T.C.

ALTINBAS UNIVERSITY

Institute of Graduate Studies

Information Technologies

IOT BASED SMART GREENHOUSE MONITORING AND CONTROL SYSTEM WITH EARLY FLOOD DETECTION

Sarmad MOHAMMED

Master of Science

Supervisor

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Istanbul, 2020

IOT BASED SMART GREENHOUSE MONITORING AND CONTROL SYSTEM WITH EARLY FLOOD DETECTION

by

Sarmad Abdullah Mohammed Mohammed

Information Technologies

Submitted to the Institute of Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2020

The thesis titled "IOT BASED SMART GREENHOUSE MONITORING AND CONTROL SYSTEM WITH EARLY FLOOD DETECTION" prepared and presented by Sarmad MOHAMMED was accepted as a Master of Science Thesis in Information Technologies.

____

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Supervisor

Thesis Defense Jury Members:

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	School of Engineering and Architecture, Altinbas University	___________________
Asst. Prof. Dr. Mesut CEVIK	School of Engineering and Architecture, Altinbas University	___________________
Asst. Prof. Dr. Tariq ABED	Faculty of Computer Science and Information Technology, Karkuk University	___________________

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

Approval Date of Institute of Graduate Studies: ____/____/____

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

Saramd Abdullah Mohammed Mohammed

DEDICATION

I devote and pledge this research work to my supervisor who is salient for guiding me through whole research work as well as my family for always assisting me in my hard time.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisor Asst. Prof. Dr. Abdullahi Abdu IBRAHIM for guiding and helping me along the way in writing this dissertation. Discussing my progress, problems, and ideas with my supervisor Asst. Prof. Dr. Abdullahi Abdu IBRAHIM a couple of times every week helped me tremendously in understanding the logic behind the research. It made me better realize the technical need for this research work.

ABSTRACT

IOT BASED SMART GREENHOUSE MONITORING AND CONTROL SYSTEM WITH EARLY FLOOD DETECTION

Mohammed, Sarmad,

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Date: Jul /2021

Pages: 58

The aim of this thesis is to develop an internet of things based smart greenhouse with flood detection monitoring system that can be deployed using the existing WiFi networks, avoiding hard-installation processes and allowing users to add or remove devices from the system according to their needs using the simulated environment for automation as well as the greenhouse monitoring. The research took advantage of Internet-of-Things (IoT) based Message Queuing Telemetry Transport (MQTT) protocol to guarantee detection and interoperability in our system, such as the QoS levels that we use to specify the flood data, where according to the defined levels, the flood data could be processed with early forecast delivery assurance achieving an accuracy of 97.36%. Thus, the objective of this work is to present a solution for a smart greenhouse system that only uses the WiFi network for communication with MQTT protocol, avoiding intrusive installations (it does not need to add physical wiring, just effective python programming based advance simulation), and allows to connect and control different devices, from any vendor, remotely from a mobile phone, a tablet or any other device with an internet connection where the SEN-12 dataset has been used for flood data providing different geographical regions for potential flood. A complex and modular system has been developed, which uses standard technologies and protocols and consists in a central server (smart greenhouse server), which offers a set of services to control and manage the system; an auxiliary database to minimize information that devices need to store; a MQTT broker that allows the devices to communicate using this protocol, the MQTT, and finally the devices whose requirements are minimum, they only must able to connect to an WiFi network. The system does not limit the quantity of connected devices since its architecture was designed and implemented to allow horizontal scalability and high availability to handle with a growing amount of devices using the flood detection with low tariff of unit consumption.

Keywords: Internet of Things, MQTT, Greenhouse, Monitoring, Wifi Network .

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

LIST OF ABBREVIATIONS

IoT	:	Internet of Things
RF	:	Radio Frequency
MAC	:	Medium Access Control
QoS	:	Quality of Service
TCP	:	Transmission Control Protocol
REST	:	REpresentational State Transfer
HTTP	:	Hyper Text Transport Protocol
M2M	:	Machine-to-Machine
MQTT	:	MQ Telemetry Transport
LTE	:	Light Term Evolution
DSL	:	Digital Subscriber Line
R2L	:	Remote to Local

1. INTRODUCTION

1.1 INTRODUCTION

The Internet of Things can be defined as the network of physical devices embedded with electronics, software, sensors, actuators and connectivity which enables them to exchange data with each other and the Internet using their unique identifiers as mentioned in [1]. Such an infrastructure allows the development of applications and services that facilitate the lives of citizens and workers around the world and bring us closer to the convergence of the physical and digital worlds. These applications have provided us with solutions in fields like smart-grids, smart-greenhouses, intelligent transportation or smart-cities. Some successful examples include, amongst others, Nest and Ecobee for smart-heating, Phillips Hue for smart-lighting, Smart things as a provider for multiple smart-greenhouse solutions, researchers in [2] worked for personal fitness trackers, as well as the personal assistants like Alexa from Amazon or Siri from Apple.

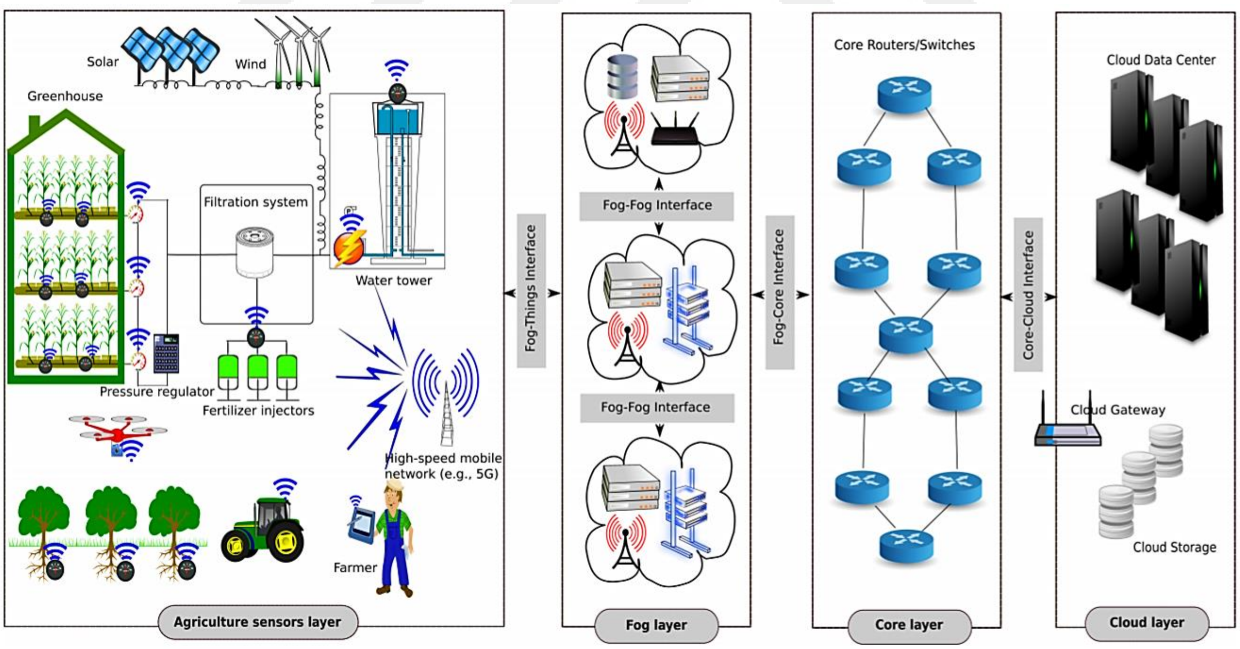


Figure 1.1: A smart and connected four-tier green IoT-based greenhouse architecture [2].

