which provided the best performance overall was chosen. The mean F1-scores for each output class for each classifier are shown in Figure 22. It can be observed that in general K-NN and SVM perform better compared to the ensemble models, the RFC and XGBoost. However, since the K-NN slightly outperforms the SVM in eight of the ten considered classes, we choose K-NN as the classifier for this framework.

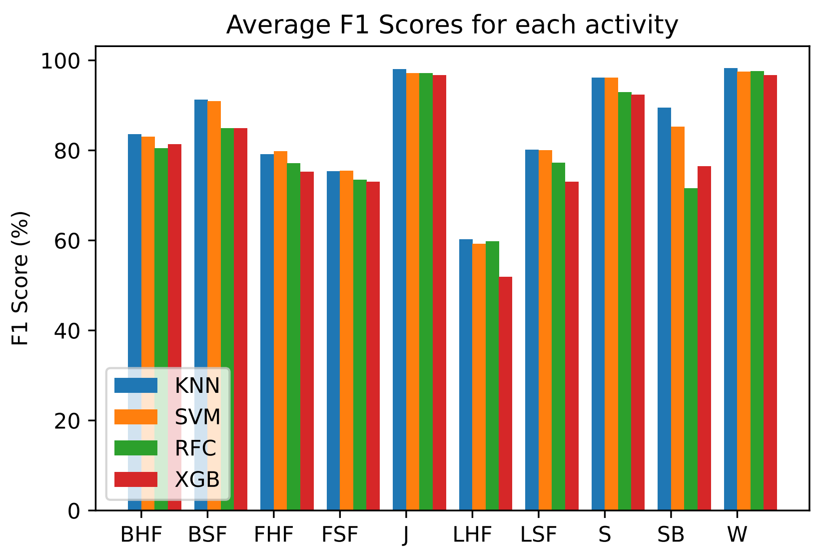

Figure 22. Average F1-scores for each activity for the four classifiers.

Observation Window Duration

An important consideration in working with activity recognition systems is to determine the appropriate observation window size for the analysis of sensor signals to accomplish the ADL recognition/fall detection task. The size of the observation window is important as a smaller observation window increases the response time of the activity recognition/fall detection system and it can also impact the time taken in the computation of features. In order to find the best observation window size, we perform experiments using five values, 2, 3, 4, 5 and 6 seconds. The classification results in terms of the F1-score are presented in Table 41. For each case, samples of duration equal to half of the observation window were extracted around the peak value of the SMV. From the table, it can be observed that an observation window of size 3 seconds produces the best results for six out of the ten output classes. It only produces poorer results for the classes *BHF*, *BSF* and *S*, *SB* where window sizes of 2 seconds, 6 seconds and 4 seconds respectively perform better than the 3 second windowing case. Upon further investigation of this phenomenon using the result of other classifiers, it was observed that the activities of (*BHF* and *BHF*) were best recognized by all the classifiers with a window size of 2 seconds (for the case of K-NN, there is a small difference between the 2 second and 6 second case), for the other two activities of *S* and *SB* too the F1-score was obtained for the 4 second duration (for the activity *S*, the difference in performance over windows larger than 4 seconds is very small). This could be attributed to the feature aggregation process in the max pooling operation in the different spatial segments.

Table 41. Performance for different observation window sizes.

Activity/Fall	**Observation Window Size (F1-score [%])**				
	2 sec	**3 sec**	**4 sec**	**5 sec**	**6 sec**
BHF	**86.79**	83.02	79.25	83.64	85.19
BSF	92.17	90.76	89.08	90.76	**93.22**
FHF	78.53	**80.47**	78.32	79.21	78.83
FSF	73.39	**77.18**	72.5	76.83	76.79
LHF	52.83	**67.8**	62.75	59.26	58.62
LSF	79.69	**82.73**	77.57	81.46	79.41
J	97.53	**98.27**	98.08	98.00	98.16
S	95.27	96.20	97.60	**95.84**	95.93
SB	87.29	85.71	**91.98**	90.61	91.71
W	98.08	**98.46**	98.12	98.35	98.16
Average	*84.16*	*86.68*	*84.52*	*85.40*	*85.60*

Sensor Modality

The second experiment in designing the proposed system is the determination of the best sensor modality to use. Using a single sensor would result in less data, faster processing and reduced hardware costs compared to the multisensor approach combining accelerometer and gyroscope. To do this, the classification framework was tested with 3 second windowed segments of the combined acceleromter and gyroscope data as well as data of the accelerometer and gyroscope sensors individually. The results of this experiment are presented in Table 42. It can be observed that using a combination of both accelerometer and gyroscope data together produces the best results for eight of the ten output classes. An accelerometer-only system produces better results for the detection of activity *SB* and the fall *FHF*. The outcome of this experiment agrees with previous work for fall detection by Waheed et. al. [436] on the SisFall dataset.

2.8.2.4 Discussion

Table 43 reports on the best results obtained for the proposed classification framework. These results were achieved by using windowed segments of 3 seconds and

Table 42. Performance for different sensing modalities.

Activity/Fall	Sensing Modality (F1-score [%]) Accelerometer + Gyroscope	Accelerometer	Gyroscope
BHF	**83.02**	67.92	82.14
BSF	**90.76**	85.48	78.18
FHF	80.47	**83.33**	71.17
FSF	**77.18**	73.21	63.96
J	**98.27**	97.79	95.59
LHF	**67.80**	54.55	55.56
LSF	**82.73**	76.34	73.21
S	**96.20**	95.61	91.17
SB	85.71	**86.21**	76.09
W	**98.46**	98.24	96.30
Average	*86.06*	*81.87*	*78.33*

combined data from the accelerometer and the gyroscope with a weighted F1-score of 94.67% on the test set.

Table 43. Best Results (Obs. Window : 3 sec, Sensing Modality: Acc. + Gyro.)

Activity/Fall	Precision (%)	Sensitivity/Recall (%)	Specificity (%)	F1-Score (%)
BHF	95.65	73.33	99.96	83.02
BSF	91.53	90.00	99.80	90.76
FHF	86.08	75.56	99.57	80.47
FSF	76.86	77.50	98.88	77.18
LHF	68.97	66.67	99.65	67.80
LSF	79.85	85.83	98.96	82.73
J	97.87	98.68	99.36	98.27
S	95.00	97.44	99.31	96.20
SB	93.75	78.95	99.80	85.71
W	97.95	98.97	98.36	98.46
Average	*88.35*	*84.29*	*99.37*	*86.06*

From Table 43, the best recognized ADLs are *W* and *J* whereas the best recognized fall is *BSF*. The worst performing class in ADLs is *SB* whereas the worst performing fall is *LHF*. Upon further inspection of the cause of the bad performance with *LHF*, looking at the confusion matrix, it was observed that *LHF* was most commonly confused with *FSF* which resulted in a reduction of the classification performance for this class. On the other hand, in the case of *FSF* (the second worse performing class), looking at the confusion matrix, it was observed that *FSF* was confused with *LSF* and *FHF*. Furthermore, the specificity values indicate that there has been very little mis-identification for each of the classes. When talking about the activity *S*, it was observed that samples from this activity were confused with the activity *W* which resulted in the sub-par performance of the classifier for its recognition.

2.8.3 Fall Detection with Severity and Direction along with ADL consideration using CNN-XGBoost

This section presents a scheme for performing fall detection considering fall direction and severity as well as activity recognition. Inertial sensor data taken from the SisFall dataset is used to develop the methodology. Data pre-processing is first carried out in terms of windowing and relabeling. Then, data augmentation is carried out for classes which do not have a sufficient number of samples. Lastly, feature extraction is performed along with classification. This work considers fall and activity recognition as a holistic problem, in that different types of falls and activities are considered thereby producing a more 'complete' recognition system for use in cyber-physical systems. Moreover, towards this end, a CNN-XGBoost combination is proposed.

2.8.3.1 Methodology The proposed scheme follows a typical deep learning solution framework. First, inertial sensor data from the IMU sensors contained within the SisFall dataset is pre-processed to extract windowed segments, then data augmentation is performed for minority classes, followed by feature extraction and then classification. This has been illustrated in Figure 23 and a discussion is provided for each of the steps in the proceeding sections.

Data Pre-processing

Before the IMU sensor recordings can be used for ADL and fall detection, raw sensor measurements need to be suitably processed. In this experiment, data pre-processing consists of two steps, the first is the extraction of uniform sized windows as was performed in the previous section from the IMU recordings and the next is data augmentation. Since windowing of the signals has already been discussed previously, only data augmentation is discussed here.

Data Augmentation The use of deep learning methods require a suitable amount of data to be present for them to learn the data pattern sufficiently well. Unfortunately, due to the nature of the problem considered, the relabeled data from the SisFall dataset contains a reduced amount of data for some of the classes, especially fall classes and also for the ADL of Standing. In order to alleviate this shortcoming, data augmentation was employed to increase the number of samples from these classes. Three augmentations were performed for each of the extracted recordings for the classes *SB*, *FHF*, *FSF*, *BHF*, BSF, *LHF*and *LSF*. These were the addition of noise, scaling and resampling after inerpolation [439]. For augmentation with noise, white gaussian noise was added to the extracted windows of the considered classes. The noise was generated using a standard deviation equal to 0.01. The addition of the noise simulates measurement noise during recording which might be encountered when IMU based fall detection systems are employed. For scale based augmentation, the original extracted window was multiplied by a random number from the uniform distribution between 0.8 and 1.2 thereby scaling it between 80% and 120% of its original form. By doing so, changes in amplitude over the same type of activity/fall are incorporated. This could indicate to a change in fixation (loosening etc) of the sensing

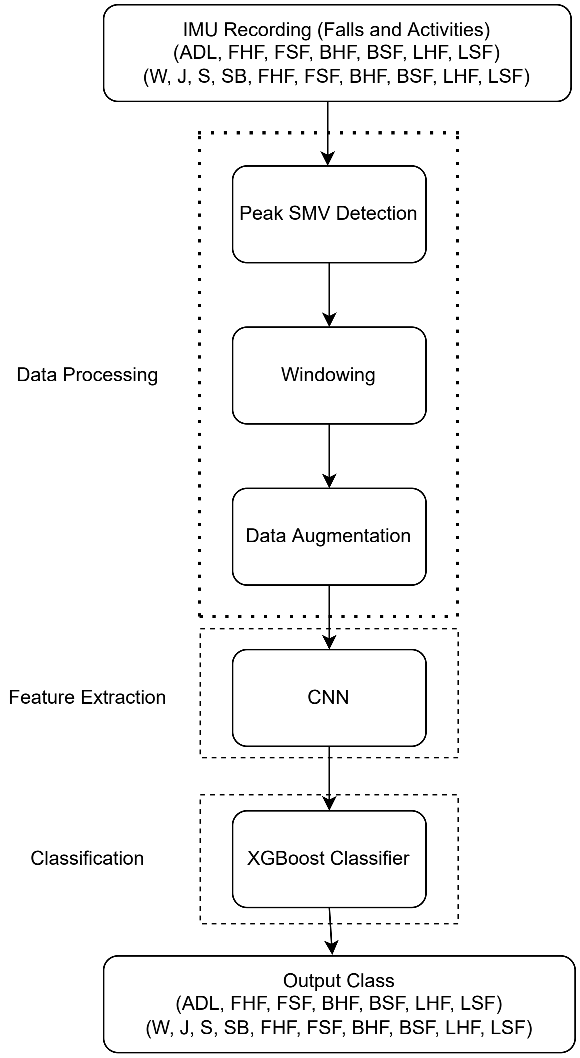

Figure 23. CNN-XGBoost based classification scheme for ADL and Fall detection.

unit to a subject or their different physique and subsequent fall intensity response. Lastly, in order to incorporate sampling inconsistencies, the windows are first upsampled and then downsampled. This was done by a scale of ten. With this strategy, each from original window were produced three additional samples. An illustration for the results of the augmentation process for an X-axis accelerometer measurement for a lateral fall has been shown in Figure 24.

Feature extraction

The aim of feature extraction for a classification problem is to produce a representation of the input that can be better used to indicate to the output class. In this regard, research in the area of fall detection with inertial sensors has made use of different types of hand crafted features such as statistical, time and frequency domain as well as wavelet transforms [479, 480, 481, 482]. Convolutional Neural Networks (CNNs or Covnets) are a set of neural networks developed following the visual cortex within the brain [483]. CNNs perform operations on inputs by introducing convolutions of

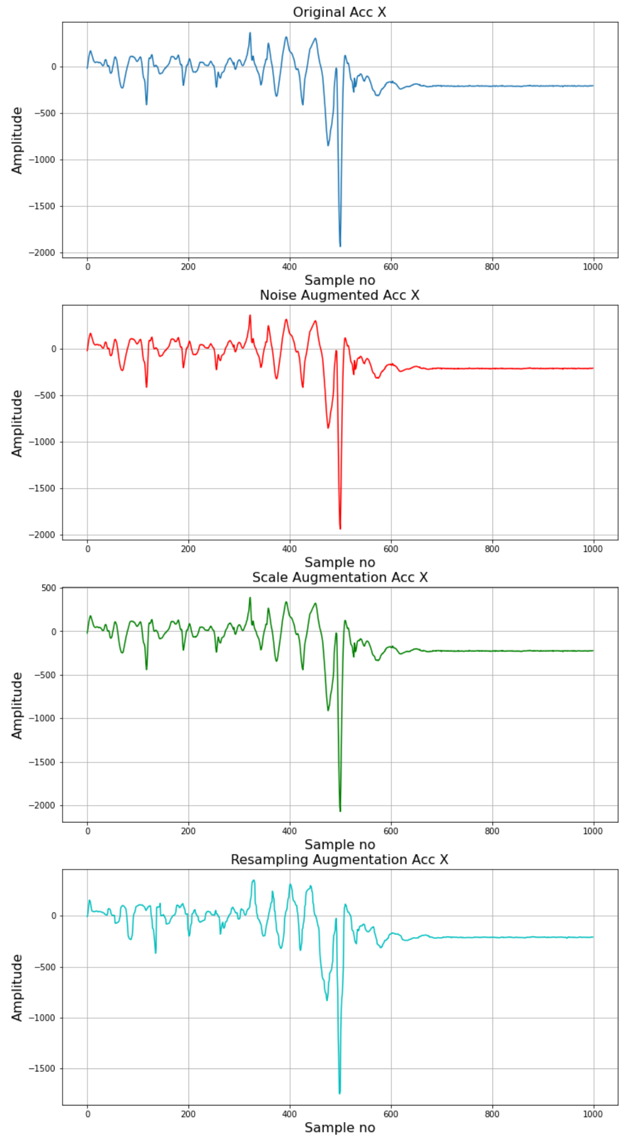

Figure 24. Illustration of data augmentation. (X component of Accelerometer, Lateral Fall)

several filters with learnable weights to gauge the importance of each datapoint in the input. These layers containing the filters and to which the input is provided are called convolutional layers. Through these learnable weights, CNNs are able to capture temporal and spatial dependencies of the input. Moreover, using the same filters for different inputs reduces the number of parameters as the weights are reused. This allows CNNs to develop a deeper understanding of the provided input compared to typical multilayer perceptron models. CNNs have revolutionized the field of computer vision where they have been used for a variety of tasks such as classification, object detection, segmentation and object counting [484, 485] and they have also successfully been used for applications within the speech and other time series signal application domain [486, 487, 488]. In this experiment, rather than using hand crafted features as in the previous experiments, a CNN has been used to perform feature extraction in order to take advantage of the spatial and temporal dependency capturing capabilities of CNNs. CNNs are usually comprised of several convolutional (Conv) layers. Within a multilayer CNN, the earlier Conv layers capture low level features from the input with more complex features being computed by the successive layers. Also, CNNs may employ pooling layers between convolutional layers so as to reduce the size of the input passed on to successive layers and therefore reduce computation. The proposed network along with the XGBoost classification stage is illustrated in Figure 25.

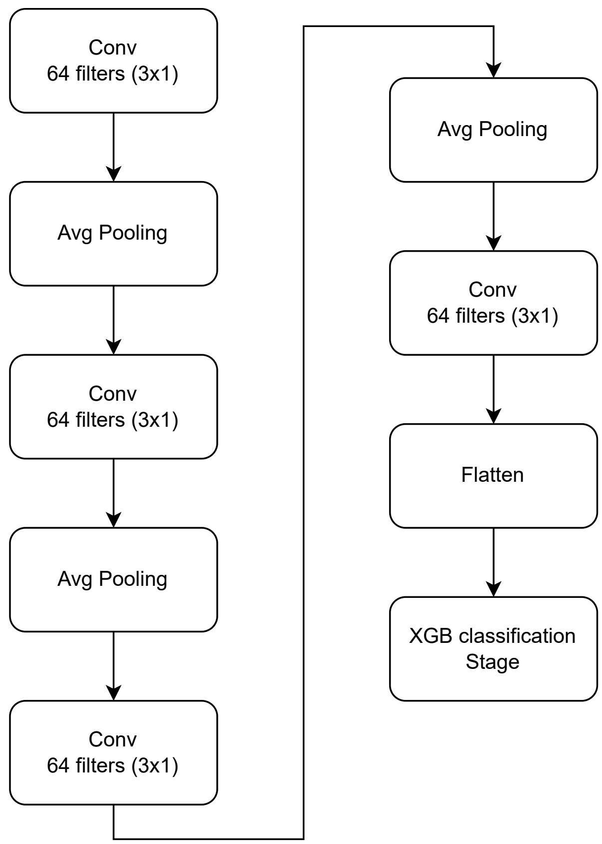

Figure 25. CNN Network for feature extraction and XGBoost classification stage.

As seen from the figure, the network consists of four Conv layers of 64 filters each. These layers have filters of size 3x1 and are used to extract features from raw inertial sensor measurements. Each Conv layer is followed by a Relu activation function which applies a non-linearity to the output of the Conv layers. To condense the output of the first three Conv layers, a pooling layer is utilized. Instead of max pooling, average pooling is used in this network. Max pooling picks out the largest value of the patch of data being observed currently, in contrast, average pooling uses the average of the data being observed. Average pooling has been successfully used in place of max pooling in a variety of scenarios [489, 490]. Normalization is performed using a batch normalization layer for each Conv layer. The output of the last Conv layer are feature maps derived from the input raw inertial sensor measurements from both the accelerometer and the gyroscope.

Classification

Classification is carried out by using a eXtreme Gradient Boosting (XGBoost) classifier. The output from the CNN is first flattened and then provided as an input to the XGBoost stage. The parameters of the XGBoost classifier are tuned through a parameter search over a range of values. We make use of an XGBoost classifier due to its suitability for a large dimensional input which results from the flattened CNN output.

2.8.3.2 Results

In order to use the proposed CNN for feature extraction, it must first be trained accordingly. In order to train the network, a fully connected layer with a softmax output was added as the final stage to serve as the intermediate temporary output determinant stage. The windowed data from the SisFall dataset was divided in to three sets, train, validation and test in a stratified manner. A learning rate of 0.01 was used for the network with a batch size of 20 and the stochastic gradient descent was used as the optimizer. Moreover, the metric chosen was the average of the recall scores of all classes together, also called as the unweighted average recall (UAR). The recall is considered as one wants the system correctly classify as many positive samples for every class as possible. The final network was determined using early stopping. The unweighted average recall (UAR) can be computed as,

$$UAR = \sum_{n=0}^{k} \frac{\left(\frac{TP_n}{TP_n+FN_n}\right)}{k} \tag{11}$$

where k stands for number of classes and $\mathrm{TP_n}$ stands for the number of True Positive samples in the nth class and $\mathrm{FN_n}$ stands for number False Negative samples in the nth class. Therefore, the average of the individual recall scores from all classes was aimed to be maximized.

Data from the training set was provided to the CNN network after performing data augmentation on the minority classes, during training, the validation set was used to observe performance of the network and determine the best performing instance. Once training was finished, the last fully connected classification layer was removed and replaced by an XGBoost classification stage. To train the XGBoost stage, the weights of the final best performing CNN model were loaded in to CNN layers of the network and input samples were then passed through them as before. Using the output of the CNN stage as an input for the XGBoost, a search was then carried out to determine the optimal parameter values. After training of the XGBoost stage, the completely trained CNN-XGBoost model was tested using the test set. Two sets of tests were performed, one while considering all ADLs as a single class versus individual falls and the other considering common individual ADLs and individual falls.

One ADL vs Individual Falls

In this experiment, all ADLs present in the SisFall dataset were combined in to one class to build a generic ADL versus fall system. The results are presented in Table 44. It can be observed from the table that the *ADL* class has been detected with perfect recall score. Among the falls, the worst performance was observed for *LHF* with the best performance for *FHF*.

Table 44. One ADL vs. Individual Falls.

Activity/ Fall	Precision (%)	Sensitivity/ Recall (%)	Specificity (%)	F1-score (%)
BHF	100	91.67	100	95.65
BSF	100	95.83	100	97.87
FHF	85.37	97.22	99.41	90.91
FSF	92.86	81.25	99.71	86.67
LHF	72.73	66.67	99.70	69.57
LSF	89.58	89.58	99.50	89.58
ADL	99.54	100	97.78	99.77
Average	*91.44*	*88.89*	*99.44*	*90.02*

Individual ADLs vs Individual Falls

The results of the experiment have been shown in Table 45.

Table 45. Individual ADLs vs. Individual Falls.

Activity/Fall	Precision (%)	Sensitivity/Recall (%)	Specificity (%)	F1-Score (%)
BHF	100	75.00	100	85.71
BSF	95.83	95.83	99.90	95.83
FHF	76.19	88.89	99.01	82.05
FSF	90.24	77.08	99.60	83.15
LSF	86.79	95.83	99.30	91.09
LHF	75.00	75.00	99.71	75.00
J	96.71	96.71	99.01	96.71
S	96.77	96.77	99.57	96.77
SB	91.18	83.78	99.70	87.32
W	97.21	97.63	97.77	97.42
Average	*90.59*	*88.25*	*99.36*	*89.11*

An average recall of 88.25% was observed for the experiment. From Table 45, when looking at the recall achieved for the individual activities, it can be observed that the best recognized activities are *W* and *J* with each achieving a recall of around 97%. The worst performance in terms of ADLs was achieved for the activity *SB* for which a recall of 83.78% was attained. Considering the case of falls, out of the six falls, the worst recall score of 75% was achieved for BHF and LHF whereas the best recall score was of 95.83% was achieved for *BSF* and *LSF*. The fall class *FSF* was also not identified well, achieving a recall score of 77.08%. Furthermore, as observed from the table, a high value of specificity was obtained for all ADLs and falls indicating to correct determination of negative samples for each class as well. Investigating the performance of the network from the confusion matrix, it was observed that the worst performing activity *SB* was equally confused as *S* and *W*, this can be attributed to the fact that the *SB* activity includes slight bending which could lead to confusion for the classifier. For the worst performing falls, it was observed that *BHF* was confused with *LHF* and *LSF* whereas *LHF* was confused with *FHF* and *FSF*. The confusion between the falls is apparent from plots shown in Appendix A where *BHF* has very similar accelerometer signal values to these classes.

2.8.3.3 Discussion

Inorder to better understand the performance of the network for the various fall detection types from Tables 44 and 45, the performance of the network for fall severity has been illustrated in Figure 26. It can be observed that soft falls are better detected compared to hard falls with UARs for the individual ADL and fall experiment being 79.63% for hard falls and that for soft falls is 89.58%. For the one ADL and individual fall experiment, the difference is less stark, being 85.18% for hard falls and 88.88% for soft falls.

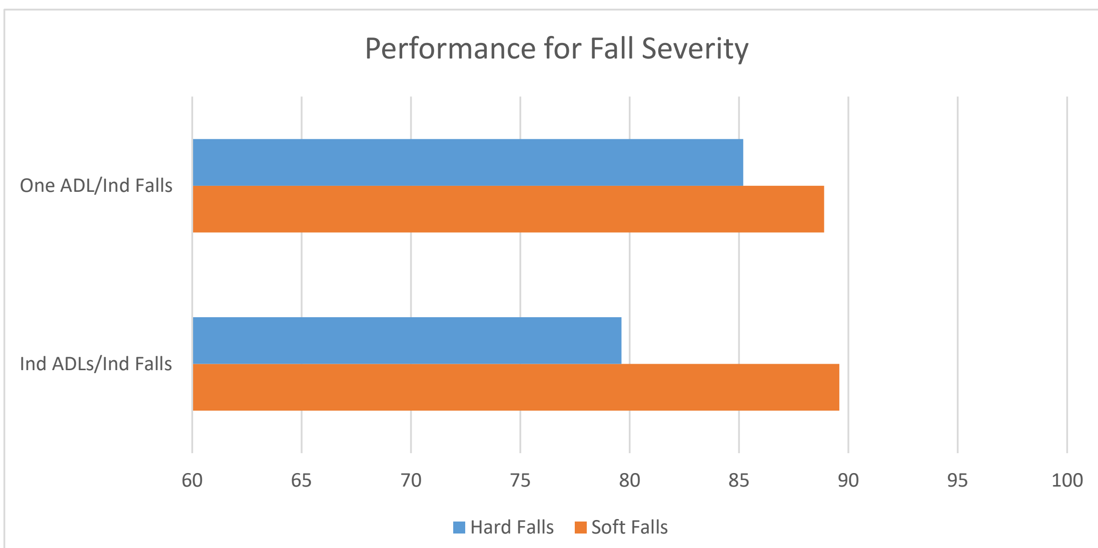

Figure 26. Network performance for different fall directions.

Figure 27 illustrates the performance of the network in fall detection for the three directions considered for both experiments, regardless of severity. It can be observed that falls in backward and lateral directions are determined with equal effectiveness, an average recall score (UAR) of 85.42% was achieved for both cases where as for the forward direction, the UAR was 82.99%. As can be seen, there is a only a small difference (3.43%) between the achieved UARs.

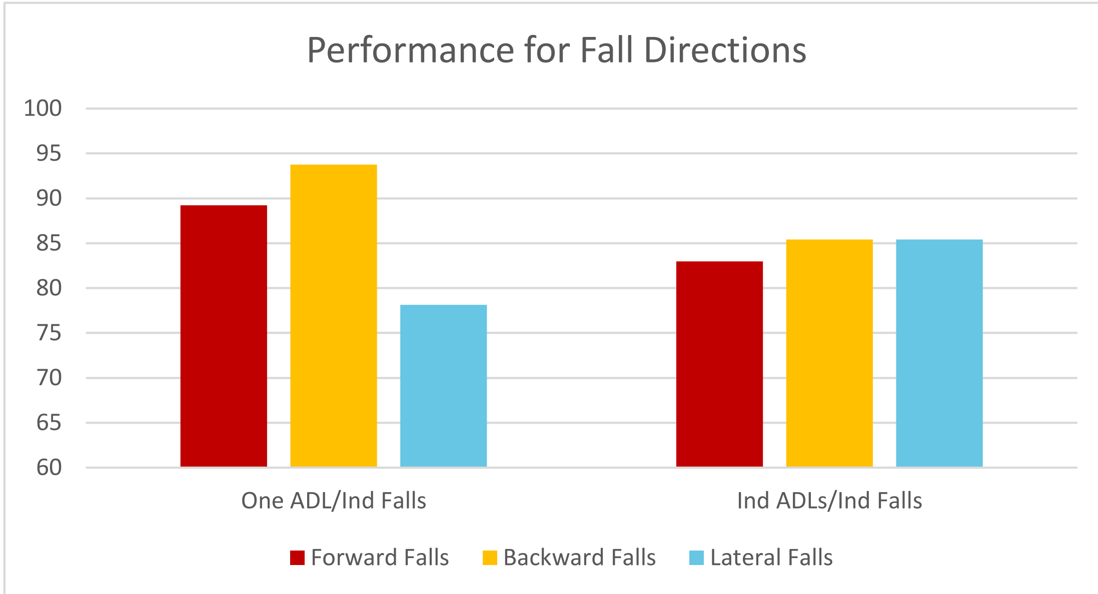

Figure 27. Network performance for different fall directions.

In order to further test the performance of the developed CNN-XGBoost scheme, tests were performed with the CNN network architecture presented in [430]. We choose the technique of [430] as the authors provide very good results for fall detection using the SisFall dataset using a deep learning model. The results for the performance of the considered technique in comparison to the method presented in this experiment is shown Table 46 and 47. The Recall scores have been presented. It can be observed that the proposed method outperforms the work of Casilari et. al. for all classes except LHF. Moreover, there is a large difference in the average recall achieved.

Table 46. Comparison with State of the art (Individual ADLs vs. Individual Falls.)

Activity/Fall []	**Sensitivity/Recall (%)**	
	Work of [430]	**Proposed work**
BHF	66.67	91.67
BSF	64.58	95.83
FHF	95.83	97.22
FSF	75	81.25
LHF	77.78	66.67
LSF	87.5	89.58
ADL	99.31	100
Average	*80.95*	*88.89*

The mean recall score achieved is 85.69% for [430] for the individual ADLs vs Individual Fall experiment compared to more than 88% for the proposed scheme. It can be observed from Table 47 that the proposed CNN-XGBoost combination outperforms the work of Casilari et al. [430] in seven out of the ten output classes for recall while achieving a similar performance for the classes of *BHF*, *LHF* and *S*.

Table 47. Comparison with State of the art (Individual ADLs vs. Individual Falls.)

Activity/Fall	**Sensitivity/Recall (%)**	
[]	**Work of [430]**	**Proposed work**
BHF	75.00	75.00
BSF	87.50	95.83
FHF	80.56	88.89
FSF	72.92	77.08
LHF	75.00	75.00
LSF	93.75	95.83
J	96.30	96.71
S	96.77	96.77
SB	81.08	83.78
W	95.04	97.63
Average	*85.39*	*88.25*

2.8.4 Cross dataset non-binary fall detection with a ConvLSTM-attention network

This section presents a discussion on fall detection with severity and direction along with ADL recognition on the SisFall and the K-Fall datasets. Two experiments were conducted in this regard, one with all ADLs as one class versus the six fall types considered and another for individual ADLs and falls.

2.8.4.1 Data Labeling

Since this experiment also uses the K-Fall dataset, the labeling scheme for K-Fall is shown in Table 48. The labeling for the SisFall dataset was retained the same for the individual ADL and fall experiment. However, for the one ADL and fall experiment, all activities were labeled as a single class *ADL*.

2.8.4.2 Methodology

The methodology consists of the two steps of data preprocessing and classification. Data preprocessing has been performed in the same manner as in section 2.8.3.1. Therefore, this section focuses on the ConvLSTM network used. The network used in this work is illustrated in Figure 28. It consists of four layers, the first of these is the ConvLSTM layer. The ConvLSTM layer combines the properties of sequential learning associated with LSTMs with the feature extraction capabilities of convolutional neural networks and they have found successful use in the human activity recognition/fall detection domain [491, 492]. By replacing the simple matrix multiplication within LSTM cells by a convolutional operation, the ConvLSTM can capture spatio-temporal dependencies as opposed to the temporal only qualities offered by LSTMs. Such spatial information is useful for the problem of fall detection where not only the sequence contained within the recordings is important but also the spatial information is also important to determine things like direction for e.g. The ConvLSTM layer is followed by a Self-Attention layer to incorporate attention

Table 48. Labeling for K-Fall Dataset

Activity Code	Assigned Label	Fall code	Assigned Label
D01	ADL/-	F01	FSF
D02	ADL/SB	F02	BSF
D03	ADL/SB	F03	LSF
D04	ADL/-	F04	FSF
D05	ADL/S	F05	LSF
D06	ADL/W	F06	FSF
D07	ADL/W	F07	LSF
D08	ADL/J	F08	BSF
D09	ADL/J	F09	FSF
D10	ADL/-	F10	LSF
D11	ADL/-	F11	FHF
D12	ADL/-	F12	FHF
D13	ADL/S	F13	FHF
D14	ADL/S	F14	LHF
D15	ADL/S	F15	BHF
D16	ADL/S	-	-
D17	-/ADL	-	-
D18	ADL/S	-	-
D19	ADL/S	-	-
D20	ADL/W	-	-
D21	ADL/-	-	-

mechanism. Through the attention layer, important parts in the signal which can help to determine the output class correctly are identified. For the current work, a global attention mechanism which looks at the complete sequence is used. After the attention layer follow two fully connected layers, one with a relu activation function and the other with a softmax function to determine the output class.

2.8.4.3 Results

Two experiments were performed in total on the two datasets considered. The experiments consisted of a fall vs ADL scenario and another scenario in which ADLs were conisdered individually to provide a combined fall and ADL recognition system. The results for each of the experiments have been presented in this section. The network was trained using an Adam optimizer for 40 epochs with early stopping being used to retain the best performing model before running it on the test set. Also, during training, the network was tasked to maximize the average recall of all classes combined to ensure that positive samples of each class were prioritized for correct detection, however, in addition to recall, the precision, specificity and F1-score have also been reported in the results presented for the sake of completeness.

One ADL vs Individual Falls

In this experiment, all the ADLs were considered a single class while the categories for falls were retained as per individual directions and severity levels. Table 49 presents the results for this experiment with the SisFall dataset for which an average recall of 90.02% was achieved. It can be observed that the network is able to determine ADLs very well achieving a recall of 99.93%. Considering falls, the class *LHF* is the worst

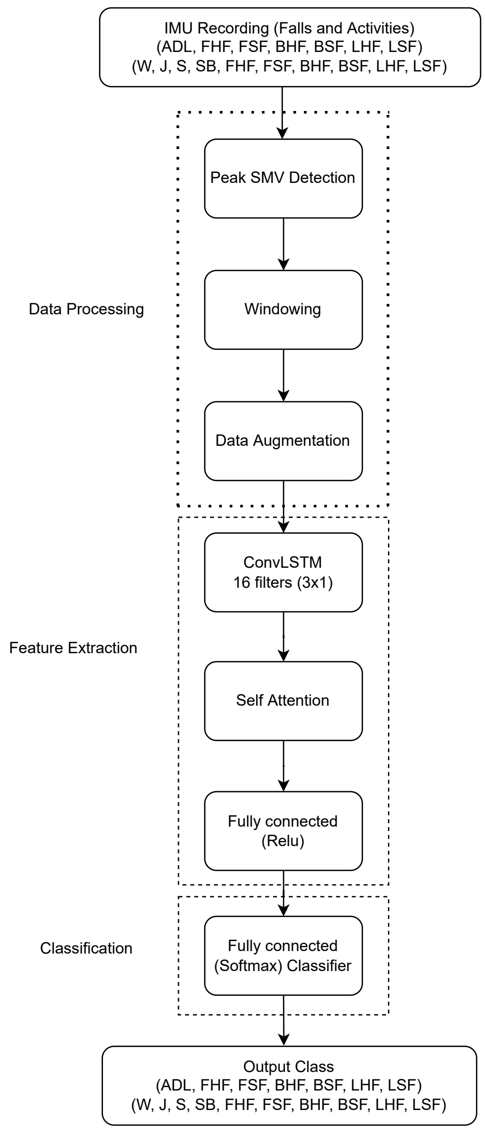

Figure 28. ConvLSTM-attention Network.

performing fall which has been detected with a recall of only 68.42%. It was observed that falls from *LHF* were confused with forward falls and *LSF*. The best performing fall is *FHF* with a recall of 98.28%.

Table 49. Results for SisFall: One ADL vs. Individual Falls

Activity/ Fall	**Precision** (%)	**Sensitivity/ Recall** (%)	**Specificity** (%)	**F1-score** (%)
BHF	100	94.74	100	97.30
BSF	94.74	94.74	99.88	94.74
FHF	83.82	98.28	99.36	90.48
FSF	94.20	84.42	99.76	89.04
LHF	81.25	68.42	99.83	74.29
LSF	89.61	89.61	99.53	89.61
ADL	99.8	99.93	98.96	99.87
Average	*91.92*	*90.02*	*99.62*	*90.76*

Table 50. Results for K-Fall: One ADL vs. Individual Falls

Activity/ Fall	**Precision** (%)	**Sensitivity/ Recall** (%)	**Specificity** (%)	**F1-score** (%)
BHF	100	92.00	100	95.83
BSF	98.00	96.08	99.90	97.03
FHF	82.93	91.89	98.63	87.18
FSF	93.41	85.00	99.4	89.01
LHF	78.57	88.00	99.44	83.02
LSF	96.97	96.00	99.70	96.48
ADL	99.45	99.72	98.93	99.59
Average	*92.76*	*92.67*	*99.43*	*92.59*

Table 50 presents the results for the K-Fall dataset when all ADLs are combined in to one class. It can be observed that like the case for the SisFall dataset, the ADLs in this case have been detected very well, resulting in a recall of 99.72%. Considering the performance of falls, the best performing fall classes were *BSF* and *LSF* for which a recall of 96% was achieved. The worst performing fall class in this case was *FSF* with a recall of 85%.

Individual ADL vs Individual Falls

In this experiment, individual ADLs were considered as separate classes in addition to the six fall classes mentioned before. This exercise was carried out to assess the designed systems performance as a combined ADL recognition and fall detection system. The result for the SisFall dataset are presented in Table 51. The average recall achieved for this experiment using the SisFall dataset is 91.49%.

Considering fall performance, it can be observed from Table 51 that the class *BSF* is the best recognized fall, having a recall of 94.74% with the fall *LHF* being the worst recognized one with a recall of 73.68%. The best recognized ADL is the class *W* with a near perfect recall of 99.06%. The worst recognized ADL is *S* with 96.98% which is not a large difference compared to the performance for falls. Table **??** presents the results of the network for the K-Fall dataset for this experiment. It

Table 51. Results for SisFall: Individual ADLs vs. Individual Falls

Activity/ Fall	**Precision** (%)	**Sensitivity/ Recall** (%)	**Specificity** (%)	**F1-score** (%)
BHF	89.47	89.47	99.88	89.47
BSF	100	94.74	100	97.30
FHF	86.89	92.98	99.51	89.83
FSF	94.37	87.01	99.75	90.54
LHF	70.00	73.68	99.64	71.79
LSF	88.89	93.51	99.44	91.14
J	98.44	97.42	99.53	97.93
S	97.97	96.98	99.73	97.47
SB	94.74	90.00	99.81	92.31
W	98.00	99.06	98.39	98.53
Average	*91.88*	*91.49*	*99.57*	*91.63*

can be observed that in this case, the best performance for the fall categories is both for *BSF* and *LSF* each of which have a recall of 98%. The worst performing fall is the class *BHF* , it was observed that samples from this class were confused with the class *BSF*. For the case of ADLs, the best performing ADL for the K-Fall dataset is *SB* with a recall of 97.5% with the worst being the class *J*.

2.8.4.4 Discussion

When comparing the performance of ADLs and falls overall for both experiments, it can be observed that ADL recognition performance was much better than the performance for fall detection. For both experiments conducted, when comparing the performance of the network across datasets, the network performs better on the K-Fall dataset compared to the SisFall dataset. This could be attributed to the difference in the volunteer make up of the participants as K-Fall mostly had young people volunteering to perform falls whereas SisFall contains a wide variety of age groups and gender make up in its volunteers. To gather further insights in to the performance of the network in terms of fall detection for the two experiments conducted across both datasets, Figure 29 illustrates the networks performance for falls for both datasets and experiments. An average recall of 88.37% was achieved for the SisFall dataset and 91.49% for the K-Fall dataset for the one ADL vs individual falls experiment and 88.57% and 91.33% respectively for the individual ADL and individual falls experiment. It can be observed that the classes *LSF*, *FHF* and *BSF* were recognized very well, each achieving a recall of nearly 90% or higher. The class *FHF* was also detected sufficiently well, achieving an average recall of more than 86% across all experiments. For SisFall, the least performing class was *LHF* and for K-Fall it was *BHF*.

Figure 30 investigates the performance of the network when considering different fall severity levels for all the experiments. It can be observed that soft falls have been consistently better recognized compared to hard falls. This difference is largest (6%) when individual ADLs are considered as separate classes for both datasets. For the experiment where ADLs are grouped in to a single class, there is only a slight

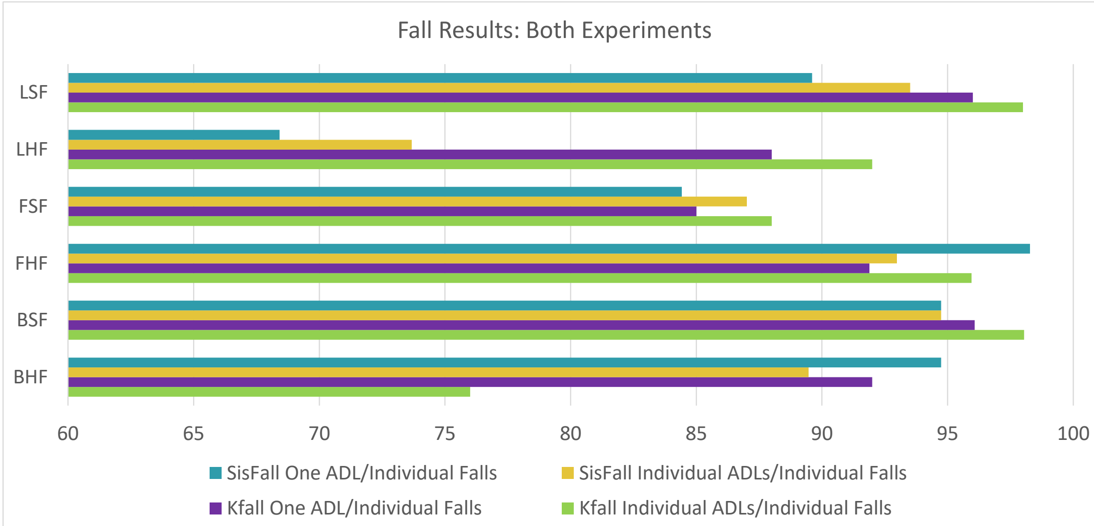

Figure 29. Fall detection performance for both experiments.

difference observed between the fall severity levels for the K-Fall dataset whereas for SisFall, there is still a significant difference of more than 5% between the detection of soft and hard falls.

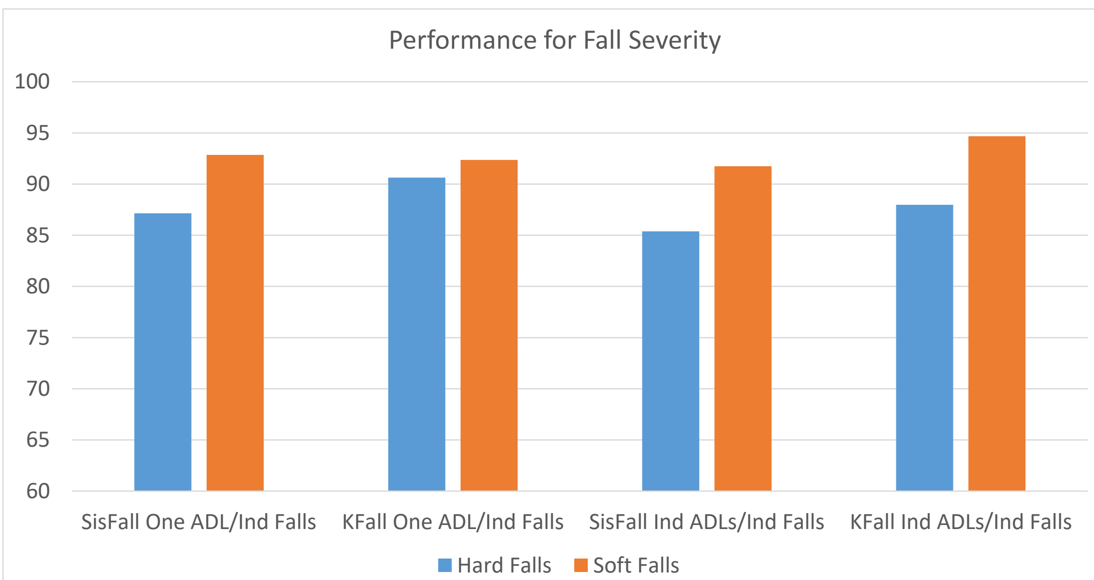

Figure 30. Fall severity performance.

In Figure 31, the results from the experiments for fall direction have been illustrated. In this case, the performance of the network for tests using SisFall yield best results for falls in the backward direction with falls in the lateral directions being recognized the worst. In contrast, for the K-Fall dataset, falls in the lateral direc-

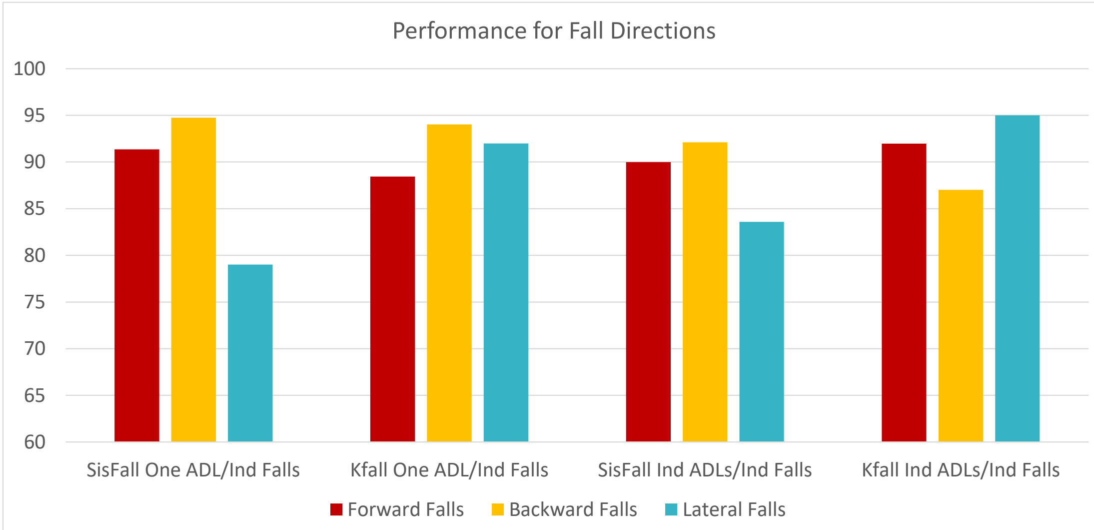

Figure 31. Fall direction performance.

tion have been detected quite well, followed by backfward falls and forward falls. In order to gauge the performance of the network, the work of We et al. [459] was used as a comparison as they also utilize sequential modeling in their work and cater to non-binary fall detection as well. Table 53 and 54 presents the results for both experiments with the cases where the work of Wu et al. outperforms the presented method highlighted. It can be observed that the presented method outperforms the method of [459] sufficiently well. Another thing to note is that as observed with the results presented, a similar pattern of variation was observed for both data sets with respect to the two experiments in that a degradation was observed for experiments for K-Fall whereas an improvement was observed for SisFall.

Table 53. Comparison with State of the art (Individual ADLs vs. Individual Falls.)

Activity/Fall	**Sensitivity/Recall (%)**			
▯	**Work of [459]**	**Proposed work**	**Work of [459]**	**Proposed work**
BHF	63.16	94.74	92.00	92.00
BSF	86.84	94.74	96.08	96.08
FHF	77.59	98.28	83.78	91.89
FSF	76.62	84.42	90.00	85.00
LHF	68.42	68.42	64.00	88.00
LSF	77.92	89.61	93.00	96.00
ADL	98.78	99.93	99.58	99.72
Average	*78.48*	*90.02*	*88.35*	*92.67*

3 Conclusion

Fall detection is an important task in the field of ambient assisted living. Towards this end, four experiments were performed in this chapter. In the first experiemt, a fall

Table 54. Comparison with State of the art (Individual ADLs vs. Individual Falls.)