Despite the huge growth of such IoT applications, the usefulness of the data that originate from these systems is still to be validated and proved. The volume of data that can be generated from a single sensor device installed in a house or a wearable device worn by a person can be overwhelming for the device itself and in most cases needs to be offloaded to a data-processing application from which users can access it and connect it to a more meaningful use [3]. Taking into account the rate with which such devices are being integrated into our everyday lives, the result of offloading so much information in the cloud can create huge volumes of data. Such data need to be communicated, processed and stored in real time, while accessing it in its raw format tends to be useless when compared with the knowledge it can generate (e.g., information about a person's lifestyle and diet compared with their daily step count and trips as latitude-longitude coordinates) as mentioned in [4]. As a result, the need for a holistic solution on the subject of managing the data generated is still an open subject for research in both academic and enterprise scopes. Similarly, multiple interfaces for communicating and representing such data formats have been used, with none getting a clear step ahead of the others in the path for a global standard, if and when this will be possible.

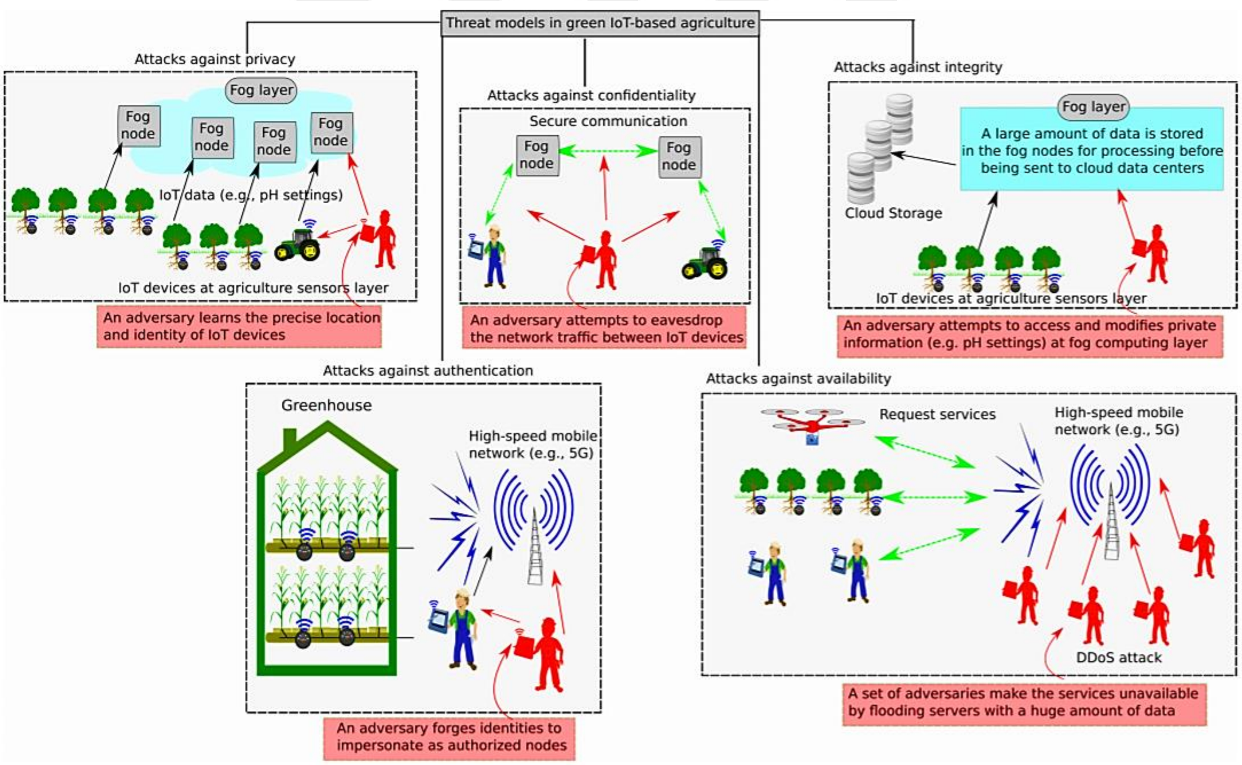


Figure 1.2: Data Streams Of The Threat Models In Green Iot-Based Control System [2].

Another important characteristic of the IoT domain is the huge diversity of the communication mediums, protocols and data representation and exchange formats used. Figure 1.2 showcases how IoT devices communicate the collected data with cloud services using multiple technologies and protocols for local-only or local-to-cloud communications as mentioned in [5]. Each protocol offers different capabilities in terms of throughput, communication range or power consumption, based on the needs for the application use case. For example, communication protocols with limited range like Zigbee[6], Bluetooth LE[7] or WiFi [7] are used to communicate data between wearables or smart-greenhouse devices and local gateways. Zigbee and Bluetooth LE are better suited for devices that operate on batteries and exchange limited amounts of data (e.g., wearable fitness trackers or beacons). On the other hand, WiFi is more suited for devices that need more real time communication and can exchange larger chunks of information like smart-plugs or cameras. Such devices are mostly connected to wall outlets as WiFi needs significantly more power and using batteries is not feasible on a long-term basis. On the other end, to communicate information with the Internet, gateways or smart-devices use technologies like DSL [8] lines, LTE [9] or LoRa [10] and Sigfox [10]. DSL and LTE offer a higher throughput when compared to LoRa and Sigfox but have a higher installation and operational cost as they require on premises infrastructures, maintenance and recurring costs either as monthly or pay-as-you-go data plans. LoRa and Sigfox are simpler solutions, with no installations on the premises of the user, other than the device itself, and minimal (i.e., a yearly subscription for Sigfox) or no costs (LoRa) but offer significantly reduced throughput. In all the above cases, the characteristics of the communication module used defines the sampling rate that can be achieved by the sensing modules of the device. A general rule followed in most applications or devices currently available is that devices try to sample and transmit data as fast as possible while maintaining a decent amount of battery life. This is done to ensure that the phenomena monitored are well observed and no desired events are missed due to devices sleeping to conserve power as mentioned in [11]. This behavior leads to an ever increasing load of data that needs to be processed in order to extract useful information and conclusions.

This fragmentation in the IoT domain presents the main challenge of integration and interoperability to achieve the desired end result of a unified virtual environment. For instance, a data analytics solution needs to ingest data streams from multiple of the sources presented above. Mixing different hardware and technologies makes this much more complex in order to finally make every device or system “talk to” and “understand” each other. Finally, there are all the cyber-security and cyber-privacy issues for not only wireless communication, but also for the control and operation of the connected assets as mentioned in [12].

From the user's or customer's perspective, it is a costly and time-consuming experience to become familiar and maintain multiple solutions from different vendors, as well as integrating them all together in a single system. Industry standards are being developed (i.e. as wireless protocols), and continue to evolve through open- source frameworks that encourage collaborations among solution developers but global acceptance is hard to achieve due to contradicting interests from the involved partners.

1.2 MOTIVATION

To better understand the collection mechanisms of data from more distributed and uncontrolled IoT installations we also developed a solution that can collect data using smartphones in smart cites carried by volunteers in the context of crowd-sourcing campaigns. This method is used to augment smart-city installations with mobile and inexpensive infrastructures using the power of volunteer's local activist groups. Such solutions can be deployed with minimal cost by the officials of a city as a new service that can provide them with important insights for the local conditions, environmental, social or other. This smartphone application was used in more than 10 experimentation campaigns in cities around Europe by more than 50 users over the past 2 years as mentioned in [13].

Finally, we also present a framework for characterizing the data from all the above inputs using Machine Learning in real time by taking advantage of the streaming data mechanism developed in the data processing engine described above. This framework was used to analyze real world data from smart-city IoT infrastructures as well as the data that originate from smaller scale IoT installations. The data generated in the context of the IoT do not follow the typical characteristics for data processing technologies and methodologies used in past computer science use cases as their volume usually exceeds the size a single machine can process in real time. The first attempts to harness such data were made using big data technologies like Hadoop [14]. Such systems use a set of processes that can be run in rounds one after the other and reduce the actual volume of data available to a small and usable information, and were capable of overcoming the original problems encountered in IoT systems. However, they lack a viable solution in the most recent use cases of IoT, where the actual data are not a priori available to the system that will process them. Such data are called streaming and are becoming a more and more common example of how the information from IoT installations will arrive to the processing system.

For managing large data streams produced by the Internet of Things for smart greenhouses, several research prototype implementations and some more commercially-oriented solutions have been presented. There is a large set of requirements, like publishing/subscribing on data streams, interfacing with various technologies and performing real-time analysis that makes it crucial to build systems extending a lot of classical RDBMS systems. One of the fist options made available in this area was flood detection monitoring system [15] It is a secure, scalable platform that connects devices and products with applications to provide real-time control and data storage. Other enterprises, that focused more on offering computing services, like Amazon and Microsoft also offer solutions for IoT data processing with AWS IoT [16] and Azure IoT Suite [16]. Both systems focus on helping device manufacturers collect and send data to the cloud, while making it easy to load and analyze them.

At the core of all similar solutions lies the concept of message exchanges using normally a central service that plays the role of a flood detection generation. A flood detection generation, in the general case, maintains a list of available topics on which client applications can publish updates while other application can subscribe for updates. This design concept allows the development of asynchronous system designs that can be developed independently without any restriction on the hardware or software used.

1.3 PROBLEM STATEMENT

In order to provide such a holistic solution, it is important to be able to integrate the data that are generated by all of these smart greenhouses in a common base. We need to be able to process them, view them and operate on them using a common methodology and also be able to gain common knowledge from them with flood detection monitoring. This is the most important step towards the unification of the information originating from a fragmented ecosystem that can lead to the development of next-level applications and services with significantly reduced development effort. In this direction and in the context of this dissertation, we study the possible solutions to the aforementioned problems in an environment filled with IoT devices. We focus our work in the following research problems:

a) The representation of the IoT environment together with its metadata and semantics in a suitable representation format capable of handling the complex relationships that are generated in such a densely populated environment for monitoring and controlling the early flood detection system.

b) The efficient collection and processing of streaming IoT data for smart greenhouses in real time with the minimum processing time and latency, using a modular system that can be extended to support additional data types and devices with limited interventions for early flood detection.

c) The extraction of knowledge from the collected and processed greenhouse smart IoT data. Raw data are of limited value to end users in most cases. The real value lies on the conclusions that can be extracted from them.

Once we have achieved all objectives, we will be able to analyze IoT data in real time for smart greenhouse with flood detection generation offering the outcomes of our work to researchers that can then build upon our work and generate a truly intelligent and connected environment.

1.4 RESEARCH CONTRIBUTIONS

Based on the problem set in Section1.3 in this dissertation we contribute to the field of data analytics for the Internet of Things in smart greenhouse environments and the post processing analysis of the IoT data to provide a better understanding of the physical surroundings with the help of flood system monitoring. We propose new and extensible methods to store the information of IoT installations and our findings allow for fast look-ups across the whole set of IoT devices in the network. Processing the data generated in real-time becomes a more streamlined work flow, using a standardized methodology that allows for changing the data analysis from a single point. Data generated can be easily stored and retrieved from a specifically designed, reliable and efficient storage engine. Using the original and calculated data in a feedback loop also allows us to build additional processing and analysis layers that generate more data and information resulting in more accurate outcomes.

The results of our work can be categorized in the following points:

a) The first goal of our work was to setup a base set of infrastructures for bringing the physical world with its digital representations. We wanted to be able to understand what is where and how it interacts with its environment as well as what it observes in real time for flood detection. Our work towards this direction was twofold:

b) We have developed a graph-based schema to represent IoT installations together with their semantics and meta information. The schema is not based on the traditional relational databases but builds on the ideas of graph theory and utilizes a new database model, a graph database. Each entity that participates in real world interactions or can be observed by an IoT device is represented as a nodes of a graph, while the interactions themselves are the vertices of the graph.

c) We therefore build a web of entities and relations that are easy to visualize and traverse to find answers to various queries that may arise inside a smart environment. Such queries can look for the causes of events observed (e.g., what caused the rise of luminosity in room), the available information for an area (e.g., what information is sensed for the building) or even the social interactions and connections of individuals (e.g., which people use the same appliances).

d) For the analysis of the data we provide a template implementation for setting up a system that can receive, process and analyze an unbounded number of input data streams with no impact on its operation. Such data streams can originate from sources that range from single smartphones to city wide sensor installations.

e) The system is also capable of handling data streams that provide unbalanced volumes of information with no performance drops. The calculation methodology itself is developed as to be easily customized and extended, based on the data types and the calculations required in every environment it is deployed to. This processing engine is capable of identifying the data from each of the sensing devices routing them to the appropriate processor without any prior per-device configuration.

f) Both the database and processing engine are used to support a publicly available system that supports over 300 users and consumes data from more than 20 IoT installations with almost 5000 sensing points. Based on our evaluation the database model can easily respond to all queries in real time, even in use cases where the database ranges to tens or even hundreds of thousands of entities.

g) We also performed an extensive evaluation of the processing engine in real world conditions for more than 3 months achieving sub-millisecond processing times per measurement. We stressed our system with data volumes that exceed by far the data collection rates of our real world installation to prove that it is future-proof and scalable.

1.5 THESIS ORGANIZATION

In the first chapter, background and motivation, research question, methodology and outline are presented. The smart greenhouse based on IOT network and IOT network challenges are presented with a specific emphasis in introduction of dissertation with dependencies of problem statement and research contribution. The chapter ends with a description of the characteristics of an organization of thesis.

The second chapter gives a basic description of the literature review on smart greenhouses and internet of things, beginning with a small historical note, a definition of the commercially available types of available solutions and ending with a description of the major drawbacks of not classifying the normal IoT network for smart greenhouses and the definition of flood detection system for real time monitoring in large scale environment.