Activity/Fall	Sensitivity/Recall (%)			
[]	Work of [459]	Proposed work	Work of [459]	Proposed work
BHF	58.33	89.47	76.00	76.00
BSF	87.50	94.74	100	98.04
FHF	83.33	92.98	91.89	95.95
FSF	81.25	87.01	89.00	88.00
LHF	41.67	73.68	64.00	92.00
LSF	81.25	93.51	95.00	98.00
J	98.77	97.42	55.45	87.13
S	88.71	96.98	81.17	90.91
SB	83.78	90.00	87.50	97.50
W	96.12	99.06	87.30	95.24
Average	*80.07*	*91.48*	*82.74*	*91.88*

Table 55. Results for the four Experiments for the SisFall dataset (F1-Score[%])

Activity/Fall	Exp. 1	Exp. 2	Exp. 3 (One ADL)	Exp. 3 (Ind ADLs)	Exp. 4 (One ADL)	Exp. 4 (Ind ADLs)
BHF	87.27	83.02	95.65	85.71	97.30	89.47
BSF	95.08	90.76	97.87	82.05	94.74	97.30
FHF	72.43	80.47	90.91	75.00	90.48	89.83
FSF	73.25	77.18	86.67	95.83	89.04	90.54
LHF	66.67	67.80	69.57	83.15	74.29	71.79
LSF	84.65	82.73	89.58	91.09	89.61	91.14
J	-	98.27		96.71	-	97.93
S	-	96.20		96.77	-	97.47
SB	-	85.71		87.32	-	92.31
W	-	98.46		97.42	-	98.53
ADL	-	-	99.77	-	99.93	-
Overall Average	-	*86.06*	*90.02*	*89.11*	*90.76*	*91.63*
Fall Average	*79.89*	*80.32*	*88.38*	*85.47*	*89.24*	*88.34*
ADL Average	-	*94.66*	*99.77*	*94.55*	*99.87*	*96.56*

only detection system that considers the direction as well as the severity of falls was presented. Inertial measurement sensor data from a publicly available dataset was used to carry out this task. Following a typical machine learning pipeline, the data was first preprocessed by extracting equal duration segments from the accelerometer and gyroscope sensor signals based on the acceleration magnitude. This is followed by the computation of various time and frequency domain features for each of these segments. The next step is feature selection and classification which is performed by using four different machine learning algorithms popular in fall detection systems. First, a baseline performance is established and then feature reduction is performed aiming to maintain or improve algorithm performance by elimination of redundant features. It was observed that the weighted F1-score improved slightly (by just over 1%) for both experiments.

The other three experiments considered activity of daily living and fall detection simultaneously. For the second experiment, utilizing inertial sensor data, a hierarchical classification framework using wavelets and adaptive pooling was presented. To achieve this, inertial sensor recordings (accelerometer and gyroscope) from the SisFall dataset were utilized and windowed segments were extracted from each recording. Following this, a level-4 haar wavelet was used to extract wavelet coefficients from these windowed segments and then 4-2-1 1-D Spatial Pyramid pooling was used to summarize the output of the wavelet feature coefficients at each approximation and detail level before the max pooled coefficients were concatenated to form the final feature vector. A hierarchical classification scheme was then used consisting of three classification stages, one for determining individual ADLs versus a generic fall and the second and third for fall direction and severity respectively with both voting together to determine the severity and direction of a fall. Towards this end, experiments were conducted to determine the most appropriate size of the observation window as well as sensing modality used. It was found that for the proposed setup, a window duration of 3 seconds produced the best results while using data from both the accelerometer and gyroscope. In the third experiment, Inertial Measurement Unit (accelerometer and gyroscope) data from the SisFall dataset is first windowed into non-overlapping segments of duration 3 seconds. After suitable data augmentation for the minority classes, the windowed segments are passed to a Convolutional Neural Network for feature extraction. The CNN is trained to maximize the unweighted average recall for the validation partition. Once the CNN is trained, an XGBoost last stage is used for classification. Experiments conducted on the test set achieve an unweighted average recall of 88% for both the one ADL versus individual falls and individual ADLs versus fall experiments. In comparison with other techniques used for this task, the proposed scheme produces sufficiently better results, thereby demonstrating the efficacy of the proposed method. Lastly, a ConvLSTM network with attention has been used for detection of falls considering fall direction and severity. Using data from the SisFall and the K-Fall datasets, two experiments were conducted on inertial sensor data. The first considered falls versus ADLs and the other combined ADL recognition and fall detection. Results for both experiments achieved an average recall of more than 90%. The results from the cross-dataset performance indicate to the robustness of the designed methodology.

Table 55 presents the F-1 scores for the four experiments considering fall detection with direction and severity for the SisFall dataset. For experiment 1, only the fall detection with direction and severity part has been reported. A progression can be observed in the detection of falls wherein an increase of nearly 8.5% in the average F1-score was achieved.

CHAPTER VI

CONCLUSION AND FUTURE WORK

1 Conclusion

This dissertation provided a coverage of the use of Internet of Things towards the development of smart communities. While doing so, the main applications of ML/DL algorithms as well as optimization algorithms were discussed and mapped. Furtheremore, other technological components of the IoT such as sensing, communication, security and privacy were also talked about. Lastly, considering the case of smart health, specifically fall detection, experimentation and analysis was conducted and the results presented.

In this regard, the usage of inertial sensor data has been very popular as they do not restrict a users movement and are also easy to deploy compared to other methods. Four approaches that considered fall detection with direction and severity were presented. In this regard, four experiments were conducted, first was the development of a fall only detection system whereas the other three were for a combined ADL and fall system. For the other three experiments, a hierarchical and two deep learning based approaches were tested. The designed methodologies were compared to the state of the art and were found to outperform them.

2 Contribution

The Internet of Things has spearheaded tremendous change in society by allowing for capturing measurements in different facets of our daily lives. With such a fundamental impact being made, it is pertinent that researchers commit to analyzing the current penetration that IoT possesses in city operation and also providing impetuses for new and novel application development. This work aims to address these needs by discussing the use and role of the internet of things in smart communities. The contribution of the work are as follows:

1. Provide a coverage of the IoT based smart city ecosystem in terms of the technologies utilized. Discussion has been provided about the sensors, communication technologies as well as the IoT architectures that enable IoT smart city development. Finally a review of the security and privacy issues was also discussed.

2. Present a study of the uses of ML and DL methods for different applications of smart cities in an IoT context. To this end, the type of architecture employed and the data source type were also covered.

3. The use of optimization algorithms in IoT smart cities has been provided. To the best of our knowledge, this is the first comprehensive review of optimization algorithms in IoT Smart Cities.

4. Considering the case of smart health, methodologies have been devised for the novel case of fall detection with direction and severity detection. In this respect, the performance time, frequency domain and statistical features on inertial sensor data has been analyzed and presented.

5. A combined activity of daily living and fall detection framework with fall direction and severity consideration has been discussed. Four approaches in this regard have been presented, one utilizing time-frequency information using wavelets along with a hierarchical classification scheme, third a Convolutional Neural Network-eXtreme Gradient Boost combination and the last being a ConvLSTM has been proposed. The proposed approaches has been shown to outperform the state of the art in the field.

3 Future Work

The IoT is revolutionizing development of smart city applications. There are several opportunities. Some of these are listed below:

- *IoT Systems for Smart Cities*
    - A major research area is in the security and privacy of IoT in smart cities in terms of encryption techniques, authentication protocols, data anonymization techniques and other methods to prevent unvalidated access to the IoT network. As mentioned before technologies such as blockchain could help introduce access tracking and control, secure device discovery, prevention of spoofing, data loss while ensuring that end to end encryption is also used.
    - Of the data transfer standards developed till now for IoT, many are not compatible with each other. Work needs to be carried out in this regard to enable intercommunication of sensor nodes using different protocols while utilizing low power, which is imperative for sensor nodes in the network.
    - Another area to work on, is the development of efficient storage techniques and low power hardware which can reduce operational costs. From a deployment perspective, decentralized systems have been proposed as the best solutions to increase reliability of the application. Techniques such as federated learning allow for decentralized DL system deployments.
- AI/Combinatorial Optimization in IoT Smart Cities
    - The development of data fusion techniques that can make the use of heterogeneous data sources easier, intelligent data reduction/feature selection methods to ensure that redundant or 'uninteresting' data is not part of the

AI development pipeline. This will help in a quicker turnaround time as well as improved performance of deployments. Current methods need to be used as well as new ones be researched for making ML and DL algorithms more explainable to suit the various applications in a smart city.

  - Hybrid and novel optimization methodologies (for e.g. graywolf optimization [493]) which combine characteristics of multiple heuristic schemes could potentially outperform singular methodologies. There have been several works which combine multiple optimization techniques.
  - Reinforcement learning (RL) has the potential to provide solutions to combinatorial optimization problems as covered in [494]. The idea is to use machine learning and reinforcement learning to get rid of human created heurists which may lead to optimizations towards local optimums. Agents can be trained to search for these heuristics to automate the process.

- *Smart Health: Fall detection*

  - The addition of additional sensor modalities apart from intertial sensor measurements might help improve performance for the worst detected classes from the experiments conducted. Various authors have incorporated medical or pressure sensors in their fall detection systems. Data from these sensors can be used together as an input to a deep learning network. This additional information gathered for subjects while undergoing a fall could also describe valuable health parameters that can be used for diagnosis or early detection of ailments which might be the underlying cause of falls.
  - Another work opportunity in the data science domain would be the assessment of this problem considering a data centric approach, where, in contrast to a model centric approach where the focus is on developing the best model, data centric focuses on working with data to improve application performance using systematic procedures for labeling, augmentation etc. Such systematic procedures and qualitative data analysis would aid in cross-dataset algorithm deployment as well.

REFERENCES

[1] Worldometers, “World population forecast - worldometers,” (2019).

[2] H. Ahvenniemi, A. Huovila, I. Pinto-Seppä, and M. Airaksinen, “What are the differences between sustainable and smart cities?” Cities **60**, 234–245 (2017).

[3] United Nations, “About the Sustainable Development Goals - United Nations Sustainable Development,” .

[4] P. Cardullo and R. Kitchin, “Being a ‘citizen’ in the smart city: up and down the scaffold of smart citizen participation in dublin, ireland,” GeoJournal **84**, 1–13 (2019).

[5] J. Desdemoustier, N. Crutzen, and R. Giffinger, “Municipalities’ understanding of the smart city concept: An exploratory analysis in belgium,” Technological Forecasting and Social Change **142**, 129–141 (2019).

[6] M. S. Khan, M. Woo, K. Nam, and P. K. Chathoth, “Smart city and smart tourism: A case of dubai,” Sustainability (Switzerland) **9** (2017).

[7] D. Wu, N. Jiang, W. Du, K. Tang, and X. Cao, “Particle swarm optimization with moving particles on scale-free networks,” IEEE Transactions on Network Science and Engineering **7**, 497–506 (2018).

[8] M. L. Finlayson and E. W. Peterson, “Falls, aging, and disability,” Physical Medicine and Rehabilitation Clinics **21**, 357–373 (2010).

[9] B. M. Kistler, J. Khubchandani, G. Jakubowicz, K. Wilund, and J. Sosnoff, “Peer reviewed: Falls and fall-related injuries among us adults aged 65 or older with chronic kidney disease,” Preventing chronic disease **15** (2018).

[10] L. Bolton, “Preventing fall injury.” Wounds: a compendium of clinical research and practice **31**, 269–271 (2019).

[11] Y. K. Haddad, G. Bergen, and C. Florence, “Estimating the economic burden related to older adult falls by state,” Journal of public health management and practice: JPHMP **25**, E17 (2019).

[12] C. S. Florence, G. Bergen, A. Atherly, E. Burns, J. Stevens, and C. Drake, “Medical costs of fatal and nonfatal falls in older adults,” Journal of the American Geriatrics Society **66**, 693–698 (2018).

[13] A. Bourke, J. O’brien, and G. Lyons, “Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm,” Gait & posture **26**, 194–199 (2007).

[14] R. G. Hollands, "Will the real smart city please stand up? intelligent, progressive or entrepreneurial?" City **12**, 303–320 (2008).

[15] L. G. Anthopoulos and C. G. Reddick, "Understanding electronic government research and smart city: A framework and empirical evidence," Information Polity **21**, 99–117 (2016).

[16] Z. Khan, A. Anjum, K. Soomro, and M. A. Tahir, "Towards cloud based big data analytics for smart future cities," Journal of Cloud Computing **4** (2015).

[17] A. Koubaa, A. Aldawood, B. Saeed, A. Hadid, M. Ahmed, A. Saad, H. Alkhouja, A. Ammar, and M. Alkanhal, "Smart palm: An iot framework for red palm weevil early detection," Agronomy **10**, 1–21 (2020). No ML or dL used.

[18] M. O'Grady, D. Langton, and G. O'Hare, "Edge computing: A tractable model for smart agriculture?" Artificial Intelligence in Agriculture **3**, 42–51 (2019).

[19] I. Rojek and J. Studzinski, "Detection and localization of water leaks in water nets supported by an ict system with artificial intelligence methods as away forward for smart cities," Sustainability (Switzerland) **11** (2019).

[20] K. Pardini, J. J. Rodrigues, S. A. Kozlov, N. Kumar, and V. Furtado, "Iot-based solid waste management solutions: A survey," Journal of Sensor and Actuator Networks **8** (2019).

[21] J. Dutta, C. Chowdhury, S. Roy, A. I. Middya, and F. Gazi, "Towards smart city: Sensing air quality in city based on opportunistic crowd-sensing," ACM International Conference Proceeding Series (2017).

[22] F. Al-Turjman and A. Malekloo, "Smart parking in iot-enabled cities: A survey," Sustainable Cities and Society **49** (2019).

[23] E. Shirazi and S. Jadid, "Autonomous self-healing in smart distribution grids using multi agent systems," IEEE Transactions on Industrial Informatics **3203**, 1–11 (2018).

[24] R. V. Andreão, M. Athayde, J. Boudy, P. Aguilar, I. de Araujo, and R. Andrade, "Raspcare: A telemedicine platform for the treatment and monitoring of patients with chronic diseases," in "Assistive Technologies in Smart Cities," (IntechOpen London, UK, 2018).

[25] P. A. Keane and E. J. Topol, "With an eye to ai and autonomous diagnosis," npj Digital Medicine **1**, 10–12 (2018).

[26] G. Trencher and A. Karvonen, "Stretching "smart": advancing health and well-being through the smart city agenda," Local Environment **24**, 610–627 (2019).

[27] B. R. Haverkort and A. Zimmermann, "Smart industry: How ict will change the game!" IEEE Internet Computing **21**, 8–10 (2017).

[28] F. Tao, J. Cheng, and Q. Qi, "Iihub: An industrial internet-of-things hub toward smart manufacturing based on cyber-physical system," IEEE Transactions on Industrial Informatics **14**, 2271–2280 (2018).

[29] P. Trakadas, P. Simoens, P. Gkonis, L. Sarakis, A. Angelopoulos, A. P. Ramallo-González, A. Skarmeta, C. Trochoutsos, D. Calvo, T. Pariente, K. Chintamani, I. Fernandez, A. A. Irigaray, J. X. Parreira, P. Petrali, N. Leligou, and P. Karkazis, "An artificial intelligence-based collaboration approach in industrial iot manufacturing: Key concepts, architectural extensions and potential applications," Sensors (Switzerland) **20**, 1–20 (2020). Ni as this is about architecture of AI in to Smart industry.

[30] J. Wan, J. Yang, Z. Wang, and Q. Hua, "Artificial intelligence for cloud-assisted smart factory," IEEE Access **6**, 55419–55430 (2018).

[31] Y. Huang, Z. Dang, Y. Choi, J. Andrade, and A. Bar-ilan, "High-precision smart system on accelerometers and inclinometers for structural health monitoring: development and applications," in "2018 12th France-Japan and 10th Europe-Asia Congress on Mechatronics," (IEEE, 2018), pp. 52–57.

[32] S. G. Farag, "Application of smart structural system for smart sustainable cities," in "2019 4th MEC International Conference on Big Data and Smart City (ICBDSC)," (2019), pp. 1–5.

[33] Y. Wang, S. Ram, F. Currim, E. Dantas, and L. A. Sabóia, "A big data approach for smart transportation management on bus network," IEEE 2nd International Smart Cities Conference: Improving the Citizens Quality of Life, ISC2 2016 - Proceedings pp. 0–5 (2016).

[34] A. Lele, "Internet of things (iot)," Smart Innovation, Systems and Technologies **132**, 187–195 (2019).

[35] P. Mell and T. Grance, "The nist-national institute of standars and technology- definition of cloud computing," NIST Special Publication 800-145 p. 7 (2011).

[36] J. B.-M. Numhauser, "Fog computing introduction to a new cloud evolution," University of Alcalá (2012).

[37] M. Aazam, S. Zeadally, and K. A. Harras, "Fog computing architecture, evaluation, and future research directions," IEEE Communications Magazine **56**, 46–52 (2018).

[38] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C. T. Lin, "Edge of things: The big picture on the integration of edge, iot and the cloud in a distributed computing environment," IEEE Access **6**, 1706–1717 (2017).

[39] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," Journal of Systems Architecture **98**, 289–330 (2019).

[40] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of internet of things for smart home: Challenges and solutions," Journal of Cleaner Production **140**, 1454–1464 (2017). The authors perform a literature reiew of the architectures available for IoT in the Smart home. After doing so, they provide an architecture of their own which has been derived from the other architectures discussed. ¡br/¿¡br/¿*They provide a good description of the types of appliances.

[41] A. Saini and A. Malik, "Routing in internet of things: A survey," Communication and Computing Systems - Proceedings of the International Conference on Communication and Computing Systems, ICCCS 2016 **I**, 855–860 (2017).

[42] E. Casilari, J. A. Santoyo-Ramón, and J. M. Cano-García, "On the heterogeneity of existing repositories of movements intended for the evaluation of fall detection systems," Journal of Healthcare Engineering **2020** (2020). The authors in EC20 analyse various datasets which contain accelerometer signal measurements for people performing fall activities. Within this:¡br/¿¡br/¿Table 1 lists the datasets considered¡br/¿Table 2 illustrates the characteristics of the collected data (duration etc) and the participants¡br/¿Table 3 discusses the sampling rate and other properties of the data ¡br/¿¡br/¿The analysis involves extracting sliding time windows of 0.5 seconds from the signals and calculate the window with the maximum difference of acceleration as described in ref [60] to determine the window where the fall has taken place. Once this is done, they calculate 12 statistical features from the data.

[43] M. Rahnemoonfar and C. Sheppard, "Deep count : Fruit counting based on deep simulated learning," Sensors **17**, 1–12 (2017).

[44] D. Thakker, B. K. Mishra, A. Abdullatif, and S. Mazumdar, "Explainable artificial intelligence for developing smart cities solutions," Smart Cities **3**, 1353–1382 (2020).

[45] A. Rahman, M. S. Hossain, N. A. Alrajeh, and N. Guizani, "B5g and explainable deep learning assisted healthcare vertical at the edge : Covid-19 perspective," IEEE Network **34**, 98–105 (2020).

[46] M. Janssen, S. Luthra, S. Mangla, N. P. Rana, and Y. K. Dwivedi, "Challenges for adopting and implementing iot in smart cities," Internet Research **ahead-of-p** (2019).

[47] R. Sánchez-Corcuera, A. Nuñez-Marcos, J. Sesma-Solance, A. Bilbao-Jayo, R. Mulero, U. Zulaika, G. Azkune, and A. Almeida, "Smart cities

survey: Technologies, application domains and challenges for the cities of the future," International Journal of Distributed Sensor Networks **15** (2019).

[48] B. N. Silva, M. Khan, and K. Han, "Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities," Sustainable Cities and Society **38**, 697–713 (2018).

[49] W. Ejaz and A. Anpalagan, "Internet of things for smart cities: overview and key challenges," Internet of Things for Smart Cities pp. 1–15 (2019).

[50] G. Marques, N. Garcia, and N. Pombo, "A survey on iot: architectures, elements, applications, qos, platforms and security concepts," in "Advances in Mobile cloud computing and big data in the 5G era," (Springer, 2017), pp. 115–130.

[51] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, "Security and privacy in smart city applications: Challenges and solutions," IEEE Communications Magazine **55**, 122–129 (2017).

[52] Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani, "Internet-of-things-based smart cities: Recent advances and challenges," IEEE Communications Magazine **55**, 16–24 (2017).

[53] W. Rong, Z. Xiong, D. Cooper, C. Li, and H. Sheng, "Smart city architecture: A technology guide for implementation and design challenges," China Communications **11**, 56–69 (2014).

[54] E. Ahmed, I. Yaqoob, A. Gani, M. Imran, and M. Guizani, "Internet-of-things-based smart environments: State of the art, taxonomy, and open research challenges," IEEE Wireless Communications **23**, 10–16 (2016).

[55] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of iot: Applications, challenges, and opportunities with china perspective," IEEE Internet of Things Journal **1**, 349–359 (2014).

[56] B. Ahlgren, M. Hidell, and E. C. Ngai, "Internet of things for smart cities: Interoperability and open data," IEEE Internet Computing **20**, 52–56 (2016).

[57] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," Business Horizons **58**, 431–440 (2015).

[58] C. M. Morais, D. Sadok, and J. Kelner, "An iot sensor and scenario survey for data researchers," Journal of the Brazilian Computer Society **25** (2019).

[59] A. Sharma and M. Sinha, "A differential evolution-based routing algorithm for multi-path environment in mobile ad hoc network," International Journal of Hybrid Intelligence **1**, 23–40 (2019).

[60] G. P. Hancke, B. de Carvalho de Silva, and G. P. Hancke, *The role of advanced sensing in smart cities*, vol. 13 (Multidisciplinary Digital Publishing Institute, 2013).

[61] M. Penza, D. Suriano, M. G. Villani, L. Spinelle, and M. Gerboles, "Towards air quality indices in smart cities by calibrated low-cost sensors applied to networks," Proceedings of IEEE Sensors **2014-Decem**, 2012–2017 (2014).

[62] F. Folianto, Y. S. Low, and W. L. Yeow, "Smartbin: Smart waste management system," 2015 IEEE 10th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP 2015 pp. 1–2 (2015).

[63] G. Bedi, G. K. Venayagamoorthy, R. Singh, R. R. Brooks, and K. C. Wang, "Review of internet of things (iot) in electric power and energy systems," IEEE Internet of Things Journal **5**, 847–870 (2018).

[64] E. Y. Song, G. J. Fitzpatrick, and K. B. Lee, "Smart sensors and standard-based interoperability in smart grids," IEEE Sensors Journal **17**, 7723–7730 (2017).

[65] A. A. Abdellatif, A. Mohamed, C. F. Chiasserini, M. Tlili, and A. Erbad, "Edge computing for smart health: Context-aware approaches, opportunities, and challenges," IEEE Network **33**, 196–203 (2019).

[66] K. Fan, S. Zhu, K. Zhang, H. Li, and Y. Yang, "A lightweight authentication scheme for cloud-based rfid healthcare systems," IEEE Network **33**, 44–49 (2019).

[67] A. Prati, C. Shan, and K. I. Wang, "Sensors, vision and networks: From video surveillance to activity recognition and health monitoring," Journal of Ambient Intelligence and Smart Environments **11**, 5–22 (2019).

[68] D. Ding, R. A. Cooper, P. F. Pasquina, and L. Fici-Pasquina, "Sensor technology for smart homes," Maturitas **69**, 131–136 (2011).

[69] B. Jan, H. Farman, M. Khan, M. Talha, and I. U. Din, "Designing a smart transportation system: An internet of things and big data approach," IEEE Wireless Communications **26**, 73–79 (2019).

[70] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, "Sensor technologies for intelligent transportation systems," Sensors (Switzerland) **18**, 1–24 (2018).

[71] A. Gharaibeh, M. A. Salahuddin, S. J. Hussini, A. Khreishah, I. Khalil, M. Guizani, and A. Al-Fuqaha, "Smart cities: A survey on data management, security, and enabling technologies," IEEE Communications Surveys and Tutorials **19**, 2456–2501 (2017).

[72] S. Talari, M. Shafie-Khah, P. Siano, V. Loia, A. Tommasetti, and J. P. Catalão, "A review of smart cities based on the internet of things concept," Energies **10**, 1–23 (2017).

[73] D. M. Park, S. K. Kim, and Y. S. Seo, "S-mote: Smart home framework for common household appliances in iot network," Journal of Information Processing Systems **15**, 449–456 (2019).

[74] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, "Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges," IEEE wireless communications **24**, 10–16 (2017).

[75] P. Sakhardande, S. Hanagal, and S. Kulkarni, "Design of disaster management system using iot based interconnected network with smart city monitoring," 2016 International Conference on Internet of Things and Applications, IOTA 2016 pp. 185–190 (2016).

[76] L. Kang, S. Poslad, W. Wang, X. Li, Y. Zhang, and C. Wang, "A public transport bus as a flexible mobile smart environment sensing platform for iot," Proceedings - 12th International Conference on Intelligent Environments, IE 2016 pp. 1–8 (2016).

[77] T. Adiono, M. Y. Fathany, R. V. W. Putra, K. Afifah, M. H. Santriaji, B. L. Lawu, and S. Fuada, "Live demonstration: Minds - meshed and internet networked devices system for smart home: Track selection: Embedded systems," 2016 IEEE Asia Pacific Conference on Circuits and Systems, APCCAS 2016 pp. 736–737 (2017).

[78] A. Ghosh and N. Chakraborty, "Design of smart grid in an university campus using zigbee mesh networks," 1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems, ICPEICES 2016 pp. 1–6 (2017).

[79] Y. Yan, Y. Qian, and H. Sharif, "A secure data aggregation and dispatch scheme for home area networks in smart grid," GLOBECOM - IEEE Global Telecommunications Conference pp. 1–6 (2011).

[80] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," IEEE Communications Surveys and Tutorials **19**, 855–873 (2017).

[81] M. Kuzlu and M. Pipattanasomporn, "Assessment of communication technologies and network requirements for different smart grid applications," 2013 IEEE PES Innovative Smart Grid Technologies Conference, ISGT 2013 pp. 1–6 (2013).

[82] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, "Internet of things (iot) communication protocols: Review," ICIT 2017 - 8th International Conference on Information Technology, Proceedings pp. 685–690 (2017).

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

[84] S. S. I. Samuel, “A review of connectivity challenges in iot-smart home,” 2016 3rd MEC International Conference on Big Data and Smart City, ICBDSC 2016 pp. 364–367 (2016).

[85] P. Kuppusamy, S. Muthuraj, and S. Gopinath, “Survey and challenges of li-fi with comparison of wi-fi,” Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016 pp. 896–899 (2016).

[86] B.-S. A. Heile and M. H. T. P. C (Futurewei) Liu, B(Huawei Technologies) Zhang, “Wi-sun fan overview,” (2017).

[87] B. Hammi, R. Khatoun, S. Zeadally, A. Fayad, and L. Khoukhi, “Iot technologies for smart cities,” IET Networks **7**, 1–13 (2018).

[88] Z. A. Baig, P. Szewczyk, C. Valli, P. Rabadia, P. Hannay, M. Chernyshev, M. Johnstone, P. Kerai, A. Ibrahim, K. Sansurooah, N. Syed, and M. Peacock, “Future challenges for smart cities: Cyber-security and digital forensics,” Digital Investigation **22**, 3–13 (2017).

[89] T. Braun, B. C. Fung, F. Iqbal, and B. Shah, “Security and privacy challenges in smart cities,” Sustainable Cities and Society **39**, 499–507 (2018).

[90] D. Eckhoff and I. Wagner, “Privacy in the smart city - applications, technologies, challenges, and solutions,” IEEE Communications Surveys and Tutorials **20**, 489–516 (2018).

[91] A. S. Elmaghraby and M. M. Losavio, “Cyber security challenges in smart cities: Safety, security and privacy,” Journal of Advanced Research **5**, 491–497 (2014).

[92] D. POPESCUL and L. D. RADU, “Data security in smart cities: Challenges and solutions,” Informatica Economica **20**, 29–38 (2016).

[93] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, “A survey on iot security: Application areas, security threats, and solution architectures,” IEEE Access **7**, 82721–82743 (2019).

[94] D. E. Whitehead, K. Owens, D. Gammel, and J. Smith, “Ukraine cyber-induced power outage: Analysis and practical mitigation strategies,” 70th Annual Conference for Protective Relay Engineers, CPRE 2017 (2017).

[95] I. B. A. Ang, F. D. Salim, and M. Hamilton, “Human occupancy recognition with multivariate ambient sensors,” 2016 IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom Workshops 2016 pp. 1–6 (2016).

[96] L. Yang, K. Ting, and M. B. Srivastava, "Inferring occupancy from opportunistically available sensor data," 2014 IEEE International Conference on Pervasive Computing and Communications, PerCom 2014 pp. 60–68 (2014).

[97] J. D. Howard and T. A. Longstaff, "A common language for computer security incidents," Sandia National Laboratories (1998).

[98] R. Agrawal, P. Verma, R. Sonanis, U. Goel, A. De, S. A. Kondaveeti, and S. Shekhar, "Continuous security in iot using blockchain," ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings **2018-April**, 6423–6427 (2018).

[99] S. Rizvi, J. P. Iii, A. Kurtz, and M. Rizvi, "Securing the internet of things (iot): A security taxonomy for iot," in "2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)," (IEEE, 2018), pp. 163–168.

[100] A. Aldairi and L. Tawalbeh, "Cyber security attacks on smart cities and associated mobile technologies," Procedia Computer Science **109**, 1086–1091 (2017).

[101] M. Ali, N. Abrishamchi, A. H. Abdullah, A. D. Cheok, and K. S. Bielawski, "Side channel attacks on smart home systems : A short overview," in "IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society," (2017), pp. 8144–8149.

[102] J. Choi and Y. Kim, "An improved lea block encryption algorithm to prevent side-channel attack in the iot system," in "2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)," (Asia Pacific Signal and Information Processing Association, 2016), pp. 1–4.

[103] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (iot) security," IEEE Communications Surveys & Tutorials **22**, 1646–1685 (2020).

[104] X. Ouyang and Z. Ma, "Using lstm networks to identify false data of smart terminals in the smart grid," Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS **2017-Decem**, 765–768 (2018).

[105] A. Diro and N. Chilamkurti, "Leveraging lstm networks for attack detection in fog-to-things communications," IEEE Communications Magazine **56**, 124–130 (2018).

[106] N. Balakrishnan, A. Rajendran, D. Pelusi, and V. Ponnusamy, "Deep belief network enhanced intrusion detection system to prevent security breach in the internet of things," Internet of Things p. 100112 (2019).

[107] R. Khatoun and S. Zeadally, "Cybersecurity and privacy solutions in smart cities," IEEE Communications Magazine **55**, 51–59 (2017).

[108] H. F. Atlam, A. Alenezi, M. O. Alassafi, and G. B. Wills, "Blockchain with internet of things : Benefits , challenges , and future directions," I.J. Intelligent Systems and Applications **10**, 40–48 (2018).

[109] J. T. de Souza, A. C. de Francisco, C. M. Piekarski, and G. F. do Prado, "Data mining and machine learning to promote smart cities: A systematic review from 2000 to 2018," Sustainability (Switzerland) **11** (2019).

[110] Z. Rayan, M. Alfonse, and A. B. M. Salem, "Machine learning approaches in smart health," Procedia Computer Science **154**, 361–368 (2018).

[111] R. Varghese and S. Sharma, "Affordable smart farming using iot and machine learning," Proceedings of the 2nd International Conference on Intelligent Computing and Control Systems, ICICCS 2018 pp. 645–650 (2019).

[112] S. AlZu'bi, B. Hawashin, M. Mujahed, Y. Jararweh, and B. B. Gupta, "An efficient employment of internet of multimedia things in smart and future agriculture," Multimedia Tools and Applications **78**, 29581–29605 (2019).

[113] K. S. P. Reddy, Y. M. Roopa, L. N. K. Rajeev, and N. S. Nandan, "Iot based smart agriculture using machine learning," Proceedings of the 2nd International Conference on Inventive Research in Computing Applications, ICIRCA 2020 pp. 130–134 (2020).

[114] A. Goap, D. Sharma, A. K. Shukla, and C. R. Krishna, "An iot based smart irrigation management system using machine learning and open source technologies," Computers and Electronics in Agriculture **155**, 41–49 (2018).

[115] S. Rodríguez, T. Gualotuña, and C. Grilo, "A system for the monitoring and predicting of data in precision a agriculture system for the and predicting of wireless data in precision in a monitoring rose greenhouse based on sensor agriculture in a rose greenhouse based on wireless sensor networks ne," Procedia Computer Science **121**, 306–313 (2017).

[116] N. Kitpo, Y. Kugai, M. Inoue, T. Yokemura, and S. Satomura, "Internet of things for greenhouse monitoring system using deep learning and bot notification services," 2019 IEEE International Conference on Consumer Electronics, ICCE 2019 (2019).

[117] A. K. Saha, J. Saha, R. Ray, S. Sircar, S. Dutta, S. P. Chattopadhyay, and H. N. Saha, "Iot-based drone for improvement of crop quality in agricultural field," in "2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)," (IEEE, 2018), pp. 612–615.

[118] A. A. Araby, M. M. Abd Elhameed, N. M. Magdy, N. Abdelaal, Y. T. Abd Allah, M. S. Darweesh, M. A. Fahim, H. Mostafa *et al.*, "Smart iot monitoring system for agriculture with predictive analysis," in "2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST)," (IEEE, 2019), pp. 1–4.

[119] S. A. Nandhini and R. H. S. Radha, "Web enabled plant disease detection system for agricultural applications using wmsn," Wireless Personal Communications **102**, 725–740 (2018).

[120] L. Ale, A. Sheta, L. Li, Y. Wang, and N. Zhang, "Deep learning based plant disease detection for smart agriculture," 2019 IEEE Globecom Workshops, GC Wkshps 2019 - Proceedings pp. 1–6 (2019).

[121] F. Balducci, D. Impedovo, and G. Pirlo, "Machine learning applications on agricultural datasets for smart farm enhancement," Machines **6** (2018).

[122] X.-B. Jin, N.-X. Yang, X.-Y. Wang, Y.-T. Bai, T.-L. Su, and J.-L. Kong, "Hybrid deep learning predictor for smart agriculture sensing based on empirical mode decomposition and gated recurrent unit group model," Sensors **20**, 1334 (2020).

[123] S. A. M. Varman, A. R. Baskaran, S. Aravindh, and E. Prabhu, "Deep learning and iot for smart agriculture using wsn," 2017 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2017 pp. 0–5 (2018).

[124] X. bo Jin, X. hong Yu, X. yi Wang, Y. ting Bai, and T. li Su, "Deep learning predictor for sustainable precision agriculture based on internet of things system," Sustainability **12**, 1433 (2020).

[125] K. Aliev, M. M. Jawaid, S. Narejo, E. Pasero, and A. Pulatov, "Internet of plants application for smart agriculture," International Journal of Advanced Computer Science and Applications **9**, 421–429 (2018).

[126] A. L. Diedrichs, F. Bromberg, D. Dujovne, K. Brun-laguna, T. Watteyne, and S. Member, "Prediction of frost events using machine learning and iot sensing devices," IEEE Internet of Things Journal **5**, 4589–4597 (2018).

[127] R. Reshma, V. Sathiyavathi, T. Sindhu, K. Selvakumar, and L. SaiRamesh, "Iot based classification techniques for soil content analysis and crop yield prediction," in "2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)," (IEEE, 2020), pp. 156–160.

[128] M. Pawar and G. Chillarge, "Soil toxicity prediction and recommendation system using data mining in precision agriculture," in "2018 3rd international conference for convergence in technology (I2CT)," (IEEE, 2018), pp. 1–5.

[129] K. Lee, B. N. Silva, and K. Han, "Deep learning entrusted to fog nodes (dlefn) based smart agriculture," Applied Sciences (Switzerland) **10** (2020).

[130] M. A. Guillén, A. Llanes, B. Imbernón, R. Martínez, E. Andrés, B. Crespo, and J. Carlos, "Performance evaluation of edge - computing platforms for the prediction of low temperatures in agriculture using deep learning," The Journal of Supercomputing **77**, 818–840 (2021).

[131] M. A. Guillén-navarro, R. Martínez-españa, and J. M. Cecilia, "A high-performance iot solution to reduce frost damages in stone fruits," Concurrency and Computation: Practice and Experience **33**, e5299 (2019).

[132] M. A. Guillén, A. Llanes, B. Imbernón, R. Martínez-España, A. Bueno-Crespo, J. C. Cano, and J. M. Cecilia, "Performance evaluation of edge-computing platforms for the prediction of low temperatures in agriculture using deep learning," (2020).

[133] D. Rutqvist, D. Kleyko, and F. Blomstedt, "An automated machine learning approach for smart waste management systems," IEEE Transactions on Industrial Informatics **PP**, 1–1 (2019).

[134] A. Hussain, U. Draz, T. Ali, S. Tariq, M. Irfan, A. Glowacz, J. Alfonso, A. Daviu, S. Yasin, and S. Rahman, "Waste management and prediction of air pollutants using iot and machine learning approach," Energies **13**, 3930 (2020).

[135] D. Zhang, G. Lindholm, and H. Ratnaweera, "Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring," Journal of Hydrology **556**, 409–418 (2018).

[136] P. Varalakshmi, S. Vandhana, and S. Vishali, "Prediction of water quality using naive bayesian algorithm," 2016 8th International Conference on Advanced Computing, ICoAC 2016 pp. 224–229 (2017).

[137] D. Jalal, "Toward a Smart Real Time Monitoring System for Drinking Water Based on Machine Learning," 2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM) pp. 1–5 (2019).

[138] T. Bin, M. M. Alam, N. Absar, K. Andersson, and M. Shahadat, "Iot based real-time river water quality monitoring system," Procedia Computer Science **155**, 161–168 (2019).

[139] P. Liu, J. Wang, A. K. Sangaiah, Y. Xie, and X. Yin, "Analysis and prediction of water quality using lstm deep neural networks in iot environment," Sustainability **11**, 2058 (2019).

[140] Z. Tariq, S. K. Shah, and Y. Lee, "Smart 311 request system with automatic noise detection for safe neighborhood," 2018 IEEE International Smart Cities Conference, ISC2 2018 pp. 1–8 (2019).

[141] J. Zhang, S. I. Levitan, and J. Hirschberg, "Multimodal deception detection using automatically extracted acoustic, visual, and lexical features." in "INTERSPEECH," (2020), pp. 359–363.

[142] Y. Santur, E. Karaköse, M. Karaköse, and E. Akın, "Deep learning based artificial manager for smart city," in "5th International Conference on Advanced Technology & Sciences," (2017), pp. 197–201.

[143] S. Sanaei, B. Majidi, and E. Akhtarkavan, "Deep multisensor dashboard for composition layer of web of things in the smart city," 9th International Symposium on Telecommunication: With Emphasis on Information and Communication Technology, IST 2018 pp. 211–215 (2019).

[144] P. D. Rosero-Montalvo, J. A. Caraguay-Procel, E. D. Jaramillo, J. M. Michilena-Calderon, A. C. Umaquinga-Criollo, M. Mediavilla-Valverde, M. A. Ruiz, L. A. Beltran, and D. H. Peluffo-Ordónez, "Air quality monitoring intelligent system using machine learning techniques," Proceedings - 3rd International Conference on Information Systems and Computer Science, INCISCOS 2018 **2018-Decem**, 75–80 (2018).

[145] C. Feng, W. Wang, Y. Tian, X. Que, and X. Gong, "Estimate air quality based on mobile crowd sensing and big data," 18th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks, WoWMoM 2017 - Conference (2017).

[146] R. Yu, Y. Yang, L. Yang, G. Han, and O. A. Move, "RAQ–A random forest approach for predicting air quality in urban sensing systems," Sensors (Switzerland) **16** (2016).

[147] I. Kök, M. U. Şimşek, and S. Özdemir, "A deep learning model for air quality prediction in smart cities," Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017 **2018-Janua**, 1983–1990 (2017).

[148] A. Al-Wakeel, J. Wu, and N. Jenkins, "K -means based load estimation of domestic smart meter measurements," Applied Energy **194**, 333–342 (2017).

[149] O. Valgaev, F. Kupzog, and H. Schmeck, "Building power demand forecasting using k-nearest neighbours model - practical application in smart city demo aspern project," CIRED - Open Access Proceedings Journal **2017**, 1601–1604 (2017).

[150] P. Vrablecová, A. B. Ezzeddine, V. Rozinajová, S. Šárik, and A. K. Sangaiah, "Smart grid load forecasting using online support vector regression," Computers and Electrical Engineering **65**, 102–117 (2018).

[151] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on lstm recurrent neural network," IEEE Transactions on Smart Grid **10**, 841–851 (2019).

[152] S. Hosein and P. Hosein, "Load forecasting using deep neural networks," 2017 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference, ISGT 2017 (2017).

[153] K. Ke, S. Hongbin, Z. Chengkang, and C. Brown, "Short-term electrical load forecasting method based on stacked auto-encoding and gru neural network," Evolutionary Intelligence **12**, 385–394 (2019).

[154] T. Han, K. Muhammad, T. Hussain, J. Lloret, and S. W. Baik, "An efficient deep learning framework for intelligent energy management in iot networks," IEEE Internet of Things Journal (2020).

[155] X. Liu, Z. Xiao, R. Zhu, J. Wang, L. Liu, and M. Ma, "Edge sensing data-imaging conversion scheme of load forecasting in smart grid," Sustainable Cities and Society p. 102363 (2020).

[156] A. Taïk and S. Cherkaoui, "Electrical load forecasting using edge computing and federated learning," in "ICC 2020-2020 IEEE International Conference on Communications (ICC)," (IEEE, 2020), pp. 1–6.

[157] D. Nguyen, R. Barella, S. A. Wallace, X. Zhao, and X. Liang, "Smart grid line event classification using supervised learning over pmu data streams," 2015 6th International Green and Sustainable Computing Conference (2016).

[158] Z. Zheng, Y. Yang, X. Niu, H. N. Dai, and Y. Zhou, "Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids," IEEE Transactions on Industrial Informatics **14**, 1606–1615 (2018).

[159] Y. Huang, Y. Lu, F. Wang, X. Fan, J. Liu, and V. C. Leung, "An edge computing framework for real-time monitoring in smart grid," Proceedings - 2018 IEEE International Conference on Industrial Internet, ICII 2018 pp. 99–108 (2018).

[160] S. Batool, N. A. Saqib, and M. A. Khan, "Internet of things data analytics for user authentication and activity recognition," 2017 2nd International Conference on Fog and Mobile Edge Computing, FMEC 2017 pp. 183–187 (2017).

[161] H. Zhang, Z. Xiao, J. Wang, F. Li, and E. Szczerbicki, "A novel iot-perceptive human activity recognition (har) approach using multihead convolutional attention," IEEE Internet of Things Journal **7**, 1072–1080 (2020).

[162] D. Castro, "Wearable-based human activity recognition using an iot approach," Journal of Sensor and Actuator Networks **6**, 28 (2017).

[163] D. N. Tran and D. D. Phan, "Human activities recognition in android smartphone using support vector machine," in "Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS," , vol. 0 (IEEE, 2016), vol. 0, pp. 64–68.

[164] G. L. Santos, P. T. Endo, K. Henrique, and D. C. Monteiro, “Accelerometer-based human fall detection using,” Sensors **19**, 1–12 (2019).

[165] D. Yacchirema, “Fall detection system for elderly people using iot and ensemble machine learning algorithm,” Personal and Ubiquitous Computing **23**, 801–817 (2019).

[166] I. Machorro-Cano, G. Alor-Hernández, M. A. Paredes-Valverde, U. Ramos-Deonati, J. L. Sánchez-Cervantes, and L. Rodríguez-Mazahua, “Pisiot: A machine learning and iot-based smart health platform for overweight and obesity control,” Applied Sciences (Switzerland) **9** (2019).

[167] R. Ani, S. Krishna, N. Anju, A. M. Sona, and O. S. Deepa, “Iot based patient monitoring and diagnostic prediction tool using ensemble classifier,” 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017 **2017-Janua**, 1588–1593 (2017).

[168] A. Ukil and U. K. Roy, “Smart cardiac health management in iot through heart sound signal analytics and robust noise filtering,” IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC **2017-Octob**, 1–5 (2018).

[169] A. Alamri, “Monitoring system for patients using multimedia for smart healthcare,” IEEE Access **6**, 23271–23276 (2018).

[170] Y. Qu, X. Ming, S. Qiu, M. Zheng, and Z. Hou, “An integrative framework for online prognostic and health management using internet of things and convolutional neural network,” Sensors (Switzerland) **19** (2019).

[171] M. S. Hossain and G. Muhammad, “Cloud-assisted industrial internet of things (iiot) - enabled framework for health monitoring,” Computer Networks **101**, 192–202 (2016).

[172] M. Alhussein, G. Muhammad, M. S. Hossain, and S. U. Amin, “Cognitive iot-cloud integration for smart healthcare: Case study for epileptic seizure detection and monitoring,” Mobile Networks and Applications **23**, 1624–1635 (2018).

[173] M. Ganesan and N. Sivakumar, “Iot based heart disease prediction and diagnosis model for healthcare using machine learning models,” 2019 IEEE International Conference on System, Computation, Automation and Networking, ICSCAN 2019 pp. 1–5 (2019).

[174] S. Mohapatra, P. K. Patra, S. Mohanty, and B. Pati, “Smart health care system using data mining,” Proceedings - 2018 International Conference on Information Technology, ICIT 2018 pp. 44–49 (2018).

[175] P. Kaur, R. Kumar, and M. Kumar, “A healthcare monitoring system using random forest and internet of things (iot),” Multimedia Tools and

Applications **78**, 19905–19916 (2019). Ni, its one of those random papers where they take data and run algorithms on them.

[176] M. Alhussein, “Monitoring parkinson’s disease in smart cities,” IEEE Access **5**, 19835–19841 (2017).

[177] S. Tuli, N. Basumatary, S. S. Gill, M. Kahani, R. C. Arya, G. S. Wander, and R. Buyya, “Healthfog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated iot and fog computing environments,” Future Generation Computer Systems **104**, 187–200 (2020).

[178] M. Devarajan and L. Ravi, “Intelligent cyber-physical system for an efficient detection of parkinson disease using fog computing,” Multimedia Tools and Applications **78**, 32695–32719 (2019).

[179] M. A. Sayeed, S. P. Mohanty, E. Kougianos, V. P. Yanambaka, and H. Zaveri, “A robust and fast seizure detector for iot edge,” Proceedings - 2018 IEEE 4th International Symposium on Smart Electronic Systems, iSES 2018 pp. 156–160 (2018).

[180] J. P. Queralta, T. N. Gia, H. Tenhunen, and T. Westerlund, “Edge-ai in lora-based health monitoring: Fall detection system with fog computing and lstm recurrent neural networks,” in “2019 42nd international conference on telecommunications and signal processing (TSP),” (IEEE, 2019), pp. 601–604.

[181] M. Awais, M. Raza, N. Singh, K. Bashir, U. Manzoor, S. ul Islam, and J. J. Rodrigues, “Lstm based emotion detection using physiological signals: Iot framework for healthcare and distance learning in covid-19,” IEEE Internet of Things Journal (2020).

[182] P. Pirzada, N. White, and A. Wilde, “Sensors in smart homes for independent living of the elderly,” 5th International Multi-Topic ICT Conference: Technologies For Future Generations, IMTIC 2018 - Proceedings pp. 1–8 (2018).

[183] J. Park, K. Jang, and S. B. Yang, “Deep neural networks for activity recognition with multi-sensor data in a smart home,” IEEE World Forum on Internet of Things, WF-IoT 2018 - Proceedings **2018-Janua**, 155–160 (2018).

[184] A. Wang, G. Chen, C. Shang, M. Zhang, and L. Liu, “Human activity recognition in a smart home environment with stacked denoising autoencoders,” in “International Conference on Web-Age Information Management,” (Springer, 2016), pp. 29–40.

[185] D. Singh, E. Merdivan, I. Psychoula, J. Kropf, S. Hanke, M. Geist, and A. Holzinger, “Human activity recognition using recurrent neural

networks," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **10410 LNCS**, 267–274 (2017).