The third chapter presents the smart greenhouse IoT system with the methodology background; the concept of IoT techniques with networking concepts is presented, followed by the methodology for intelligence smart greenhouse with flood detection monitoring techniques, a characterization of the internet of things based network, and the method for classification and reducing cost of flood detection monitoring system with internet of things by providing the cost analysis, the purpose of this chapter, go through a sensitivity analysis and are presented and discussed.

The forth chapter in terms of structure is very similar to the third chapter but for the result of dissertation with aspect of this work, that is the IoT based smart greenhouse with flood detection monitoring system using time related features on internet of things based system. A typical framework outline is followed: first the concept of IoT based smart greenhouse results is described, then the objective and scope of the analysis are being found, followed by data collection for smart greenhouse using the internet of things.

The fifth chapter provides the discussion of the results with an analysis and their respective discussion based on the limitations and comparison with other existing systems on smart greenhouse with internet of things with flood detection monitoring system.

The sixth chapter presents the main conclusions of this dissertation with a summary of the main findings, the consequences these results have in policy making, followed by the main limitations of this study and ends with leads for future work.

2. RELATED WORK

This thesis work is motivated by the goal of overcoming the challenges and difficulties that smart greenhouse monitoring system faces with the help IoT techniques in real time with their structure and the environmental hostility surrounds them, that is by creating a better environment solution, a solution that will avoid the shortcomings of the traditional smart greenhouse controlling operations. For the sake of explaining the motivation of this thesis work, a detailed review of all these difficulties and constraints is discussed in depth in the sections below; furthermore, a comprehensive review of flood monitoring system and their advantages that this thesis work aims to overcome.

2.1 LITERATURE REVIEW

As we are now rapidly approaching 2030, it is as important as ever to address the skills that will enable all citizens to make informed and well-thought choices as mentioned in [17]. Also, another ubiquitous fact should also come to mind: we cannot manage what we cannot measure. It is necessary to monitor the impact of our current behavior or the effect of potential behavioral changes in order to have a clearer picture with respect to e.g., our everyday energy consumption. Furthermore, environmental education, as part of the broader issue of science, is an important component of the EU cultural heritage [18]. In fact, EU considers environmental education one of the most prominent instrument to influence human behavior towards more environmentally sustainable patterns [19]. Hence, associating environmental education and game-based learning will lead to students taking over a leading role in the educational process, setting questions, investigating the possible answers and looking for alternative explanations to come up with a fitting model.

Several software products for monitoring sensor data exist, able to capture sensor data from proprietary data formats or protocols and visualize them. Regarding visualization, [20] discusses the most common approaches with respect to feedback design in eco-conscious work for the past decade, from the perspective of both ICT and psychology, providing insights to their strengths and weaknesses. [21] is an example of a typical engineering-focused approach, that utilizes a number of skeuomorphic metaphors to enable smart greenhouse feedback creation on smartphones. While such systems provide end-users and developers with powerful tools and front-ends, we believe they should also be paired with multiple approaches, offering multiple possibilities to interact with the system. [22] are examples of large-scale smart metering deployments that used IoT portals (for electricity and water) and other visualization tools to support the system and engage end-users to participate. Their findings support the notion that the use of multiple approaches, with respect to visualization and feedback, serves such purposes well. We have followed a similar line of thought while implementing our own user interfaces and will continue to evolve our approach in future revisions of the system.

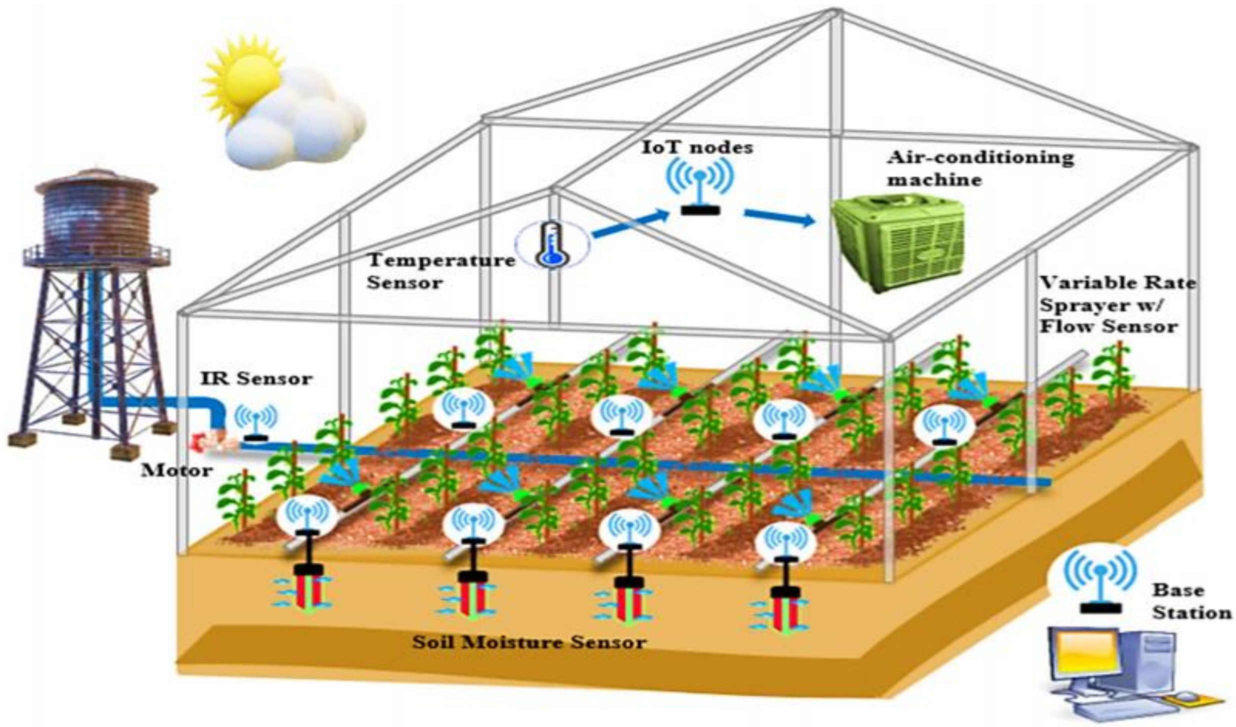


Figure 2.1: An Advance Smart Greenhouse Ecosystem Schema Developed Using IoT-technology [22].

In order to interact with an IoT installation, an appropriate representation for it in the digital world is required. Each installation comprises multiple elements that produce, consume or transfer data. Every element in this ecosystem potentially has multiple relations with other ones creating a complex schema that can be hard to understand or visualize as the scale of the installation increases as mentioned in [23]. To overcome this problem and simplify our interaction with this digital image representation, formats more appropriate than the traditional relational databases need to be applied. At the same time, these formats need to be easy to understand and be used by non tech-savvy users like artists or activists in their work with this, now accessible, living part of our environment.

Researcher in [24], focus on a simple case of a building populated with IoT devices for monitoring and limited actuation use cases to extract information about all the involved entities and their possible interactions. This information will help us define a representation schema that best fits our requirements, while it is suitable for use in a high performance computation system.