[186] G. Y. Kim, S. S. Shin, J. Y. Kim, and H. G. Kim, "Haptic Conversion Using Detected Sound Event in Home Monitoring System for the Hard-of-Hearing," HAVE 2018 - IEEE International Symposium on Haptic, Audio-Visual Environments and Games, Proceedings pp. 17–22 (2018).

[187] A. B. Adege, H. P. Lin, G. B. Tarekegn, and S. S. Jeng, "Applying deep neural network (dnn) for robust indoor localization in multi-building environment," Applied Sciences (Switzerland) **8**, 1–14 (2018).

[188] R. Adeogun, I. Rodriguez, M. Razzaghpour, G. Berardinelli, P. H. Christensen, and P. E. Mogensen, "Indoor occupancy detection and estimation using machine learning and measurements from an iot lora-based monitoring system," in "2019 Global IoT Summit (GIoTS)," (IEEE, 2019), pp. 1–5.

[189] L. Zimmermann, R. Weigel, and G. Fischer, "Fusion of nonintrusive environmental sensors for occupancy detection in smart homes," IEEE Internet of Things Journal **5**, 2343–2352 (2018).

[190] D. Chowdhry, R. Paranjape, and P. Laforge, "Smart home automation system for intrusion detection," 2015 IEEE 14th Canadian Workshop on Information Theory, CWIT 2015 pp. 75–78 (2015).

[191] V. H. Bhide and S. Wagh, "I-learning iot: An intelligent self learning system for home automation using iot," 2015 International Conference on Communication and Signal Processing, ICCSP 2015 pp. 1763–1767 (2015).

[192] F. C. C. Garcia, C. M. C. Creayla, and E. Q. B. Macabebe, "Development of an intelligent system for smart home energy disaggregation using stacked denoising autoencoders," Procedia Computer Science **105**, 248–255 (2017).

[193] C. C. Yang, C. S. Soh, and V. V. Yap, "A non-intrusive appliance load monitoring for efficient energy consumption based on naive bayes classifier," Sustainable Computing: Informatics and Systems **14**, 34–42 (2017).

[194] D. Popa, F. Pop, C. Serbanescu, and A. Castiglione, "Deep learning model for home automation and energy reduction in a smart home environment platform," Neural Computing and Applications **31**, 1317–1337 (2019).

[195] I. C. Konstantakopoulos, A. R. Barkan, S. He, T. Veeravalli, H. Liu, and C. Spanos, "A deep learning and gamification approach to improving human-building interaction and energy efficiency in smart infrastructure," Applied Energy **237**, 810–821 (2019).

[196] H. Albataineh, M. Nijim, and D. Bollampall, "The design of a novel smart home control system using smart grid based on edge and cloud computing," in "2020 IEEE 8th International Conference on Smart Energy Grid Engineering (SEGE)," (IEEE, 2020), pp. 88–91.

[197] P. H. F. Sousa, N. M. M. Nascimento, J. S. Almeida, P. P. R. Filho, and V. H. C. Albuquerque, "Intelligent incipient fault detection in wind turbines based on industrial iot environment," Journal of Artificial Intelligence and Systems **1**, 1–19 (2019).

[198] M. Salhaoui, A. Guerrero-González, M. Arioua, F. J. Ortiz, A. E. Oualkadi, and C. L. Torregrosa, "Smart industrial iot monitoring and control system based on uav and cloud computing applied to a concrete plant," Sensors (Switzerland) **19** (2019).

[199] M. A. Ferreira, L. F. F. D. Souza, F. H. S. D. Silva, E. F. Ohata, J. S. Almeida, and P. P. Filho, "Intelligent industrial iot system for detection of short-circuit failure in windings of wind turbines," Proceedings of the International Joint Conference on Neural Networks (2020).

[200] T. Nkonyana, Y. Sun, B. Twala, and E. Dogo, "Performance evaluation of data mining techniques in steel manufacturing industry," Procedia Manufacturing **35**, 623–628 (2019).

[201] J. E. Siegel, S. Pratt, Y. Sun, and S. E. Sarma, "Real-time deep neural networks for internet-enabled arc-fault detection," Engineering Applications of Artificial Intelligence **74**, 35–42 (2018).

[202] J. Windau and L. Itti, "Inertial machine monitoring system for automated failure detection," Proceedings - IEEE International Conference on Robotics and Automation pp. 93–98 (2018).

[203] A. N. Sokolov, I. A. Pyatnitsky, and S. K. Alabugin, "Research of Classical Machine Learning Methods and Deep Learning Models Effectiveness in Detecting Anomalies of Industrial Control System," Proceedings - 2018 Global Smart Industry Conference, GloSIC 2018 (2018).

[204] D. C. Huang, C. F. Lin, C. Y. Chen, and J. R. Sze, "The internet technology for defect detection system with deep learning method in smart factory," 2018 4th International Conference on Information Management, ICIM 2018 pp. 98–102 (2018).

[205] L. Li, K. Ota, and M. Dong, "Deep learning for smart industry: Efficient manufacture inspection system with fog computing," IEEE Transactions on Industrial Informatics **14**, 4665–4673 (2018).

[206] S. Y. Lin, Y. Du, P. C. Ko, T. J. Wu, P. T. Ho, V. Sivakumar, and R. subbareddy, "Fog computing based hybrid deep learning framework in effective inspection system for smart manufacturing," Computer Communications **160**, 636–642 (2020).

[207] M. S. S. Garmaroodi, F. Farivar, M. S. Haghighi, M. A. Shoorehdeli, and A. Jolfaei, "Detection of anomalies in industrial iot systems by data mining: Study of christ osmotron water purification system," IEEE Internet of Things Journal **4662**, 1–1 (2020).

[208] D. Park, S. Kim, Y. An, and J. Y. Jung, "Lired: A light-weight real-time fault detection system for edge computing using lstm recurrent neural networks," Sensors (Switzerland) **18** (2018).

[209] Y. Liu, N. Kumar, Z. Xiong, W. Y. B. Lim, J. Kang, and D. Niyato, "Communication-efficient federated learning for anomaly detection in industrial internet of things," in "GLOBECOM," , vol. 2020 (2020), vol. 2020, pp. 1–6.

[210] H. Zheng, Y. Feng, Y. Gao, and J. Tan, "A robust predicted performance analysis approach for data-driven product development in the industrial internet of things," Sensors (Switzerland) **18** (2018).

[211] A. Essien and C. Giannetti, "A deep learning model for smart manufacturing using convolutional lstm neural network autoencoders," IEEE Transactions on Industrial Informatics **16**, 6069–6078 (2020).

[212] W. Tao, Z.-H. Lai, M. C. Leu, and Z. Yin, "Worker activity recognition in smart manufacturing using imu and semg signals with convolutional neural networks," Procedia Manufacturing **26**, 1159–1166 (2018).

[213] A. R. M. Forkan, F. Montori, D. Georgakopoulos, P. P. Jayaraman, A. Yavari, and A. Morshed, "An industrial iot solution for evaluating workers' performance via activity recognition," in "2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)," (IEEE, 2019), pp. 1393–1403.

[214] B. Huang, W. Wang, S. Ren, R. Y. Zhong, and J. Jiang, "A proactive task dispatching method based on future bottleneck prediction for the smart factory," International Journal of Computer Integrated Manufacturing **32**, 278–293 (2019). Not included as it is a hybrid.

[215] B. Brik, B. Bettayeb, M. Sahnoun, and F. Duval, "Towards predicting system disruption in industry 4.0: Machine learning-based approach," Procedia Computer Science **151**, 667–674 (2019).

[216] Y. Han, C. J. Zhang, L. Wang, and Y. C. Zhang, "Industrial IoT for Intelligent Steelmaking with Converter Mouth Flame Spectrum Information Processed by Deep Learning," IEEE Transactions on Industrial Informatics **16**, 2640–2650 (2020).

[217] J. Wang, Y. Wang, and M. Yu, "A multi-period ambulance location and allocation problem in the disaster," Journal of Combinatorial Optimization **9**, 1–24 (2020). This paper presents a survey about fall detection technologies and discusses how sensor fusion is one of the most popular trends, both in terms of data as well as algorithm.

[218] L. Spendla, M. Kebisek, P. Tanuska, and L. Hrcka, "Concept of predictive maintenance of production systems in accordance with industry 4.0," in "2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)," (IEEE, 2017), pp. 405–410.

[219] S. Hwang, J. Jeong, and Y. Kang, "Svm-rbm based predictive maintenance scheme for iot-enabled smart factory," 2018 13th International Conference on Digital Information Management, ICDIM 2018 pp. 162–167 (2018).

[220] E.-J. K. J.-H. Kwon, "Failure prediction model using iterative feature selection for industrial internet of things," Symmetry pp. 1–15 (2019).

[221] W. Zhang, W. Guo, X. Liu, Y. Liu, J. Zhou, B. Li, Q. Lu, and S. Yang, "Lstm-based analysis of industrial iot equipment," IEEE Access **6**, 23551–23560 (2018).

[222] A. Diez, N. L. D. Khoa, M. M. Alamdari, Y. Wang, F. Chen, and P. Runcie, "A clustering approach for structural health monitoring on bridges," Journal of Civil Structural Health Monitoring **6**, 429–445 (2016).

[223] J. Guo, X. Xie, and R. Bie, "Structural health monitoring by using a sparse coding-based deep learning algorithm with wireless sensor networks," Personal and ubiquitous computing **18**, 1977–1987 (2014).

[224] G. Gui, H. Pan, Z. Lin, Y. Li, and Z. Yuan, "Data-driven support vector machine with optimization techniques for structural health monitoring and damage detection," KSCE Journal of Civil Engineering **21**, 523–534 (2017).

[225] H. V. Dang, H. Tran-ngoc, T. V. Nguyen, G. D. Roeck, H. X. Nguyen, and S. Member, "Data-driven structural health monitoring using feature fusion and hybrid deep learning," IEEE Transactions on Automation Science and Engineering pp. 1–17 (2020).

[226] J. Vitola, F. Pozo, D. A. Tibaduiza, and M. Anaya, "A sensor data fusion system based on k-nearest neighbor pattern classification for structural health monitoring applications," Sensors (Switzerland) **17** (2017).

[227] J. Yu, M. Kim, H. C. Bang, S. H. Bae, and S. J. Kim, "IoT as a applications: cloud-based building management systems for the internet of things," Multimedia Tools and Applications **75**, 14583–14596 (2016).

[228] S. Singaravel, P. Geyer, and J. Suykens, "Deep neural network architectures for component-based machine learning model in building energy predictions," Digital Proceedings of the 24th EG-ICE International Workshop on Intelligent Computing in Engineering 2017 pp. 260–268 (2017).

[229] N. A. Eltresy, O. M. Dardeer, A. Al-Habal, E. Elhariri, A. H. Hassan, A. Khattab, D. N. Elsheakh, S. A. Taie, H. Mostafa, H. A. Elsadek, and

E. A. Abdallah, “Rf energy harvesting iot system for museum ambience control with deep learning,” Sensors (Switzerland) **19** (2019).

[230] W. Hu, Y. Wen, K. Guan, G. Jin, and K. J. Tseng, “itcm : Toward learning-based thermal comfort modeling via pervasive sensing for smart buildings,” IEEE Internet of Things Journal **5**, 4164–4177 (2018).

[231] H. Mora and A. Jimeno-morenilla, “Deployment of iot edge and fog computing technologies to develop smart building services,” Sustainability **10**, 1–23 (2018).

[232] D. H. Stolfi, E. Alba, and X. Yao, “Predicting car park occupancy rates in smart cities,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **10268 LNCS**, 107–117 (2017).

[233] R. K. Lenka, R. K. Barik, N. K. Das, K. Agarwal, D. Mohanty, and S. Vipsita, “Psps: An iot based predictive smart parking system,” 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics, UPCON 2017 **2018-Janua**, 311–317 (2017).

[234] G. Ali, T. Ali, M. Irfan, U. Draz, M. Sohail, A. Glowacz, M. Sulowicz, R. Mielnik, Z. B. Faheem, and C. Martis, “Iot based smart parking system using deep long short memory network,” Electronics **9**, 1696 (2020).

[235] T. Ebuchi and H. Yamamoto, “Vehicle/pedestrian localization system using multiple radio beacons and machine learning for smart parking,” in “2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC),” (IEEE, 2019), pp. 086–091.

[236] F. M. Awan, Y. Saleem, R. Minerva, and N. Crespi, “A comparative analysis of machine / deep learning models for parking space availability prediction,” Sensors **20**, 322 (2020).

[237] X. Sevillano, E. Màrmol, and V. Fernandez-Arguedas, “Towards smart traffic management systems: Vacant on-street parking spot detection based on video analytics,” FUSION 2014 - 17th International Conference on Information Fusion (2014).

[238] H. Bura, N. Lin, N. Kumar, S. Malekar, S. Nagaraj, and K. Liu, “An edge based smart parking solution using camera networks and deep learning,” in “2018 IEEE International Conference on Cognitive Computing (ICCC),” (IEEE, 2018), pp. 17–24.

[239] C. Lee, S. Park, T. Yang, and S.-H. Lee, “Smart parking with fine-grained localization and user status sensing based on edge computing,” in “2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall),” (IEEE, 2019), pp. 1–5.

[240] S. S. Aung and T. T. Naing, “Naïve bayes classifier based traffic prediction system on cloud infrastructure,” Proceedings - International Conference

on Intelligent Systems, Modelling and Simulation, ISMS **2015-Octob**, 193–198 (2015).

[241] Y. Liu and H. Wu, "Prediction of road traffic congestion based on random forest," Proceedings - 2017 10th International Symposium on Computational Intelligence and Design, ISCID 2017 **2**, 361–364 (2018).

[242] W. Wei, H. Wu, and H. Ma, "An autoencoder and lstm-based traffic flow prediction method," Sensors **19**, 1–16 (2019).

[243] Y. Xiao, "Hybrid lstm neural network for short-term traffic flow prediction," Information **10**, 105 (2019).

[244] F. Sun, Y. Pan, J. White, and A. Dubey, "Real-Time and Predictive Analytics for Smart Public Transportation Decision Support System," 2016 IEEE International Conference on Smart Computing, SMARTCOMP 2016 pp. 1–8 (2016).

[245] N. Dogru and A. Subasi, "Traffic accident detection using random forest classifier," 2018 15th Learning and Technology Conference, L and T 2018 pp. 40–45 (2018).

[246] D. Nallaperuma, R. Nawaratne, T. Bandaragoda, A. Adikari, and S. Nguyen, "Online incremental machine learning platform for big data-driven smart traffic management," IEEE Transactions on Intelligent Transportation Systems **20**, 4679–4690 (2019).

[247] A. Mukherji, V. Raghuram, S. Pandey, H. Tran, and N. Bulusu, "Commuterscanner: Towards smart indoor positioning systems in urban transportation," 2019 11th International Conference on Communication Systems and Networks, COMSNETS 2019 **2061**, 619–624 (2019).

[248] S. Majumdar, M. M. Subhani, B. Roullier, A. Anjum, and R. Zhu, "Congestion prediction for smart sustainable cities using iot and machine learning approaches," Sustainable Cities and Society **64**, 102500 (2021).

[249] F. Peres and M. Castelli, "Combinatorial optimization problems and metaheuristics: Review, challenges, design, and development," Applied Sciences **11**, 6449 (2021).

[250] M. Dorigo and K. Socha, "Ant colony optimization," Handbook of Approximation Algorithms and Metaheuristics pp. 26–1–26–14 (2007).

[251] D. Whitley, "A genetic algorithm tutorial," Statistics and computing **4**, 65–85 (1994).

[252] J. Kennedy and R. Eberhart, "Particle swarm optimization," in "Proceedings of ICNN'95-international conference on neural networks," , vol. 4 (IEEE, 1995), vol. 4, pp. 1942–1948.

[253] V. Feoktistov, *Differential evolution* (Springer, 2006).

[254] D. Karaboga, "Artificial bee colony algorithm," scholarpedia **5**, 6915 (2010).

[255] D. C. H. Nguyen, J. C. Ascough II, H. R. Maier, G. C. Dandy, and A. A. Andales, "Optimization of irrigation scheduling using ant colony algorithms and an advanced cropping system model," Environmental modelling & software **97**, 32–45 (2017).

[256] B. Saeidian, M. S. Mesgari, B. Pradhan, and A. M. Alamri, "Irrigation water allocation at farm level based on temporal cultivation-related data using meta-heuristic optimisation algorithms," Water **11**, 2611 (2019).

[257] D. Guo, J. E. Olesen, K. Manevski, and X. Ma, "Optimizing irrigation schedule in a large agricultural region under different hydrologic scenarios," Agricultural Water Management **245**, 106575 (2021).

[258] S. Guo, F. Zhang, C. Zhang, C. An, S. Wang, and P. Guo, "A multi-objective hierarchical model for irrigation scheduling in the complex canal system," Sustainability **11**, 24 (2019).

[259] F. Zhang, C. He, F. Yaqiong, X. Hao, and S. Kang, "Canal delivery and irrigation scheduling optimization based on crop water demand," Agricultural Water Management **260**, 107245 (2022).

[260] A. Ikudayisi, J. Adeyemo, J. Odiyo, and A. Enitan, "Optimum irrigation water allocation and crop distribution using combined pareto multi-objective differential evolution," Cogent Engineering **5**, 1535749 (2018).

[261] J. Wu, R. Huang, T. Fang, and Y. Han, "Multi-objective model of irrigation water distribution based on particle swarm optimization," in "IOP Conference Series: Earth and Environmental Science," , vol. 344 (IOP Publishing, 2019), vol. 344, p. 012087.

[262] A. L. P. De Ocampo and E. P. Dadios, "Energy cost optimization in irrigation system of smart farm by using genetic algorithm," in "2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)," (IEEE, 2017), pp. 1–7.

[263] L. Zhuo, J. Cheng, and Z. Gong, "Optimal water allocation method based on the genetic algorithm for a system of a reservoir and two pumping stations," Water Supply **22**, 849–859 (2022).

[264] R. Li, Y. Chang, and Z. Wang, "Study of optimal allocation of water resources in dujiangyan irrigation district of china based on an improved genetic algorithm," Water Supply **21**, 2989–2999 (2021).

[265] S. K. Roy and D. De, "Genetic algorithm based internet of precision agricultural things (iopat) for agriculture 4.0," Internet of Things p. 100201 (2020).

[266] N. Lin, X. Wang, Y. Zhang, X. Hu, and J. Ruan, "Fertigation management for sustainable precision agriculture based on internet of things," Journal of Cleaner Production **277**, 124119 (2020).

[267] C. Arif, B. I. Setiawan, M. Mizoguchi, and B. D. A. Nugroho, "Genetic algorithms optimization for water management in irrigated paddy fields," in "IOP Conference Series: Earth and Environmental Science," , vol. 335 (IOP Publishing, 2019), vol. 335, p. 012002.

[268] WorldBank, "Solid waste management," (2019).

[269] S. Sharmin and S. T. Al-Amin, "A cloud-based dynamic waste management system for smart cities," Proceedings of the 7th Annual Symposium on Computing for Development, ACM DEV-7 2016 (2016).

[270] R. Fujdiak, P. Masek, P. Mlynek, J. Misurec, and E. Olshannikova, "Using genetic algorithm for advanced municipal waste collection in smart city," 2016 10th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2016 pp. 1–6 (2016).

[271] R. Assaf and Y. Saleh, "Vehicle-routing optimization for municipal solid waste collection using genetic algorithm: the case of southern nablus city," Civil and environmental engineering reports (2017).

[272] M. A. Hannan, M. S. H. Lipu, M. Akhtar, R. A. Begum, M. A. A. Mamum, A. Hussain, M. S. Mia, and H. Basri, "Solid waste collection optimization objectives, constraints, modeling approaches, and their challenges toward achieving sustainable development goals," Journal of cleaner production p. 123557 (2020).

[273] Y. Zhang, X. Luo, X. Han, Y. Lu, J. Wei, and C. Yu, "Optimization of urban waste transportation route based on genetic algorithm," Security and Communication Networks **2022**, 1–10 (2022).

[274] Q. Wei, Z. Guo, H. C. Lau, and Z. He, "An artificial bee colony-based hybrid approach for waste collection problem with midway disposal pattern," Applied Soft Computing **76**, 629–637 (2019).

[275] S. P. Raflesia and A. K. Pamosoaji, "A novel ant colony optimization algorithm for waste collection problem," in "2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)," (IEEE, 2019), pp. 413–416.

[276] M. Yazdani, K. Kabirifar, B. E. Frimpong, M. Shariati, M. Mirmozaffari, and A. Boskabadi, "Improving construction and demolition waste collection service in an urban area using a simheuristic approach: A case study in sydney, australia," Journal of Cleaner Production **280**, 124138 (2021).

[277] S. Idwan, I. Mahmood, J. A. Zubairi, and I. Matar, "Optimal management of solid waste in smart cities using internet of things," Wireless Personal Communications **110**, 485–501 (2020).

[278] U. M. Faizal, R. Jayachitra, P. Vijayakumar, and M. Rajasekar, "Optimization of inbound vehicle routes in the collection of bio-medical wastes," Materials Today: Proceedings **45**, 692–699 (2021).

[279] E. B. Tirkolaee, M. Alinaghian, A. A. R. Hosseinabadi, M. B. Sasi, and A. K. Sangaiah, "An improved ant colony optimization for the multi-trip capacitated arc routing problem," Computers and Electrical Engineering **77**, 457–470 (2019).

[280] "Branch and cut for vehicle routing," .

[281] "Capacitated vrp instances — vehicle routing problem," .

[282] M. Ettappan, V. Vimala, S. Ramesh, and V. T. Kesavan, "Optimal reactive power dispatch for real power loss minimization and voltage stability enhancement using artificial bee colony algorithm," Microprocessors and Microsystems **76**, 103085 (2020).

[283] C. K. Das, O. Bass, G. Kothapalli, T. S. Mahmoud, and D. Habibi, "Optimal placement of distributed energy storage systems in distribution networks using artificial bee colony algorithm," Applied energy **232**, 212–228 (2018).

[284] W. S. Sakr, R. A. El-Sehiemy, and A. M. Azmy, "Adaptive differential evolution algorithm for efficient reactive power management," Applied Soft Computing **53**, 336–351 (2017).

[285] T. T. Nguyen and F. Mohammadi, "Optimal placement of tcsc for congestion management and power loss reduction using multi-objective genetic algorithm," Sustainability **12**, 2813 (2020).

[286] I. I. Atteya, H. A. Ashour, N. Fahmi, and D. Strickland, "Distribution network reconfiguration in smart grid system using modified particle swarm optimization," 2016 IEEE International Conference on Renewable Energy Research and Applications, ICRERA 2016 **5**, 305–313 (2016).

[287] N. Kanwar, N. Gupta, K. R. Niazi, A. Swarnkar, and R. C. Bansal, "Simultaneous allocation of distributed energy resource using improved particle swarm optimization," Applied Energy **185**, 1684–1693 (2017).

[288] K. Utkarsh, A. Trivedi, D. Srinivasan, and T. Reindl, "A consensus-based distributed computational intelligence technique for real-time optimal control in smart distribution grids," IEEE Transactions on Emerging Topics in Computational Intelligence **1**, 51–60 (2016).

[289] A. Askarzadeh, "A memory-based genetic algorithm for optimization of power generation in a microgrid," IEEE transactions on sustainable energy **9**, 1081–1089 (2017).

[290] O. H. Mohammed, Y. Amirat, and M. Benbouzid, "Economical evaluation and optimal energy management of a stand-alone hybrid energy system handling in genetic algorithm strategies," Electronics **7**, 233 (2018).

[291] B. K. Das, R. Hassan, M. S. H. K. Tushar, F. Zaman, M. Hasan, and P. Das, "Techno-economic and environmental assessment of a hybrid renewable energy system using multi-objective genetic algorithm: A case study for remote island in bangladesh," Energy Conversion and Management **230**, 113823 (2021).

[292] D. K. Geleta and M. S. Manshahia, "Artificial bee colony-based optimization of hybrid wind and solar renewable energy system," Research Anthology on Clean Energy Management and Solutions pp. 819–842 (2021).

[293] M. M. Moghaddam, M. Marzband, and F. Azarinejadian, "Optimal energy management for a home microgrid based on multi-period artificial bee colony," in "2017 Iranian Conference on Electrical Engineering (ICEE)," (IEEE, 2017), pp. 1446–1451.

[294] H. U. R. Habib, S. Wang, A. Waqar, B. S. Farhan, K. M. Kotb, and Y.-S. Kim, "Combined heat and power units sizing and energy cost optimization of a residential building by using an artificial bee colony algorithm," IEEE Access **8**, 218289–218303 (2020).

[295] F. Lezama, J. Soares, R. Faia, T. Pinto, and Z. Vale, "A new hybrid-adaptive differential evolution for a smart grid application under uncertainty," in "2018 IEEE Congress on Evolutionary Computation (CEC)," (IEEE, 2018), pp. 1–8.

[296] F. Lezama, L. E. Sucar, E. M. de Cote, J. Soares, and Z. Vale, "Differential evolution strategies for large-scale energy resource management in smart grids," in "Proceedings of the Genetic and Evolutionary Computation Conference Companion," (2017), pp. 1279–1286.

[297] V. Palakonda, N. H. Awad, R. Mallipeddi, M. Z. Ali, K. C. Veluvolu, and P. N. Suganthan, "Differential evolution with stochastic selection for uncertain environments: A smart grid application," in "2018 IEEE Congress on Evolutionary Computation (CEC)," (IEEE, 2018), pp. 1–7.

[298] S. Mandal and K. K. Mandal, "Optimal energy management of microgrids under environmental constraints using chaos enhanced differential evolution," Renewable Energy Focus **34**, 129–141 (2020).

[299] M. Azaza and F. Wallin, "Multi objective particle swarm optimization of hybrid micro-grid system: A case study in sweden," Energy **123**, 108–118 (2017).

[300] H. A. Gabbar, M. R. Abdussami, and M. I. Adham, "Optimal planning of nuclear-renewable micro-hybrid energy system by particle swarm optimization," IEEE Access **8**, 181049–181073 (2020).

[301] B. A. Bhayo, H. H. Al-Kayiem, S. I. U. Gilani, and F. B. Ismail, "Power management optimization of hybrid solar photovoltaic-battery integrated with pumped-hydro-storage system for standalone electricity generation," Energy Conversion and Management **215**, 112942 (2020).

[302] L. F. Grisales-Noreña, O. D. Montoya, and C. A. Ramos-Paja, "An energy management system for optimal operation of bss in dc distributed generation environments based on a parallel pso algorithm," Journal of Energy Storage **29**, 101488 (2020).

[303] M. Mahdavi, A. Kimiyaghalam, H. H. Alhelou, M. Javadi, J. P. S. Catalão, and A. Ashouri, "Transmission expansion planning considering power losses, expansion of substations and uncertainty in fuel price using discrete artificial bee colony algorithm," IEEE Access (2021).

[304] T. K. Maji and P. Acharjee, "Multiple solutions of optimal pmu placement using exponential binary pso algorithm for smart grid applications," IEEE Transactions on Industry Applications **53**, 2550–2559 (2017).

[305] M. B. Rasheed and M. D. R-Moreno, "Minimizing pricing policies based on user load profiles and residential demand responses in smart grids," Applied Energy **310**, 118492 (2022).

[306] D. Zhang, Z. Fu, and L. Zhang, "An improved ts algorithm for loss-minimum reconfiguration in large-scale distribution systems," Electric Power Systems Research **77**, 685–694 (2007).

[307] J. P. Pell, J. M. Sirel, A. K. Marsden, I. Ford, and S. M. Cobbe, "Effect of reducing ambulance response times on deaths from out of hospital cardiac arrest: cohort study," Bmj **322**, 1385–1388 (2001).

[308] RapidSOS, "Outcomes: Quantifying the impact of emergency response time - rapidsos," (2015).

[309] P. Toth and D. Vigo, *The vehicle routing problem* (SIAM, 2002).

[310] A. A. Ageev and M. I. Sviridenko, "Approximation algorithms for maximum coverage and max cut with given sizes of parts," in "International Conference on Integer Programming and Combinatorial Optimization," (Springer, 1999), pp. 17–30.

[311] E. Mouhcine, Y. Karouani, K. Mansouri, and Y. Mohamed, "Toward a distributed strategy for emergency ambulance routing problem," in "2018 4th International Conference on Optimization and Applications (ICOA)," (IEEE, 2018), pp. 1–4.

[312] L. Brotcorne, G. Laporte, and F. Semet, "Ambulance location and relocation models," European journal of operational research **147**, 451–463 (2003).

[313] M. Benabdouallah, C. Bojji, and O. E. Yaakoubi, "Deployment and redeployment of ambulances using a heuristic method and an ant colony optimization - case study," Proceedings - 2016 3rd International Conference on Systems of Collaboration, SysCo 2016 pp. 1–4 (2017).

[314] M. Benabdouallah and C. Bojji, “Comparison between ga and aco for emergency coverage problem in a smart healthcare environment,” in “Proceedings of the 2017 International Conference on Smart Digital Environment,” (2017), pp. 48–55.

[315] Y. A. Kochetov and N. B. Shamray, “Optimization of the ambulance fleet location and relocation,” Journal of Applied and Industrial Mathematics **15**, 234–252 (2021).

[316] Y. Yan, Y. Kong, and Z. Fu, “Dynamic resource scheduling in emergency environment,” Journal of Information Hiding and Privacy Protection **1**, 143 (2019).

[317] Q. Lu and K.-D. Kim, “A genetic algorithm approach for expedited crossing of emergency vehicles in connected and autonomous intersection traffic,” Journal of Advanced Transportation **2017** (2017).

[318] H. M. Amer, H. A. Al-Kashoash, A. Kemp, L. Mihaylova, and M. Mayfield, “Coalition game for emergency vehicles re-routing in smart cities,” Proceedings of the IEEE Sensor Array and Multichannel Signal Processing Workshop **2018-July**, 306–310 (2018).

[319] B. N. Silva and K. Han, “Mutation operator integrated ant colony optimization based domestic appliance scheduling for lucrative demand side management,” Future generation computer systems **100**, 557–568 (2019).

[320] K. N. Bui, I. E. Agbehadji, R. Millham, D. Camacho, and J. J. Jung, “Distributed artificial bee colony approach for connected appliances in smart home energy management system,” Expert Systems **37**, e12521 (2020).

[321] S. N. Makhadmeh, A. T. Khader, M. A. Al-Betar, S. Naim, Z. A. A. Alyasseri, and A. K. Abasi, “Particle swarm optimization algorithm for power scheduling problem using smart battery,” 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology, JEEIT 2019 - Proceedings pp. 672–677 (2019).

[322] S. Abid, A. Zafar, R. Khalid, S. Javaid, U. Qasim, Z. A. Khan, and N. Javaid, “Managing energy in smart homes using binary particle swarm optimization,” in “Conference on Complex, Intelligent, and Software Intensive Systems,” (Springer, 2017), pp. 189–196.

[323] I. Fatima, A. Khalid, S. Zahoor, A. Yasmeen, S. Arif, U. Zafar, and N. Javaid, “Home energy management system using ant colony optimization technique in microgrid,” in “International Conference on Broadband and Wireless Computing, Communication and Applications,” (Springer, 2017), pp. 267–279. Use ACO for appliance scheduling in a microgrid.

[324] N. U. Rehman, N. Javaid, and Z. A. Khan, “An enhanced differential evolution based energy management system for smart grids,” in “2017

31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)," (IEEE, 2017), pp. 132–137.

[325] I. Gupta, G. N. Anandini, and M. Gupta, "An hour wise device scheduling approach for demand side management in smart grid using particle swarm optimization," 2016 National Power Systems Conference, NPSC 2016 pp. 1–6 (2017).

[326] M. A. Nasab, M. Zand, M. Eskandari, P. Sanjeevikumar, and P. Siano, "Optimal planning of electrical appliance of residential units in a smart home network using cloud services," Smart Cities **4**, 1173–1195 (2021).

[327] I. O. Essiet, Y. Sun, and Z. Wang, "Optimized energy consumption model for smart home using improved differential evolution algorithm," Energy **172**, 354–365 (2019).

[328] I. Ullah and D. Kim, "An improved optimization function for maximizing user comfort with minimum energy consumption in smart homes," Energies **10** (2017).

[329] R. Reghukumar, S. Sambhu, and V. R. Pandi, "Multi-objective optimization for efficient home energy management system using differential evolution algorithm," in "2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)," (IEEE, 2018), pp. 1157–1162.

[330] T. Pamulapati, R. Mallipeddi, and M. Lee, "Multi-objective home appliance scheduling with implicit and interactive user satisfaction modelling," Applied Energy **267**, 114690 (2020).

[331] A. R. Jordehi, "Binary particle swarm optimisation with quadratic transfer function: A new binary optimisation algorithm for optimal scheduling of appliances in smart homes," Applied Soft Computing **78**, 465–480 (2019).

[332] P. Chandra, A. Das, C. Das, A. Naskar, B. Ganguly, and S. Paul, "Differential evolution algorithm based energy management of residential microgrid under appliance scheduling dsm," in "2020 IEEE VLSI DEVICE CIRCUIT AND SYSTEM (VLSI DCS)," (IEEE, 2020), pp. 50–55.

[333] R. Faia, P. Faria, Z. Vale, and J. Spinola, "Demand response optimization using particle swarm algorithm considering optimum battery energy storage schedule in a residential house," Energies **12**, 1645 (2019).

[334] D. Madathil, V. R. Pandi, K. Ilango, and M. G. Nair, "Differential evolution based energy management system for zero net energy building," in "2017 International Conference on Technological Advancements in Power and Energy (TAP Energy)," (IEEE, 2017), pp. 1–5.

[335] H. Swalehe and B. Marungsri, "Intelligent algorithm for optimal load management in smart home appliance scheduling in distribution system," in "2018 International Electrical Engineering Congress (iEECON)," (IEEE, 2018), pp. 1–4.

[336] J.-C. Chang and T.-H. Wu, "Demand side management of power for time-of-use pricing based on particle swarm optimization," in "2020 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII)," (IEEE, 2020), pp. 317–320.

[337] C. Bharathi, D. Rekha, and V. Vijayakumar, "Genetic algorithm based demand side management for smart grid," Wireless Personal Communications **93**, 481–502 (2017).

[338] Z. Chen, Y. Chen, R. He, J. Liu, M. Gao, and L. Zhang, "Multi-objective residential load scheduling approach for demand response in smart grid," Sustainable Cities and Society **76**, 103530 (2022).

[339] W. Tao, Z. H. Lai, M. C. Leu, and Z. Yin, "Worker activity recognition in smart manufacturing using imu and semg signals with convolutional neural networks," Procedia Manufacturing **26**, 1159–1166 (2018).

[340] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," Mechanical Systems and Signal Processing **115**, 213–237 (2019).

[341] B. Huang, W. Wang, S. Ren, R. Y. Zhong, and J. Jiang, "A proactive task dispatching method based on future bottleneck prediction for the smart factory," International Journal of Computer Integrated Manufacturing **32**, 278–293 (2019).

[342] M. Sadeghi, R. Tavakkoli-Moghaddam, and R. Babazadeh, "An efficient artificial bee colony algorithm for a p-hub covering location problem with travel time reliability." International Journal of Industrial Engineering **25** (2018).

[343] K. Guo, "Research on location selection model of distribution network with constrained line constraints based on genetic algorithm," Neural Computing and Applications **32**, 1679–1689 (2020).

[344] Y. Su, J. Liu, X. Xiang, and X. Zhang, "A responsive ant colony optimization for large-scale dynamic vehicle routing problems via pheromone diversity enhancement," Complex and Intelligent Systems pp. 1–16 (2021).

[345] M. Alinaghian, M. Ghazanfari, N. Norouzi, and H. Nouralizadeh, "A novel model for the time dependent competitive vehicle routing problem: Modified random topology particle swarm optimization," Networks and Spatial Economics **17**, 1185–1211 (2017).

[346] D. M. Utama, T. A. Fitria, and A. K. Garside, "Artificial bee colony algorithm for solving green vehicle routing problems with time windows,"

in "Journal of Physics: Conference Series," , vol. 1933 (IOP Publishing, 2021), vol. 1933, p. 012043.

[347] M. Ibrahim, F. Nurhakiki, D. Utama, and A. Rizaki, "Optimised genetic algorithm crossover and mutation stage for vehicle routing problem pick-up and delivery with time windows," in "IOP Conference Series: Materials Science and Engineering," , vol. 1071 (IOP Publishing, 2021), vol. 1071, p. 012025.

[348] D. A. Mounia and D. Bachir, "A hybrid discrete artificial bee colony for the green pickup and delivery problem with time windows," Informatica **44** (2020).

[349] Z. Gu, Y. Zhu, Y. Wang, X. Du, M. Guizani, and Z. Tian, "Applying artificial bee colony algorithm to the multidepot vehicle routing problem," Software: Practice and Experience (2020).

[350] N. Norouzi, M. Sadegh-Amalnick, and R. Tavakkoli-Moghaddam, "Modified particle swarm optimization in a time-dependent vehicle routing problem: minimizing fuel consumption," Optimization Letters **11**, 121–134 (2017).

[351] K. K. H. Ng, C. K. M. Lee, S. Z. Zhang, K. Wu, and W. Ho, "A multiple colonies artificial bee colony algorithm for a capacitated vehicle routing problem and re-routing strategies under time-dependent traffic congestion," Computers and Industrial Engineering **109**, 151–168 (2017).

[352] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," Operations research **35**, 254–265 (1987).

[353] Y. Huang, S. A. Ludwig, and F. Deng, "Sensor optimization using a genetic algorithm for structural health monitoring in harsh environments," Journal of Civil Structural Health Monitoring **6**, 509–519 (2016).

[354] F. Zhao, H. Bao, S. Xue, and Q. Xu, "Multi-objective particle swarm optimization of sensor distribution scheme with consideration of the accuracy and the robustness for deformation reconstruction," Sensors **19**, 1306 (2019).

[355] C. Yang, K. Liang, X. Zhang, and X. Geng, "Sensor placement algorithm for structural health monitoring with redundancy elimination model based on sub-clustering strategy," Mechanical Systems and Signal Processing **124**, 369–387 (2019).

[356] A. Downey, C. Hu, and S. Laflamme, "Optimal sensor placement within a hybrid dense sensor network using an adaptive genetic algorithm with learning gene pool," Structural Health Monitoring **17**, 450–460 (2018).

[357] B. Cao, X. Kang, J. Zhao, P. Yang, Z. Lv, and X. Liu, "Differential evolution-based 3-d directional wireless sensor network deployment optimization," IEEE Internet of Things Journal **5**, 3594–3605 (2018).

[358] R. Marks, A. Clarke, C. A. Featherston, and R. Pullin, "Optimization of acousto-ultrasonic sensor networks using genetic algorithms based on experimental and numerical data sets," International Journal of Distributed Sensor Networks **13**, 1550147717743702 (2017).

[359] H. Liu, X. He, and Y. Jiao, "Damage identification algorithm of hinged joints for simply supported slab bridges based on modified hinge plate method and artificial bee colony algorithms," Algorithms **11**, 198 (2018).

[360] H. Tran-Ngoc, S. Khatir, G. D. Roeck, T. Bui-Tien, L. Nguyen-Ngoc, and M. A. Wahab, "Model updating for nam o bridge using particle swarm optimization algorithm and genetic algorithm," Sensors **18**, 4131 (2018).

[361] K. Gao, Y. Zhang, A. Sadollah, and R. Su, "Improved artificial bee colony algorithm for solving urban traffic light scheduling problem," in "2017 IEEE Congress on Evolutionary Computation (CEC)," (IEEE, 2017), pp. 395–402.

[362] R. F. Adebiyi, K. A. Abubilal, M. B. Mu'azu, and B. H. Adebiyi, "Development and simulation of adaptive traffic light controller using artificial bee colony algorithm," International Journal of Intelligent Systems and Applications **10**, 68–74 (2018).

[363] T. Mao, A.-S. Mihaita, and C. Cai, "Traffic signal control optimization under severe incident conditions using genetic algorithm," arXiv preprint arXiv:1906.05356 (2019).

[364] C. Tang, S. Xia, C. Zhu, and X. Wei, "Phase timing optimization for smart traffic control based on fog computing," IEEE Access **7**, 84217–84228 (2019).

[365] Z. Li, M. Shahidehpour, S. Bahramirad, and A. Khodaei, "Optimizing traffic signal settings in smart cities," IEEE Transactions on Smart Grid **8**, 2382–2393 (2017).

[366] X. Chen and Z. Yuan, "Environmentally friendly traffic control strategy-a case study in xi'an city," Journal of Cleaner Production **249**, 119397 (2020).

[367] E. Korkmaz and A. P. AKGÜNGÖR, "Delay estimation models for signalized intersections using differential evolution algorithm," Journal of Engineering Research **5** (2017).

[368] X. Zhang, X. Fan, S. Yu, A. Shan, S. Fan, Y. Xiao, and F. Dang, "Intersection signal timing optimization: A multi-objective evolutionary algorithm," Sustainability 2022, Vol. 14, Page 1506 **14**, 1506 (2022).

[369] M. Wang, H. Dong, X. Li, L. Song, and D. Pang, "A novel parking system designed for smart cities," Proceedings - 2017 Chinese Automation Congress, CAC 2017 **2017-Janua**, 3429–3434 (2017).

[370] I. Aydin, M. Karakose, and E. Karakose, "A navigation and reservation based smart parking platform using genetic optimization for smart cities," ICSG 2017 - 5th International Istanbul Smart Grids and Cities Congress and Fair pp. 120–124 (2017).

[371] F. Ferdous and M. S. Mahmud, "Intelligent traffic monitoring system using vanet infrastructure and ant colony optimization," 2016 5th International Conference on Informatics, Electronics and Vision, ICIEV 2016 pp. 356–360 (2016). The authors in FF16 use ACO in a smart transportation system to efficient route vehicles using VANETS.

[372] F. Pompei, "Ant colony optimisation and geolocation technologies for the transportation assignment problem," Proceedings - International Computer Software and Applications Conference **2**, 749–753 (2017). The authors in FP17 use ACO for planning of public transport routes for Rome using Phone data from transit users.

[373] A. Rehman, M. M. Rathore, A. Paul, F. Saeed, and R. W. Ahmad, "Vehicular traffic optimisation and even distribution using ant colony in smart city environment," IET Intelligent Transport Systems **12**, 594–601 (2018).

[374] A. Jovanović and D. Teodorović, "Fixed-time traffic control at superstreet intersections by bee colony optimization," Transportation Research Record p. 03611981211058104 (2021).

[375] T.-H. Nguyen and J. J. Jung, "Ant colony optimization-based traffic routing with intersection negotiation for connected vehicles," Applied Soft Computing **112**, 107828 (2021).

[376] R. Pitakaso, K. Sethanan, and N. Srijaroon, "Modified differential evolution algorithms for multi-vehicle allocation and route optimization for employee transportation," Engineering Optimization **52**, 1225–1243 (2020).

[377] Y. Yi, K. Choi, and Y. J. Lee, "Optimal limited-stop bus routes selection using a genetic algorithm and smart card data," Journal of Public Transportation **19**, 178–198 (2016).

[378] A. T. Buba and L. S. Lee, "Differential evolution with improved subroute reversal repair mechanism for multiobjective urban transit routing problem," Numerical Algebra, Control and Optimization **8**, 351 (2018).

[379] X. Mao, "Study on ant colony optimization algorithm for "one-day tour" traffic line," Cluster Computing **22**, 3673–3680 (2019).

[380] A. Jovanović, M. Nikolić, and D. Teodorović, "Area-wide urban traffic control: A bee colony optimization approach," Transportation Research Part C: Emerging Technologies **77**, 329–350 (2017).

[381] K. Hassoune, W. Dachry, F. Moutaouakkil, and H. Medromi, "Dynamic parking guidance architecture using ant colony optimization and multi-agent systems," Journal of Advances in Information Technology **11**, 58–63 (2020).

[382] U. Nations, "World population prospects: the 2017 revision, key findings and advance tables," Department of Economics and Social Affairs PD, editor. New York: United Nations (2017).

[383] WHO, "Falls : Key facts," (2018).

[384] R. Afable, B. Averbeck, K. Holmen, R. Dziedzicki, A. Nichols, J. Schlegelmilch, and D. Caruso, "When i'm 64: How boomers will change health care," (2007).

[385] M. K. James, M. C. Victor, S. M. Saghir, and P. A. Gentile, "Characterization of fall patients: Does age matter?" Journal of Safety Research **64**, 83–92 (2018).

[386] J. A. Stevens, M. F. Ballesteros, K. A. Mack, R. A. Rudd, E. DeCaro, and G. Adler, "Gender differences in seeking care for falls in the aged medicare population," American journal of preventive medicine **43**, 59–62 (2012).

[387] J. Morisod and M. Coutaz, "Post-fall syndrome: how to recognize and treat it?" Revue Médicale Suisse **3**, 2531–2 (2007).

[388] K. Chaccour, R. Darazi, A. H. El Hassani, and E. Andres, "From fall detection to fall prevention: A generic classification of fall-related systems," IEEE Sensors Journal **17**, 812–822 (2016).

[389] A. L. S. D. Lima, L. J. W. Evers, T. Hahn, L. Bataille, J. L. Hamilton, M. A. Little, Y. Okuma, B. R. Bloem, and M. J. Faber, "Freezing of gait and fall detection in parkinson's disease using wearable sensors: a systematic review," Journal of neurology **264**, 1642–1654 (2017).

[390] E. E. Geertsema, G. H. Visser, M. A. Viergever, and S. N. Kalitzin, "Automated remote fall detection using impact features from video and audio," Journal of biomechanics **88**, 25–32 (2019).

[391] P. Tsinganos and A. Skodras, "A smartphone-based fall detection system for the elderly," in "Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis," (IEEE, 2017), pp. 53–58.

[392] N. El Halabi, R. A. Z. Daou, R. Achkar, A. Hayek, and J. Börcsök, "Monitoring system for prediction and detection of epilepsy seizure," in "2019 Fourth International Conference on Advances in Computational Tools for Engineering Applications (ACTEA)," (IEEE, 2019), pp. 1–7.

[393] J. Fleming and C. Brayne, "Inability to get up after falling, subsequent time on floor, and summoning help: prospective cohort study in people over 90," Bmj **337** (2008).

[394] K. Ozcan, A. K. Mahabalagiri, M. Casares, and S. Velipasalar, "Automatic fall detection and activity classification by a wearable embedded smart camera," IEEE journal on emerging and selected topics in circuits and systems **3**, 125–136 (2013).

[395] A. Ramachandran and A. Karuppiah, "A survey on recent advances in wearable fall detection systems," BioMed research international **2020** (2020).

[396] Q. Guan, C. Li, X. Guo, and B. Shen, "Infrared signal based elderly fall detection for in-home monitoring," in "2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)," , vol. 1 (IEEE, 2017), vol. 1, pp. 373–376.

[397] A. Yazar, F. Erden, and A. E. Cetin, "Multi-sensor ambient assisted living system for fall detection," in "Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'14)," (2014), pp. 1–3.

[398] S. M. Adnan, A. Irtaza, S. Aziz, M. O. Ullah, A. Javed, and M. T. Mahmood, "Fall detection through acoustic local ternary patterns," Applied Acoustics **140**, 296–300 (2018).

[399] S. Palipana, D. Rojas, P. Agrawal, and D. Pesch, "Falldefi: Ubiquitous fall detection using commodity wi-fi devices," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **1**, 1–25 (2018).

[400] I. Halima, J.-M. Laferté, G. Cormier, A.-J. Fougère, and J.-L. Dillenseger, "Depth and thermal information fusion for head tracking using particle filter in a fall detection context," Integrated Computer-Aided Engineering pp. 1–14 (2020).

[401] Q. Han, H. Zhao, W. Min, H. Cui, X. Zhou, K. Zuo, and R. Liu, "A two-stream approach to fall detection with mobilevgg," IEEE Access **8**, 17556–17566 (2020).