2.2 BENEFITS OF INTERNET-OF-THINGS

In 2019 the number of Internet of Things (IoT) connected devices was 19.71 billion worldwide. By the end of 2020 it will hit 26.66 billion as mentioned in [25]. In 2025, the number of devices connected to the Internet will be 75.44 billion, causing the number to rise 5 times in a decade as mentioned in [26]. These smart devices connected to the Internet are capable of generating data on its own based on behavior or expected result and share it over the Internet. This is what constitutes the concept of Internet of Things (IoT). IoT is a massive group of devices containing sensors or actuators connected together over wired or wireless networks as mentioned in [27]

Besides all the benefits and positive characteristics that the wireless medium has brought to IoT, there are vulnerabilities. There are different forms of traffic anomalies and attacks. The most common security threats are intruders, which is generally referred, as hacker or cracker and the other is virus. This comes with the risk of personal data being broadcasted to the world and cause more cyber-attacks like phishing, denial-of-service (DoS), Probe, Remote to Local (R2L) attack etc. Since these attacks are not common so normal data flow has a pattern which can be more or less the same [28]. However, when another entity will try to distort or change the data, the pattern will change and this is where the Intrusion detection techniques comes into play in the context of internet and specifically in the context of Internet of Things.

To ensure the system and data are only accessible to authorized entities such as individuals and processes, logical controls are used as software shields. Intrusion detection and prevention system, passwords, access control etc. include logical controls as mentioned in [29].

To monitor and detect anomalies or any suspicious behavior, intrusion detection system (IDS) and intrusion prevention system (IPS) are being used. As diverse environments and latest technologies are prone to be maliciously attacked, machine-learning (ML) algorithms are able to detect, analyze and classify intruders in network accurately and rapidly as mentioned in [30].

Two types of intrusion detection system are there. They are: Anomaly based detection and Signature based detection. Anomaly based detection methods detects the attacks that are unknown by observing the whole system and the entities in it such as traffic, objects etc. Anything other than normal behavior is identified as a potential attack. Conversely to detect specific patterns of the attacks that are known by investigating network data or traffic, signature based detection methods are powerful as mentioned in [31].

After researching on specific use cases, data characteristics and system needs, such a diversity of characteristics has confirmed that there is no unique method that would provide efficient and accurate intrusion detection for every dataset under the analysis as mentioned in [32].

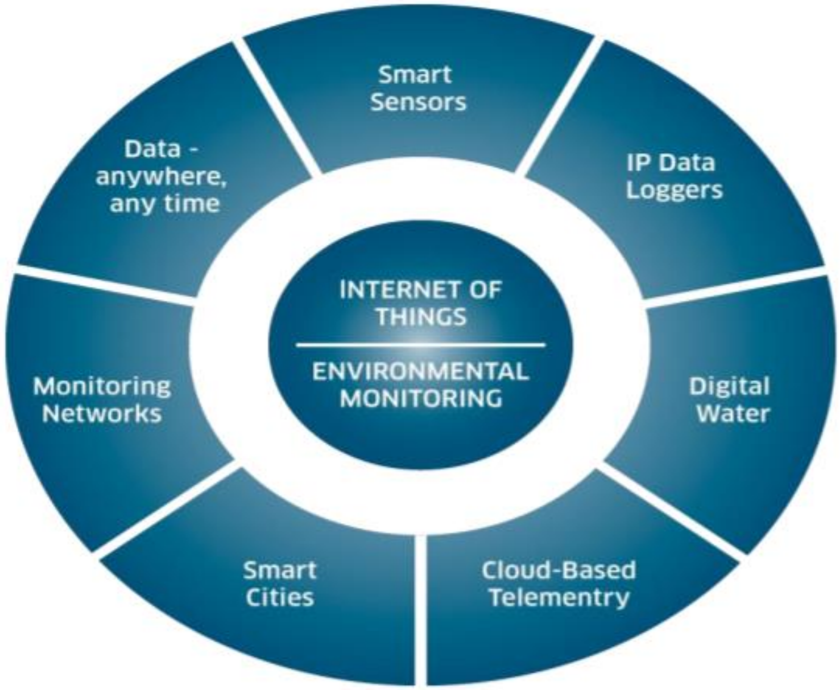


Figure 2.2: A Topography Of Different Intelligent Iot Based Choice For Flood Detection With Environment Monitoring [32].

2.3 IOT SUPPORT IN INTELLIGENT GREENHOUSE AUTOMATION

2.3.1 Advantages of Internet of Things (IoT)

IoT provides several day-to-day advantages not only for consumer's sector but for business sector as well as mentioned in [33]. This technology allows us to view real time information that was not available before. Business can improve their production efficiency by reducing material waste and unforeseen downtime. Since Artificial Intelligence is used, it saves considerable human effort and time. Sensors can be used to build infrastructure, detect damage to infrastructure. Automated traffic control can help reduce road congestion and gadgets or sensors can be implemented on the outside to recognize change in the environment and warn us for any impending natural calamity.

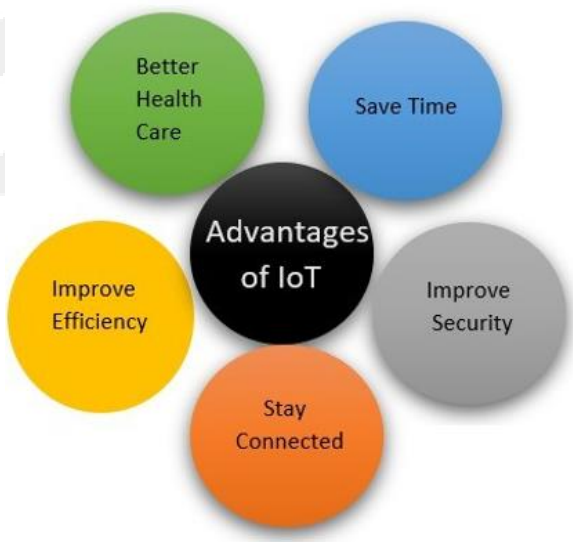


Figure 2.3: Advantages of IoT Described. Source [31]

IoT makes our greenhouses, offices and vehicles smarter, more measurable and this in turn makes our life more pleasing. Smart speakers like Amazon's echo or Google greenhouse makes it very easy to set timers, get information on the go or be it to play music. Through greenhouse security system, we can monitor our greenhouse from other places remotely and can see what is happening on the inside and outside or talk to visitors. Smart lighting systems can reduce unnecessary energy consumption as it stays off if there is no one in the house or smart air conditioners will turn on when it knows you are on your way to your greenhouse.

2.3.2 Disadvantages of Internet of Things (IoT)

Security in IoT is a double edged sword. Now that we have a system of interconnected devices on a network it makes the system somewhat secure and efficient as mentioned in [34]. Though connected devices generate colossal amount of data and are constantly being exchanged between devices so it in turn generates lots of internet traffic. These data can make the devices useful but this easily accessible nature of the internet brings up both security and privacy issues. These issues arise for many reasons.

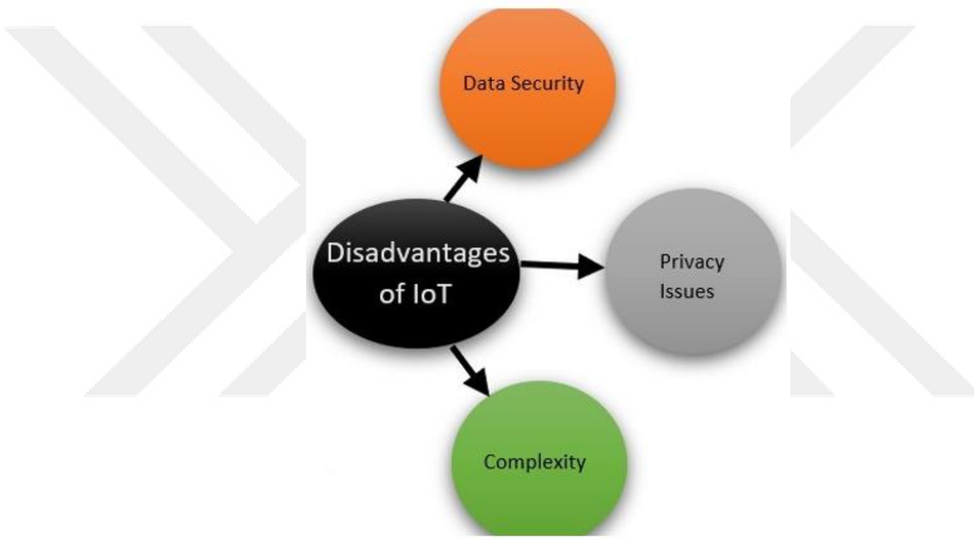


Figure 2.4: Disadvantages of IoT Described. Source [31]

Flaws in software is found very frequently because of the fact that IoT devices lack the capability to be patched or fixed. This makes them prone to security risks like DDoS attacks, one of the most common issues, which happens by setting the device's password to its default password which are easily crack able by hackers as mentioned in [35]. Ransomware and Malware both relies on encryption to lock out the user's devices completely and steals the user's data. Privacy remains among the largest issues of IoT as data is being transmitted, stored and processed and being harassed by large companies as mentioned in [36]. This very data is then being sold to various other companies which is violating our rights for privacy and data security. Greenhouse Invasion, one of the scariest threats that IoT can possess as most of the greenhouses and offices rely on automatic processes that leads to the use of IoT devices in these places. IP address of these devices are exposed to hackers who can use it to pinpoint the location and gain access to our personal data just to sell them to underground sites which are the hives of criminal activities.

2.3.3 IoT-Disconnection in Smart-Greenhouse

Internet of things disconnection happens in smart greenhouses due to many reasons. It could be because of a faulty node. Other reasons could be due to the movement of nodes, which could be a random walk to single or both nodes of each link or could be periodic due to satellites and planets movements on orbits. It is good to mention also that some cut offs occur due to the nodes periodic power saving, like low power nodes in sensors networks.

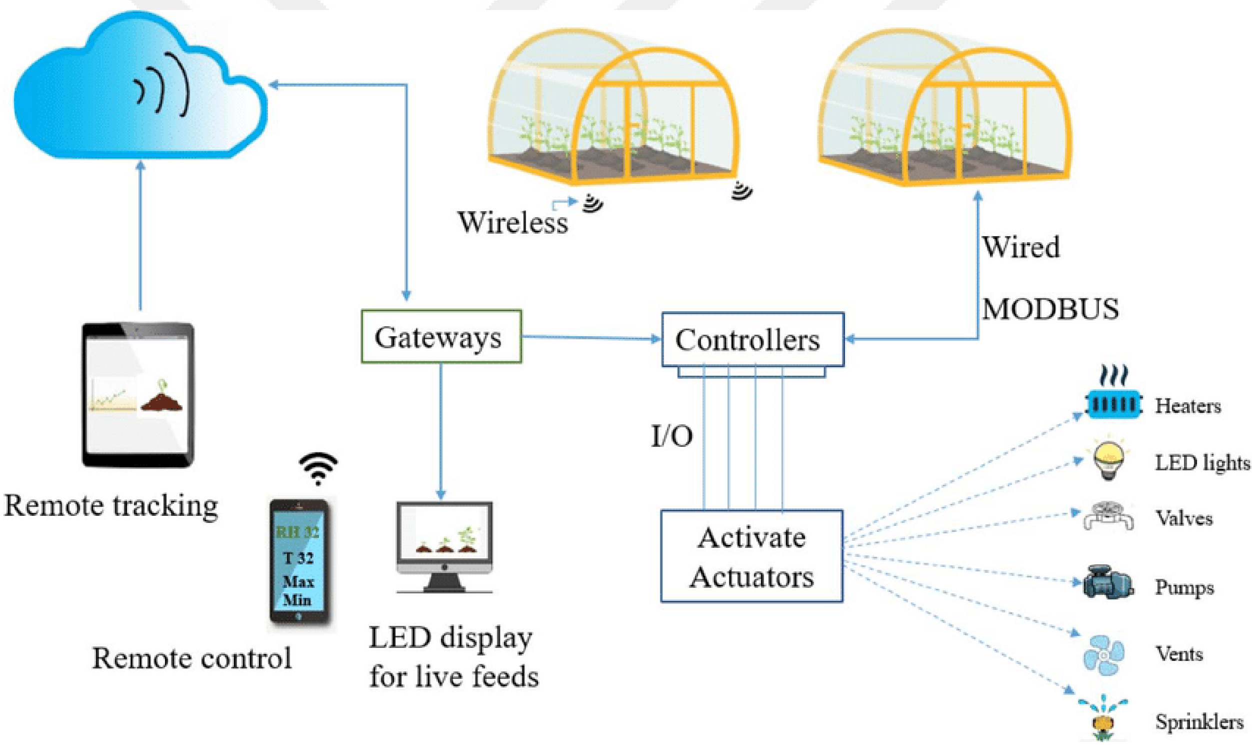


Figure 2.5: Smart Greenhouses Offer Users The Ability To Control Everything In Their Greenhouse Using The Internet Of Things [35].

2.3.4 Unexpected Lengthened Queuing in IoT Networks

In conventional IoT networks, queueing delays could be parts of seconds to seconds in extreme cases. But due to the abnormal circumstances of smart greenhouse queueing delays could reach days if not hours. What contributes to those longer queueing delays is the nature of smart greenhouses where many copies are traversing among the nodes, not to mention the retransmission that takes place when data cannot reach their destinations. Greenhouse Automation is similar to Samsung SmartThings, but is developed by Belkin Company. It provides a full smart greenhouse system, with out-of-the-box smart devices and applications (mobile and web-based) to control and manage the smart greenhouse system. Its architecture is slightly different from Samsung SmartThings, because the devices are managed directly from the applications, without an intermediary hub, since they are WiFi enabled. The system uses the WiFi router to create a smart greenhouse network. To control from outside the greenhouse, it is necessary to add a hub, such a Raspberry Pi as mentioned in [38].

Similar to Samsung SmartThings, Belkin WeMo Greenhouse Automation system does not provide flexibility to add custom made devices, without adding a WeMo product (WEMO Maker) to connect with all other devices, even if the users create devices that communicate through WiFi as mentioned in [39].