[402] J. Zhang, C. Wu, and Y. Wang, "Human fall detection based on body posture spatio-temporal evolution," Sensors **20**, 946 (2020).

[403] R. Gupta, P. Anand, S. Chaudhury, B. Lall, and S. Singh, "Compressive sensing based privacy for fall detection," arXiv preprint arXiv:2001.03463 (2020).

[404] R. Espinosa, H. Ponce, S. Gutiérrez, L. Martínez-Villaseñor, J. Brieva, and E. Moya-Albor, "Application of convolutional neural networks for fall detection using multiple cameras," in "Challenges and Trends in Multimodal Fall Detection for Healthcare," (Springer, 2020), pp. 97–120.

[405] M. T. Pourazad, A. Shojaei-Hashemi, P. Nasiopoulos, M. Azimi, M. Mak, J. Grace, D. Jung, and T. Bains, "A non-intrusive deep learning based fall

detection scheme using video cameras," in "2020 International Conference on Information Networking (ICOIN)," (IEEE, 2020), pp. 443–446.

[406] Y. Nizam and M. M. A. Jamil, "A novel approach for human fall detection and fall risk assessment," in "Challenges and Trends in Multimodal Fall Detection for Healthcare," (Springer, 2020), pp. 237–259.

[407] Z. Liu, M. Yang, Y. Yuan, and K. Y. Kan, "Fall detection and personnel tracking system using infrared array sensors," IEEE Sensors Journal (2020).

[408] A. Singh, S. U. Rehman, S. Yongchareon, and P. H. J. Chong, "Sensor technologies for fall detection systems: A review," IEEE Sensors Journal **20**, 6889–6919 (2020).

[409] M. Hubl, O. Pohl, V. Noack, P. Hahlweg, C. Ehm, M. Derleh, T. Weiland, E. Schick, H. Müller, D. Hampicke *et al.*, "Embedding of wearable electronics into smart sensor insole," in "2016 IEEE 18th Electronics Packaging Technology Conference (EPTC)," (IEEE, 2016), pp. 597–601.

[410] T. Shi, X. Sun, Z. Xia, L. Chen, and J. Liu, "Fall detection algorithm based on triaxial accelerometer and magnetometer." Engineering Letters **24** (2016).

[411] S. Hwang, M. Ryu, Y. Yang, and N. Lee, "Fall detection with three-axis accelerometer and magnetometer in a smartphone," in "Proceedings of the International Conference on Computer Science and Technology, Jeju, Korea," (2012), pp. 25–27.

[412] X. Xi, M. Tang, S. M. Miran, and Z. Luo, "Evaluation of feature extraction and recognition for activity monitoring and fall detection based on wearable semg sensors," Sensors **17**, 1229 (2017).

[413] P. Melillo, R. Castaldo, G. Sannino, A. Orrico, G. De Pietro, and L. Pecchia, "Wearable technology and ecg processing for fall risk assessment, prevention and detection," in "2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)," (IEEE, 2015), pp. 7740–7743.

[414] K.-C. Liu, C.-Y. Hsieh, S. J.-P. Hsu, and C.-T. Chan, "Impact of sampling rate on wearable-based fall detection systems based on machine learning models," IEEE Sensors Journal **18**, 9882–9890 (2018).

[415] J. Xu, Z. He, and Y. Zhang, "Cnn-lstm combined network for iot enabled fall detection applications," in "Journal of Physics: Conference Series," , vol. 1267 (IOP Publishing, 2019), vol. 1267, p. 012044.

[416] D. Mrozek, A. Koczur, and B. Małysiak-Mrozek, "Fall detection in older adults with mobile iot devices and machine learning in the cloud and on the edge," Information Sciences (2020).

[417] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, "Sisfall: A fall and movement dataset," Sensors **17**, 198 (2017).

[418] L. Márquez-Ordaz and H. Ponce, "Implementation of a svm on an embedded system: A case study on fall detection," in "Mexican International Conference on Artificial Intelligence," (Springer, 2020), pp. 76–87.

[419] D. Sarabia-Jácome, R. Usach, C. Palau, and M. Esteve, "Highly-efficient fog-based deep learning aal fall detection system," Internet of Things p. 100185 (2020).

[420] N. Zurbuchen, P. Bruegger, and A. Wilde, "A comparison of machine learning algorithms for fall detection using wearable sensors," in "2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)," (IEEE, 2020), pp. 427–431.

[421] A. Chelli and M. Pätzold, "A machine learning approach for fall detection and daily living activity recognition," IEEE Access **7**, 38670–38687 (2019).

[422] O. Ojetola, E. Gaura, and J. Brusey, "Data set for fall events and daily activities from inertial sensors," in "Proceedings of the 6th ACM multimedia systems conference," (2015), pp. 243–248.

[423] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones." in "Esann," , vol. 3 (2013), vol. 3, p. 3.

[424] O. Kerdjidj, N. Ramzan, K. Ghanem, A. Amira, and F. Chouireb, "Fall detection and human activity classification using wearable sensors and compressed sensing," Journal of Ambient Intelligence and Humanized Computing **11**, 349–361 (2020).

[425] A. Burns, B. R. Greene, M. J. McGrath, T. J. O'Shea, B. Kuris, S. M. Ayer, F. Stroiescu, and V. Cionca, "Shimmer™–a wireless sensor platform for noninvasive biomedical research," IEEE Sensors Journal **10**, 1527–1534 (2010).

[426] M. Fañez, J. R. Villar, E. de la Cal, J. Sedano, and V. M. González, "Transfer learning and information retrieval applied to fall detection," Expert Systems **37**, e12522 (2020).

[427] E. Casilari, J. A. Santoyo-Ramón, and J. M. Cano-García, "Analysis of a smartphone-based architecture with multiple mobility sensors for fall detection," PLoS one **11**, e0168069 (2016).

[428] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in "Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery," (2003), pp. 2–11.

[429] D. Giuffrida, G. Benetti, D. D. Martini, and T. Facchinetti, "Fall detection with supervised machine learning using wearable sensors," in "2019 IEEE 17th International Conference on Industrial Informatics (INDIN)," , vol. 1 (IEEE, 2019), vol. 1, pp. 253–259.

[430] E. Casilari, M. Álvarez-Marco, and F. García-Lagos, "A study of the use of gyroscope measurements in wearable fall detection systems," Symmetry **12**, 649 (2020).

[431] G. L. Santos, P. T. Endo, K. H. d. C. Monteiro, E. d. S. Rocha, I. Silva, and T. Lynn, "Accelerometer-based human fall detection using convolutional neural networks," Sensors **19**, 1644 (2019).

[432] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," Computer methods and programs in biomedicine **117**, 489–501 (2014).

[433] T. R. Mauldin, M. E. Canby, V. Metsis, A. H. Ngu, and C. C. Rivera, "Smartfall: A smartwatch-based fall detection system using deep learning," Sensors **18**, 3363 (2018).

[434] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556 (2014).

[435] A. Alarifi and A. Alwadain, "Killer heuristic optimized convolution neural network-based fall detection with wearable iot sensor devices," Measurement **167**, 108258 (2021).

[436] M. Waheed, H. Afzal, and K. Mehmood, "Nt-fds—a noise tolerant fall detection system using deep learning on wearable devices," Sensors **21**, 2006 (2021).

[437] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, and C. Peñafort-Asturiano, "Up-fall detection dataset: A multimodal approach," Sensors **19**, 1988 (2019).

[438] E. Casilari, R. Lora-Rivera, and F. García-Lagos, "A study on the application of convolutional neural networks to fall detection evaluated with multiple public datasets," Sensors **20**, 1466 (2020).

[439] R. Delgado-Escano, F. M. Castro, J. R. Cózar, M. J. Marín-Jiménez, N. Guil, and E. Casilari, "A cross-dataset deep learning-based classifier for people fall detection and identification," Computer methods and programs in biomedicine **184**, 105265 (2020).

[440] C. Medrano, R. Igual, I. Plaza, and M. Castro, "Detecting falls as novelties in acceleration patterns acquired with smartphones," PloS one **9**, e94811 (2014).

[441] D. Micucci, M. Mobilio, and P. Napoletano, "Unimib shar: A dataset for human activity recognition using acceleration data from smartphones," Applied Sciences **7**, 1101 (2017).

[442] A. T. Özdemir, "An analysis on sensor locations of the human body for wearable fall detection devices: Principles and practice," Sensors **16**, 1161 (2016).

[443] M. Tolkiehn, L. Atallah, B. Lo, and G.-Z. Yang, "Direction sensitive fall detection using a triaxial accelerometer and a barometric pressure sensor," in "2011 Annual international conference of the IEEE engineering in medicine and biology society," (IEEE, 2011), pp. 369–372.

[444] B. Watanapa, O. Patsadu, P. Dajpratham, and C. Nukoolkit, "Post-fall intelligence supporting fall severity diagnosis using kinect sensor," Applied Computational Intelligence and Soft Computing **2018** (2018).

[445] H. Jung, B. Koo, J. Kim, T. Kim, Y. Nam, and Y. Kim, "Enhanced algorithm for the detection of preimpact fall for wearable airbags," Sensors **20**, 1277 (2020).

[446] B. Koo, J. Kim, Y. Nam, and Y. Kim, "The performance of post-fall detection using the cross-dataset: Feature vectors, classifiers and processing conditions," Sensors **21**, 4638 (2021).

[447] C.-Y. Hsieh, H.-Y. Huang, K.-C. Liu, C.-P. Liu, C.-T. Chan, and S. J.-P. Hsu, "Multiphase identification algorithm for fall recording systems using a single wearable inertial sensor," Sensors **21**, 3302 (2021).

[448] M. Musci, D. De Martini, N. Blago, T. Facchinetti, and M. Piastra, "Online fall detection using recurrent neural networks," arXiv preprint arXiv:1804.04976 (2018).

[449] X. Yu, H. Qiu, and S. Xiong, "A novel hybrid deep neural network to predict pre-impact fall for older people based on wearable inertial sensors," Frontiers in bioengineering and biotechnology **8** (2020).

[450] W. Lee, T. S. Song, and J.-H. Youn, "Detection of fall direction using a velocity vector in the android smartphone environment," Journal of the Korea Institute of Information and Communication Engineering **19**, 336–342 (2015).

[451] W. Lee, T. S. Song, and J.-H. Youn, "Fall direction detection using the components of acceleration vector and orientation sensor on the smartphone environment," Journal of Korea Multimedia Society **18**, 565–574 (2015).

[452] J. K. Lee, "Determination of fall direction before impact using support vector machine," Journal of Sensor Science and Technology **24**, 47–53 (2015).

[453] S. F. Hossain, M. Z. Islam, and M. L. Ali, "Real time direction-sensitive fall detection system using accelerometer and learning classifier," in "2017 4th International Conference on Advances in Electrical Engineering (ICAEE)," (IEEE, 2017), pp. 99–104.

[454] F. Hossain, M. L. Ali, M. Z. Islam, and H. Mustafa, "A direction-sensitive fall detection system using single 3d accelerometer and learning classifier," in "2016 International Conference on Medical Engineering, Health Informatics and Technology (MediTec)," (IEEE, 2016), pp. 1–6.

[455] L. Palmerini, J. Klenk, C. Becker, and L. Chiari, "Accelerometer-based fall detection using machine learning: Training and testing on real-world falls," Sensors **20**, 6479 (2020).

[456] R. M. Gibson, A. Amira, N. Ramzan, P. C. de-la Higuera, and Z. Pervez, "Multiple comparator classifier framework for accelerometer-based fall detection and diagnostic," Applied Soft Computing **39**, 94–103 (2016).

[457] F. Hussain, F. Hussain, M. Ehatisham-ul Haq, and M. A. Azam, "Activity-aware fall detection and recognition based on wearable sensors," IEEE Sensors Journal **19**, 4528–4536 (2019).

[458] H. Li, A. Shrestha, H. Heidari, J. Le Kernec, and F. Fioranelli, "Bi-lstm network for multimodal continuous human activity recognition and fall detection," IEEE Sensors Journal **20**, 1191–1201 (2019).

[459] X. Wu, Y. Zheng, C.-H. Chu, L. Cheng, and J. Kim, "Applying deep learning technology for automatic fall detection using mobile sensors," Biomedical Signal Processing and Control **72**, 103355 (2022).

[460] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis, "The mobiact dataset: Recognition of activities of daily living using smartphones." in "ICT4AgeingWell," (2016), pp. 143–151.

[461] G. Şengül, M. Karakaya, S. Misra, O. O. Abayomi-Alli, and R. Damaševičius, "Deep learning based fall detection using smartwatches for healthcare applications," Biomedical Signal Processing and Control **71**, 103242 (2022).

[462] H.-L. Le, D.-N. Nguyen, T.-H. Nguyen, and H.-N. Nguyen, "A novel feature set extraction based on accelerometer sensor data for improving the fall detection system," Electronics **11**, 1030 (2022).

[463] H. Yhdego, J. Li, C. Paolini, and M. Audette, "Wearable sensor gait analysis of fall detection using attention network," in "2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)," (IEEE, 2021), pp. 3137–3141.

[464] M. Liu, S. Ren, S. Ma, J. Jiao, Y. Chen, Z. Wang, and W. Song, "Gated transformer networks for multivariate time series classification," arXiv preprint arXiv:2103.14438 (2021).

[465] X. Yu, J. Jang, and S. Xiong, "A large-scale open motion dataset (kfall) and benchmark algorithms for detecting pre-impact fall of the elderly using wearable inertial sensors," Frontiers in Aging Neuroscience p. 399 (2021).

[466] J. A. Santoyo-Ramón, E. Casilari, and J. M. Cano-García, "A study of one-class classification algorithms for wearable fall sensors," Biosensors **11** (2021).

[467] N. Zurbuchen, A. Wilde, and P. Bruegger, "A machine learning multi-class approach for fall detection systems based on wearable sensors with a study on sampling rates selection," Sensors **21**, 938 (2021).

[468] M. Musci, D. De Martini, N. Blago, T. Facchinetti, and M. Piastra, "Online fall detection using recurrent neural networks on smart wearable devices," IEEE Transactions on Emerging Topics in Computing **9**, 1276–1289 (2020).

[469] E. Boutellaa, K. Ghanem, H. Tayakout, O. Kerdjidj, F. Harizi, and S. Bourennane, "A tensor approach for activity recognition and fall detection using wearable inertial sensors," in "2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH)," (IEEE, 2020), pp. 203–207.

[470] S. Rosati, G. Balestra, and M. Knaflitz, "Comparison of different sets of features for human activity recognition by wearable sensors," Sensors **18**, 4189 (2018).

[471] J. O. Smith, *Mathematics of the discrete Fourier transform (DFT): with audio applications* (Julius Smith, 2007).

[472] M. A. O'Reilly, W. Johnston, C. Buckley, D. Whelan, and B. Caulfield, "The influence of feature selection methods on exercise classification with inertial measurement units," in "2017 IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks (BSN)," (IEEE, 2017), pp. 193–196.

[473] A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest," R news **2**, 18–22 (2002).

[474] O. Rioul and M. Vetterli, "Wavelets and signal processing," IEEE signal processing magazine **8**, 14–38 (1991).

[475] M. G. Abdu-Aguye and W. Gomaa, "Competitive feature extraction for activity recognition based on wavelet transforms and adaptive pooling," in "2019 International Joint Conference on Neural Networks (IJCNN)," (IEEE, 2019), pp. 1–8.

[476] Ü. Lepik and H. Hein, "Haar wavelets," in "Haar Wavelets," (Springer, 2014), pp. 7–20.

[477] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE transactions on pattern analysis and machine intelligence **37**, 1904–1916 (2015).

[478] C. N. Silla and A. A. Freitas, “A survey of hierarchical classification across different application domains,” Data Mining and Knowledge Discovery **22**, 31–72 (2011).

[479] S. B. Khojasteh, J. R. Villar, C. Chira, V. M. González, and E. De la Cal, “Improving fall detection using an on-wrist wearable accelerometer,” Sensors **18**, 1350 (2018).

[480] Q. T. Huynh and B. Q. Tran, “Time-frequency analysis of daily activities for fall detection,” Signals **2**, 1–12 (2021).

[481] J. Wang, X. Zhang, Q. Gao, X. Ma, X. Feng, and H. Wang, “Device-free simultaneous wireless localization and activity recognition with wavelet feature,” IEEE Transactions on Vehicular Technology **66**, 1659–1669 (2016).

[482] C. Ellouzi and M. Trkov, “Fast trip detection using continuous wavelet transform,” in “Proceedings of the Poster, Dynamic Walking Conference,” (2021).

[483] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai *et al.*, “Recent advances in convolutional neural networks,” Pattern Recognition **77**, 354–377 (2018).

[484] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, “Cnn variants for computer vision: History, architecture, application, challenges and future scope,” Electronics **10**, 2470 (2021).

[485] S. P. Singh, L. Wang, S. Gupta, H. Goli, P. Padmanabhan, and B. Gulyás, “3d deep learning on medical images: a review,” Sensors **20**, 5097 (2020).

[486] S. Latif, R. Rana, S. Khalifa, R. Jurdak, J. Qadir, and B. W. Schuller, “Survey of deep representation learning for speech emotion recognition,” IEEE Transactions on Affective Computing (2021).

[487] F. G. Praticò, R. Fedele, V. Naumov, and T. Sauer, “Detection and monitoring of bottom-up cracks in road pavement using a machine-learning approach,” Algorithms **13**, 81 (2020).

[488] K. Saho, S. Hayashi, M. Tsuyama, L. Meng, and M. Masugi, “Machine learning-based classification of human behaviors and falls in restroom via dual doppler radar measurements,” Sensors **22**, 1721 (2022).

[489] J. Kakarla, B. V. Isunuri, K. S. Doppalapudi, and K. S. R. Bylapudi, “Three-class classification of brain magnetic resonance images using average-pooling convolutional neural network,” International Journal of Imaging Systems and Technology **31**, 1731–1740 (2021).

[490] J. Yang, F. Xie, H. Fan, Z. Jiang, and J. Liu, “Classification for dermoscopy images using convolutional neural networks based on region average pooling,” IEEE Access **6**, 65130–65138 (2018).

[491] M. H. Mohd Noor, S. Y. Tan, and M. N. Ab Wahab, "Deep temporal conv-lstm for activity recognition," Neural Processing Letters pp. 1–23 (2022).

[492] M. I. Amara, A. Akkouche, E. Boutellaa, and H. Tayakout, "A smartphone application for fall detection using accelerometer and convlstm network," in "2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)," (IEEE, 2021), pp. 92–96.

[493] J. Du, Z. Zhang, M. Li, J. Guo, and K. Zhu, "Optimal scheduling of integrated energy system based on improved grey wolf optimization algorithm," Scientific Reports **12**, 1–19 (2022).

[494] N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev, "Reinforcement learning for combinatorial optimization: A survey," Computers & Operations Research **134**, 105400 (2021).

Appendix A: Sample Plots for different fall categories

Sample plots of the windowed accelerometer and gyroscope measurements of different fall types.