2.4 YUKTIX: A SMART GREENHOUSE IN A BOX

The goal of the YUKTIX project is to design and create a smart greenhouse kit that is ready to work” out of the box”. This means the users just have to purchase these kits and with a minimal effort can install and use all of the smart greenhouse components.

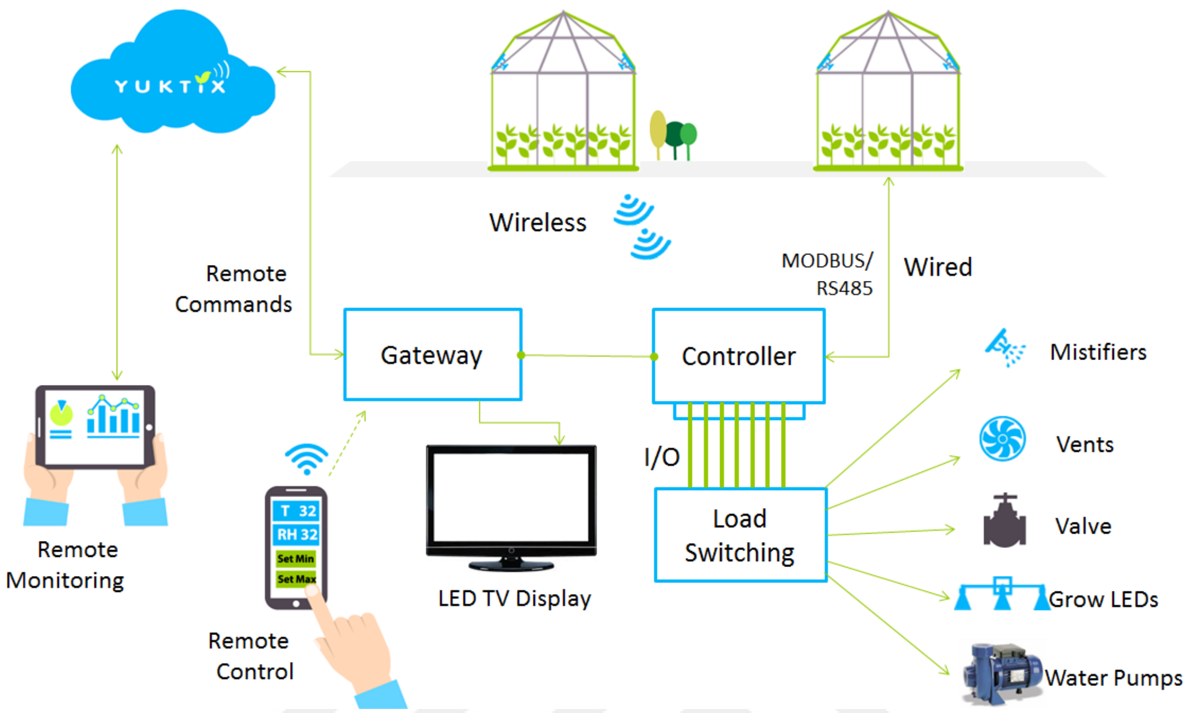


Figure 2.6: YUKTIX System Architecture [40].

The main advantage of these kits is that anyone, without technical knowledge, can create and install a smart greenhouse system. The system components are extendable, allowing the users to buy and add more components after getting this kit. The architecture of this project is illustrated in Figure 2.6. The physical layer uses a Yuktix wireless mesh which communicates directly with the hardware components. It is a good choice, because ZigBee is a low cost wireless technology with low energy requirements, as referred in [40]. The middleware layer works in a publisher/subscriber paradigm, where the manager adds named broadcast channels that allow other components to publish and receive messages. All the components of this system communicate with the publish/subscribe manager through a customized ZigBee bridge. The main goal of this project was to create a product that allows any customer to create and deploy a smart greenhouse system with minimal effort, but it lacks in some important issues. It does not offer a user interface and it does not allow to remotely control the greenhouse appliances. Authors in [41] also do not explain how they provide messages delivery assurance between the devices communication.

2.5 ATOP INTELLIGENT GREENHOUSE SYSTEM

The atop intelligent greenhouse system is an academic project that provides a generic approach to greenhouse automation, independent of any technology. The specification is based in a standard language, XML, to describe the system, supporting inter-operation with different technologies.

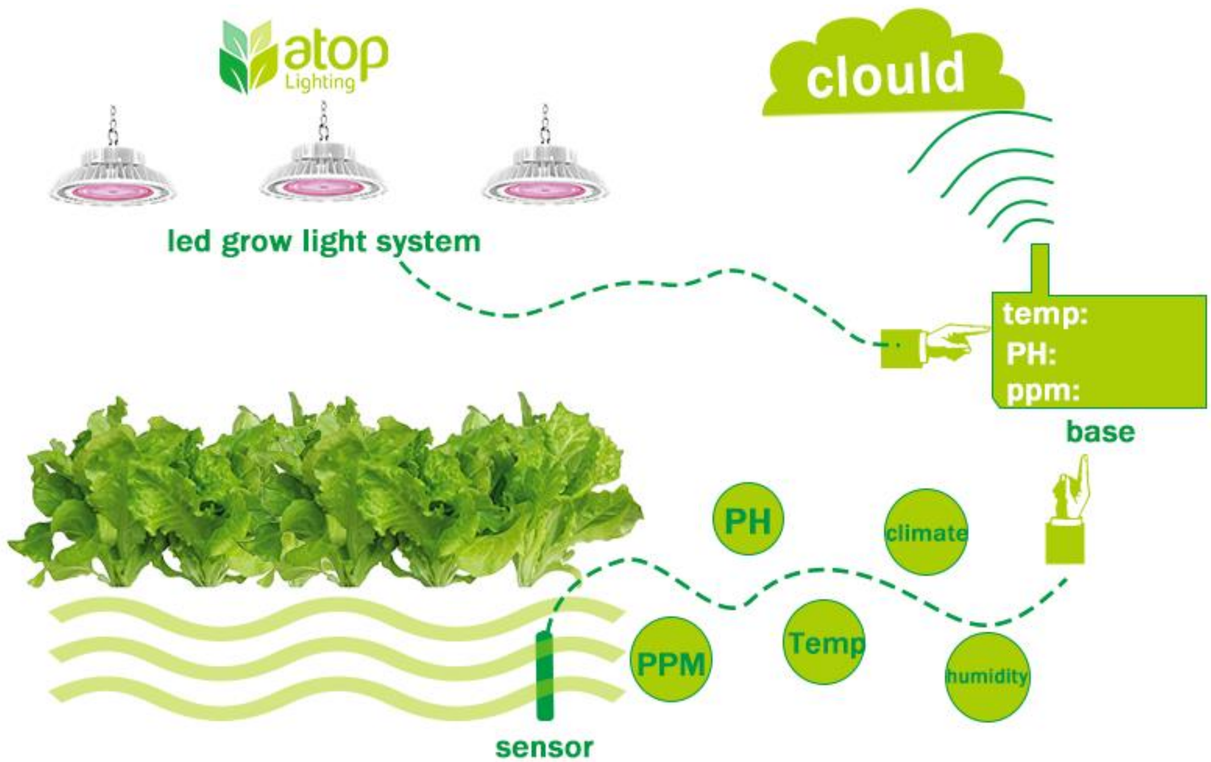


Figure 2.7: Atop System Architecture [41].