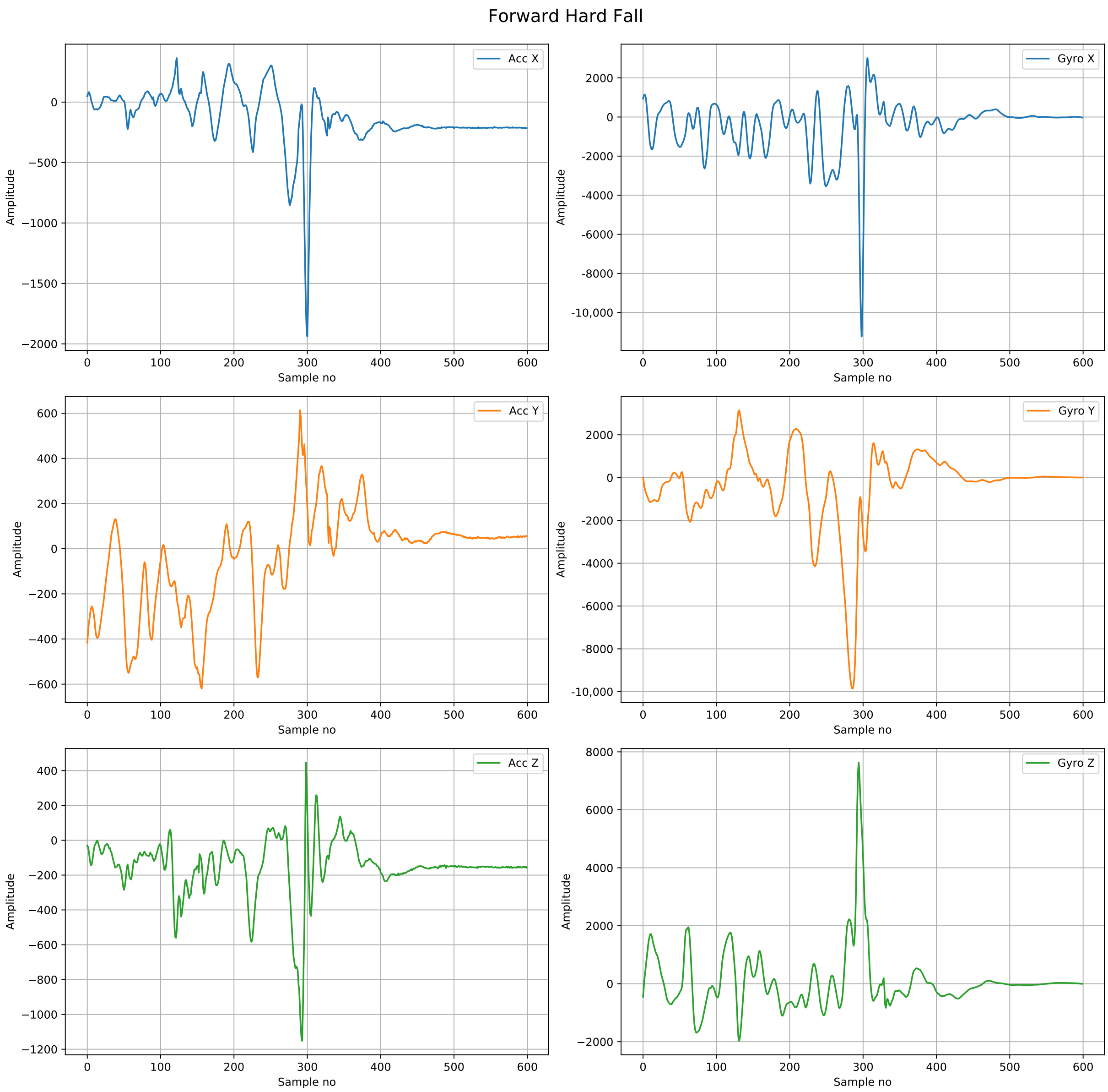

Figure 1. Accelerometer and Gyroscope measurements: Forward Hard Fall

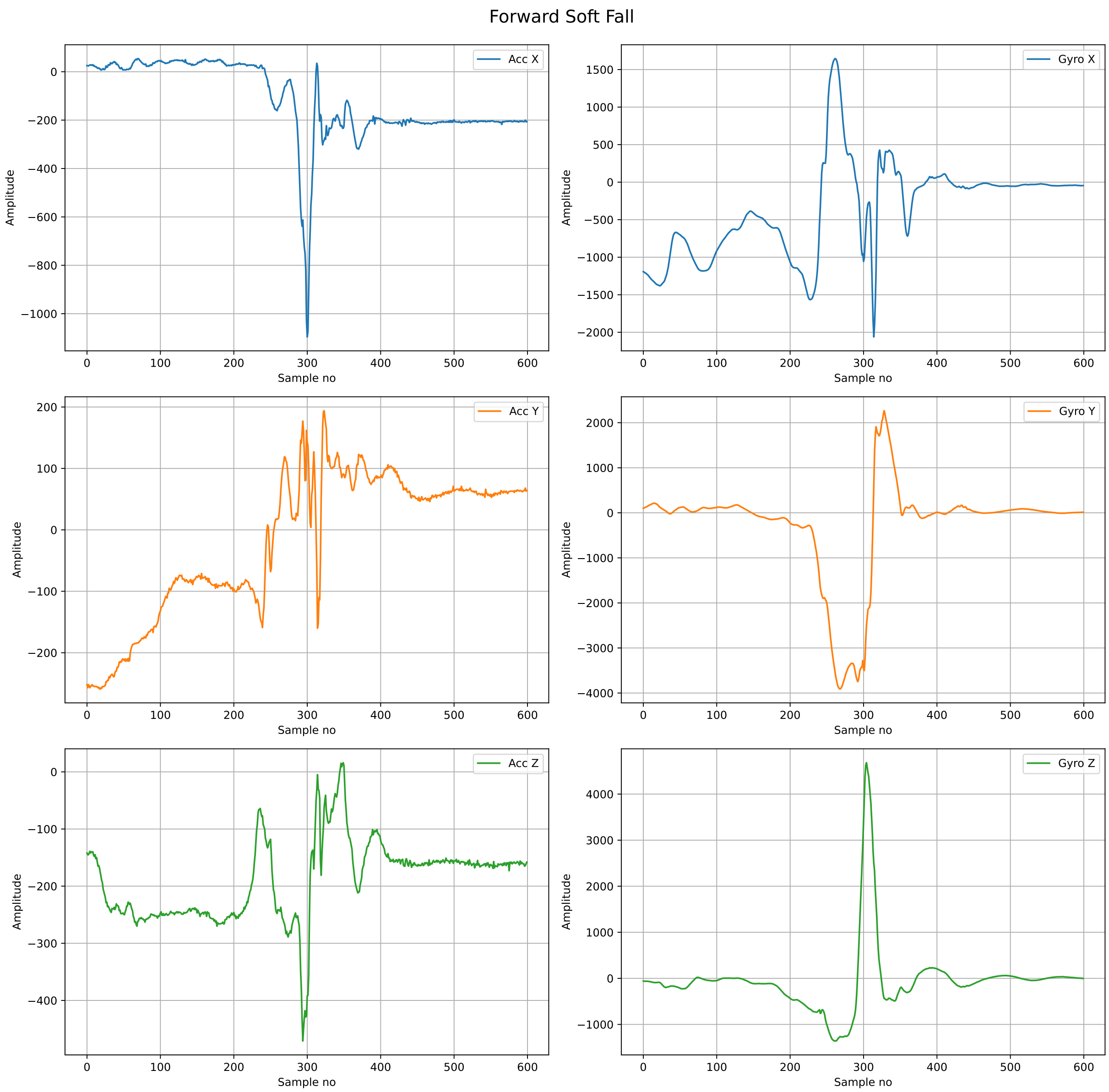

Figure 2. Accelerometer and Gyroscope measurements: Forward Soft Fall

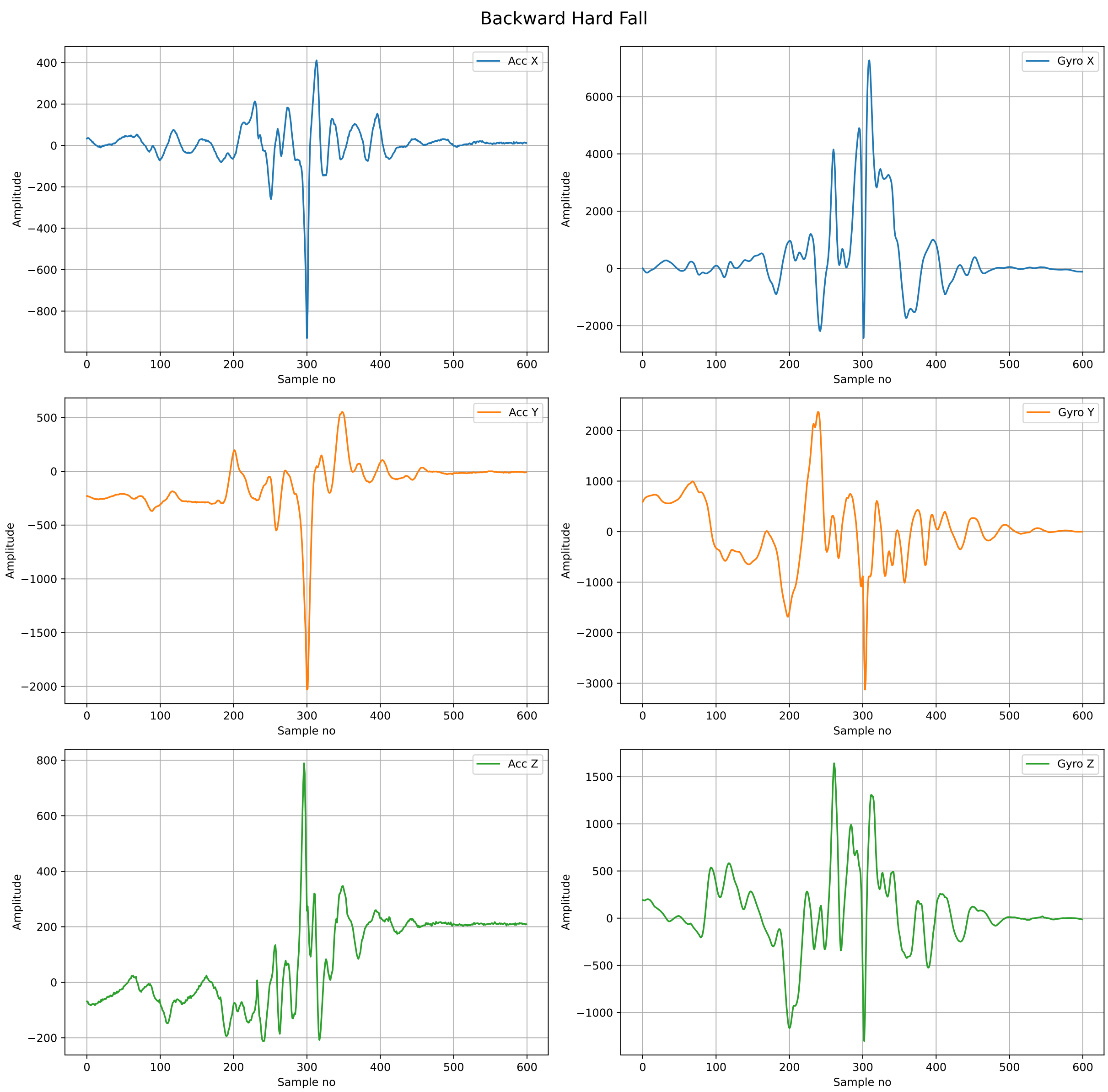

Figure 3. Accelerometer and Gyroscope measurements: Backward Hard Fall

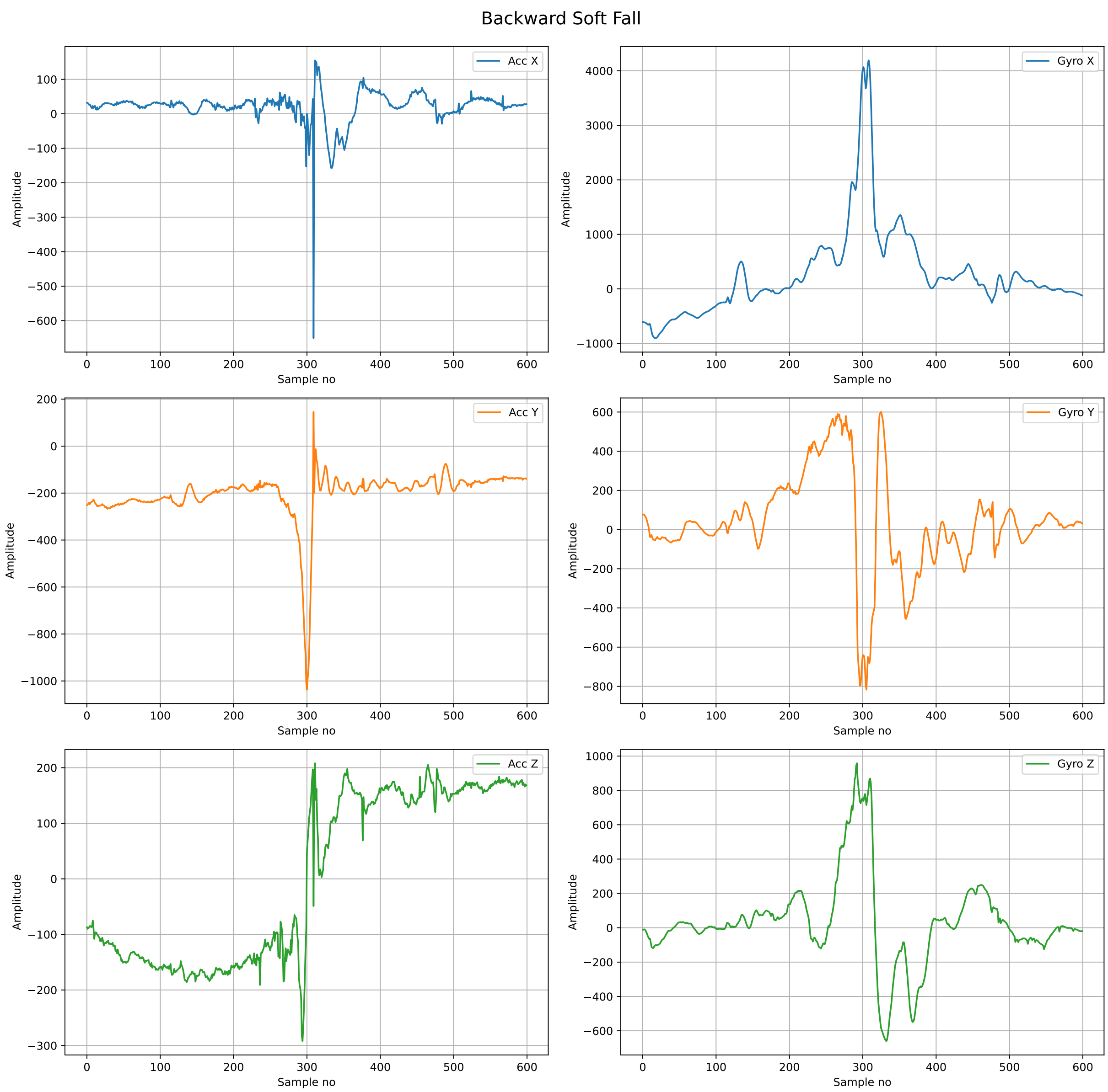

Figure 4. Accelerometer and Gyroscope measurements:f Backward Soft Fall

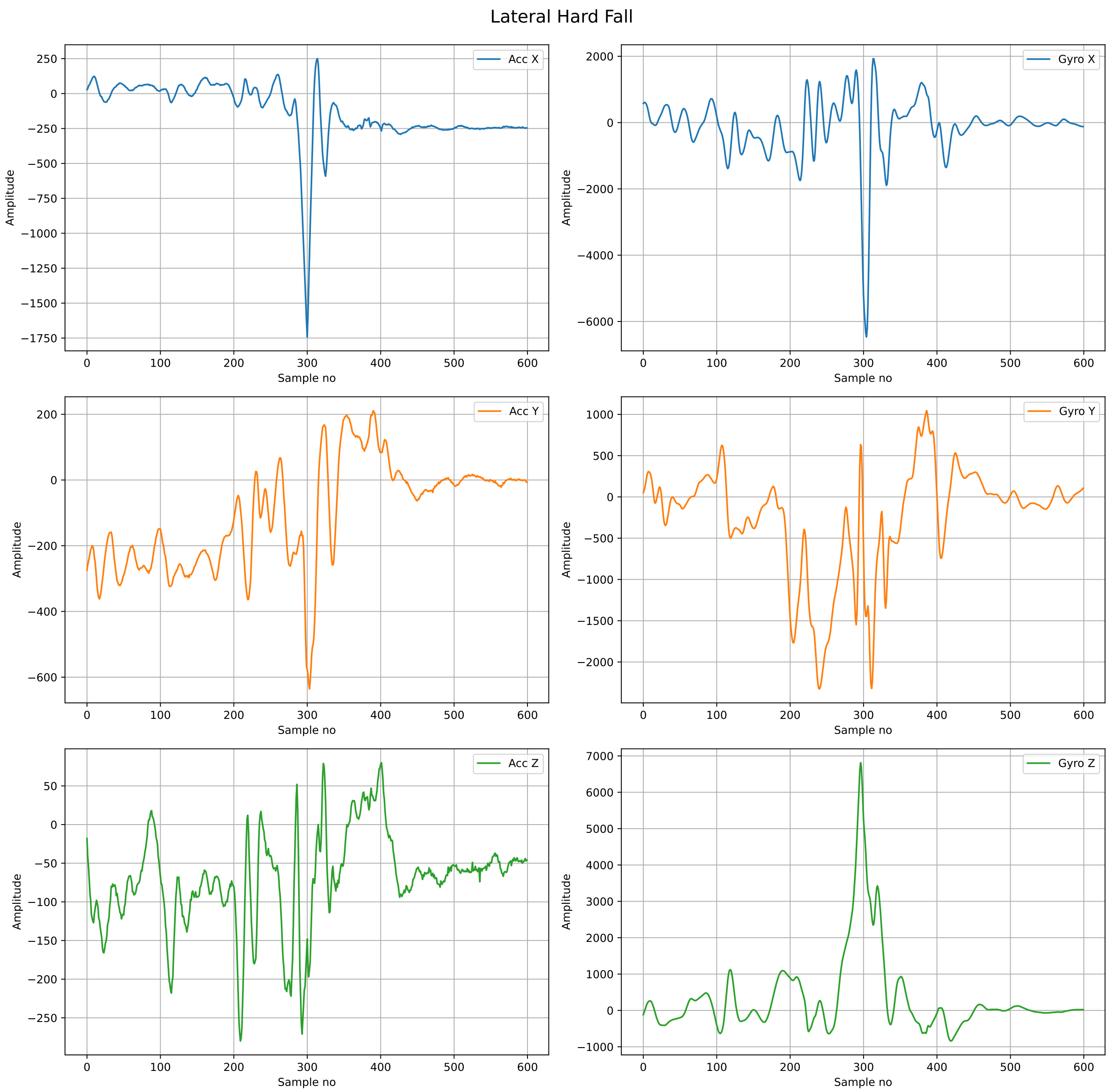

Figure 5. Accelerometer and Gyroscope measurements: Lateral Hard Fall

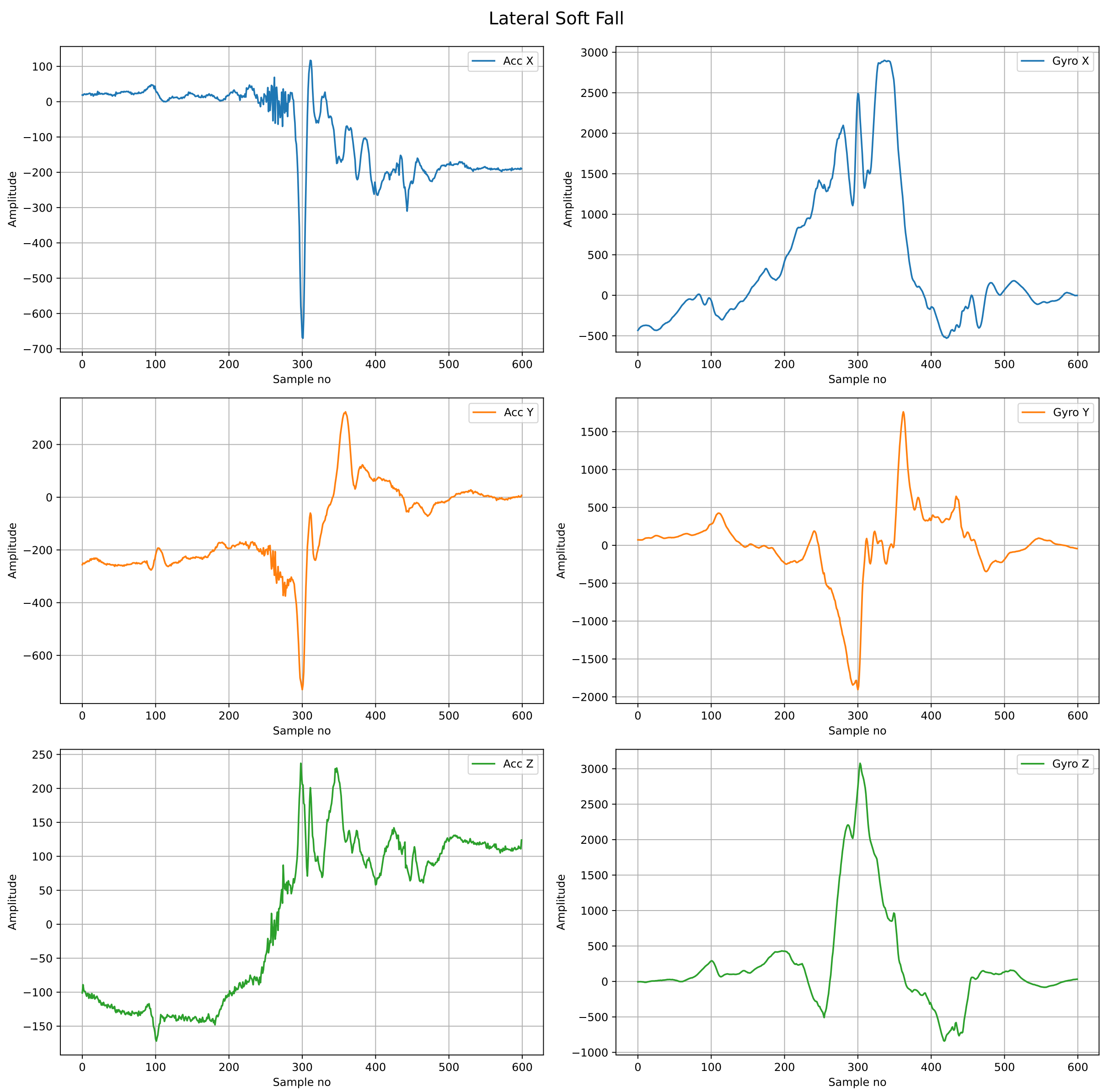

Figure 6. Acclerometer and Gyroscope measurements: Lateral Soft Fall

Appendix B: Acronyms

ABC - Artificial Bee Colony
ACO - Ant Colony Optimization
AdaBoost - Adaptive Boosting
ADL - Activity of Daily Living
AI - Artificial intelligence
ANN - Artificial Neural Networks
Bi-LSTM - Bi-Directional Long Short-Term Memory
CERT - Computer Emergency Response Team
CNN - Convolutional Neural Networks
CO - Carbon Monoxide
CO2 - Carbon Dioxide
DARPA - Defense Advanced Research Projects Agency
DDOS - Distributed Denial of Service
DE - Differential Evolution
DER - Distributed energy resource
DOS - Denial of Service
DL – Deep Learning
DLEFN - Deep Learning Entrusted to Fog Nodes
DNN - Deep Neural Networks
DT - Decision Tree
EBT - Ensemble Bagged Tree
EC - Ensemble Classifier
ECG - ElectroCardioGram
EEG - ElectroEncepheloGram
EMG – ElectroMyoGram
ETSI - European Telecommunications Standards Institute
FANs - Field Area Networks
FDS - Fall Detection System
FN - False Negative
GA - Genetic Algorithm
GRU - Gated Recurrent Units
GSM - Global System for Mobile Communications
HANs - Home Area Networks
ICT - information and communication technologies
IEEE - Institute of Electrical and Electronic Engineers
IETF - Internet Engineering Task Force
IQ Range - Interquartile Range
IMU – Inertial Measurement Unit

IoT - Internet of Things
ISM - Industrial, Scientific and Medical
K-NN - K-Nearest Neighbor
LDA - Linear Discriminant Analysis
Li-Fi - Light Fidelity
LoRaWAN - Long Range Wide Area Network
LPWAN - Low Power Wide Area Network
LR - Logistic Regression
LSDVRP - Large-scale Dynamic Vehicle Routing Problem
LSTM - Long Short-Term Machine
LTE - Long-Term Evolution
MITM - Man in the Middle
ML - Machine Learning
MVRP - Multidepot Vehicle Routing Problem
NANs - Neighborhood Area Networks
NB - Naive Bayes
NB-IoT - Narrow Band IoT
NFC - Near Field Communication
OSP - Optimal Sensor Placement
PCA - Principal Component Analysis
PIR - Passive InfraRed
PMU - Phase Measurement Units
PSD - Power Spectral Density
PSE - Power Spectral Entropy
PSO - Particle Swarm Optimization
PPCA - Probabilistic Principal Component Analysis
PV - photo voltaic
P2I - Pedestrian to Infrastructure
QSVM - Quadratic Support Vector Machine
RBF - Radial Basis Function
RF – Radio Frequency
RFC - Random Forest Classifier
RFID - Radio Frequency Identification
RF-RFE - Random Forest Recursive Feature Elimination
RMS - Root Mean Squared
RNN - Recurrent Neural Networks
SAE - Stacked Autoencoder Networks
SAX - Symbolic Aggregate Approximation
SHM - Structural Health Monitoring
SMV - Signal Magnitude Vector
SWOT - Strength Weaknesses Opportunities Threat
SVM - Support Vector Machine
TLS - Transport Layer Security
TP - True Positive
VANET - Vehicular Adhoc NETworks

VM - Voting machine
VRP - Vehicle Routing Problem
VRPPDTW - Vehicle Routing Problem Pick-up and Delivery with Time Windows
V2V - Vehicle to Vehicle
V2I - Vehicle to Infrastructure
V2P - Vehicle to Pedestrian
WANs - Wide Area Networks
Wi-Fi - Wireless Fidelity
Wi-SUN - Wireless Smart Utility Network
WPA2 - Wi-Fi Protected Access 2
WSNs - wireless sensor networks
XGBoost - Extreme Gradient Boost
Z-Wave - Zensys Wave

CURRICULUM VITA

NAME:	Muhammmad Zaigham Abbas Shah Syed
ADDRESS:	Computer Science and Engineering Department J.B Speed School of Engineering University of Louisville Louisville, KY 40292 United States of America.
EDUCATION:	Ph.D., Computer Science & Engineering, December 2022 **University of Louisville**, *Louisville, KY, USA.* M.Sc. in Electronic and Electrical Engineering, November 2013 **University of Strathclyde**, *Glasgow, United Kingdom* B.Eng in Electronics Engineering, March 2010 **Mehran University of Engineering and Technology**, *Jamshoro, Pakistan*
WORK EXPERIENCE:	**Graduate Assistant**, *Innovative Technologies Lab, University of Louisville*, KY, USA, 2022

PUBLICATIONS:

JOURNAL

1. **Abbas Shah Syed**, Daniel Sierra-Sosa, Anup Kumar, and Adel Elmaghraby. IoT in smart cities: a survey of technologies, practices and challenges. Smart Cities, vol. 4, no. 2, pp. 429–475, 2021

2. **Abbas Shah Syed**, Daniel Sierra-Sosa, Anup Kumar, and Adel Elmaghraby. Making Cities Smarter—Optimization Problems for the IoT Enabled Smart City Development: A Mapping of Applications, Objectives, Constraints.Sensors, vol. 22, no. 12, p. 4380, 2022

3. **Abbas Shah Syed**, Daniel Sierra-Sosa, Anup Kumar, and Adel Elmaghraby. A hierarchical approach to activity recognition and fall detection using wavelets and adaptive pooling.Sensors, vol. 21, no. 19, p. 6653, 2021

4. **Abbas Shah Syed**, Daniel Sierra-Sosa, Anup Kumar, and Adel Elmaghraby. A deep convolutional neural network-XGB for direction and severity aware fall detection and activity recognition.Sensors, vol. 22, no. 7, p. 2547, 2022

CONFERENCE

1. **Abbas Shah Syed**, Daniel Sierra-Sosa, Anup Kumar, and Adel Elmaghraby. Cross dataset non-binary fall detection using a ConvLSTM-attention network. **(accepted at IEEE GlobeCOM 2022).**

2. **Abbas Shah Syed**, Daniel Sierra-Sosa, Anup Kumar, and Adel Elmaghraby. Determining Fall direction and severity using SVMs.2020 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT),2020, pp. 1–7.