This system offers an easy and extendable way to create and define scalable and flexible automation projects, with a distributed architecture. The authors have proposed a multi-level architecture, illustrated in Figure 2.7. The High-Level components, Supervision Modules(SM), are PCs or Raspberry-Pi like boards and perform supervision tasks, interface with users and offer remote access through an internet access point as mentioned in [42]. The Low-Level components consist in Control Modules(CM), which are simple development boards plus interface electronics or power electronics to interconnect with physical devices - sensors and actuators and, optionally, a Router module(R) to allow to interconnect different atop low-level network segments.

In the atop system, a device is an abstract entity characterized by a set of properties where three standard operations over them are available:

h) **GET:** Read a property value;

i) **SET:** Modify a property value;

j) **NOTIFY:** Each device can be configured to, autonomously, notify its atop supervisor (AS) when a property value changes, sending the new value information.

In another project [43], researchers follow the same Device Concept to allow a generic approach and develop a system which can work with all types of sensors/actuators. The communication is established according the SMS (Short Message Service) protocol. It is a good communication protocol when users are in remote locations, without internet access. A GSM modem is connected to the greenhouse automation server which translates the commands received via SMS to greenhouse automation commands. There is also an internet server to receive internet commands (from a webpage). All of these commands are managed by the Main Control Program, which is connected to a database, and is responsible for managing the greenhouse automation system and exchanging commands with the IoT module, that translates the serial commands to IoT commands in order to communicate with the greenhouse appliances as mentioned in [44]. This projects illustrates a different way to control a smart greenhouse, using GSM, Internet and Speech, and we will take advantage of some concepts explored here, such as the Greenhouse Server component that supports different communication protocols and acts as a Gateway to the internal greenhouse automation protocol.

2.6 CHARACTERISTICS OF INTELLIGENT GREENHOUSE

A greenhouse is considered smart when the user assembles a collection of smart devices under one roof and enable them to connect to a network and communicate with one another. The term greenhouse automation is also used because the smart devices work together to automate tasks and greenhouse operation. Smart greenhouses offer users the ability to control everything in their greenhouse, as well as the ability to automate most household’s chores. There is a variety of levels of “smartification” for smart greenhouses, that goes from adopting basic communications to task automation. Researcher in [45] defines three steps towards greenhouse automation namely, adoption of basic communications, simple commands over the greenhouse, automating basic functions, tracking and taking action, prompting activities, and automating tasks. This study summarizes these steps into four steps to make a greenhouse Smart as mentioned in [46].

Automation of Basic Functions: The first step towards the smartification of a greenhouse is to automate basic functions, that is to give way to automatic controls such as room temperature, activating or deactivating the alarm system, running sprinkler system based on a specific program, etc. The second step towards greenhouse automation is to adopt sensor technology into the greenhouse so the users can track their own behavior to determine activity patterns, sleep patterns, or even health status. This step converts the greenhouse into a giant monitoring system as mentioned in [47]. At this point the greenhouse starts generating large amount of data which is handled by programmed algorithms or basic artificial intelligence to make decisions based on the users' behavior

Prompting Activities in Smart Greenhouses: The next step for a smart greenhouse which is already monitoring its users, is activities suggestion. Smart greenhouses are able to remind activities to their uses based on the collected data from the users as mentioned in [48]. At this point greenhouses stop being reactionary, and predictive thus they can be considered "Smart". Behind activity suggestion there must be a relevant data processing happening to the data collected from the users. Based on this data the greenhouse's big data algorithms can predict user behaviors or future need of information.

Automation of Tasks: Automating major tasks in the greenhouse, such as reorder medications, prepare grocery lists, or run the vacuum cleaner, can save plenty of time. Smart greenhouses, learn and adapt so they are able to schedule household's task on their own as mentioned in [49]. The automation of tasks on this level considers the data gathered through sensors and the analysis performed on such data.

2.7 USAGE OF IOT AROUND THE WORLD

The internet of things could be used anytime and anywhere as much early as desired, that could be way ahead of the time to receive any data bundle for usage in the world as mentioned in [49,50,51].

Table 2.1: Statistics of Countries With Megacities Using Internet of Things Based Devices [49,50,51].

Country	No. of Megacities	Population of Megacities	Population of Country	% of population using IoT	Environmental Performance Index
China	6	99,950,000	1,379,000,000	89.2%	43.00
India	5	85,130,000	1,324,000,000	66.4%	31.23
Japan	3	65,045,000	127,000,000	85.2%	72.35
Pakistan	2	34,210,000	193,200,000	67.7%	34.58
Brazil	2	32,750,000	207,700,000	65.8%	52.97
United States	2	36,945,000	323,100,000	71.4%	67.52
South Korea	1	24,105,000	51,250,000	87.0%	63.79
Peru	1	11,150,000	31,770,000	75.1%	45.05
Argentina	1	15,355,000	43,850,000	75.0%	49.55
Philippines	1	24,245,000	103,300,000	73.5%	44.02
Thailand	1	15,645,000	68,860,000	62.7%	52.83
Turkey	1	13,755,000	79,510,000	77.3%	54.91
Iran	1	13,805,000	80,280,000	67.2%	51.08
Egypt	1	16,225,000	95,690,000	57.0%	61.11
France	1	10,950,000	66,900,000	66.4%	71.05
Mexico	1	20,400,000	27,500,000	76.0%	55.03
United Kingdom	1	10,470,000	65,640,000	86.0%	77.35
Congo	1	11,855,000	78,740,000	45.1%	39.44
Indonesia	1	31,760,000	261,100,000	62.2%	44.36
Russia	1	16,710,000	144,300,000	51.6%	53.45
Viet Nam	1	10,380,000	92,700,000	61.2%	38.17
Bangladesh	1	16,820,000	163,000,000	60.3%	25.61
Nigeria	1	13,360,000	186,000,000	67.2%	39.20

The context of the internet of things usage in the world. The cities count with an international population demographics and plays an important role within the global economic. It belongs to the group of megacities with a population of 10.5 million by 2019 as mentioned in [52]. The scenario could be either good or bad depending on the whether megacities are considered as a positive or negative thing for the agenda on sustainable development. By attempting to correlate "countries with the highest number of megacities" with the "World Environmental Performance Index" it can be inferred that there is no correlation between countries with high percentage of population living in megacities and environmental sustainability as mentioned in [53]. Table 2.1 shows all the countries with megacities by 2020. China, India and Japan have six, five and three megacities respectively. Their environmental index does not change accordingly with their number of megacities or either with the percentage of population in megacities.

3. METHODOLOGY

Recently we hear more than ever about internet of things (IoT), artificial greenhouse cognition, and greenhouse monitoring. These terms are similar in how they sound, but they are different in other ways. This chapter will provide the implementation details IoT based intelligent greenhouse automation systems in general, and that includes, MQTT protocol for handling the automation process with greenhouse server modules and for this purpose source of programming language being used is python programming 3.8 on anaconda navigator platform with both CPU and GPU utilization. The basic flow diagram has been shown in Figure 3.1.

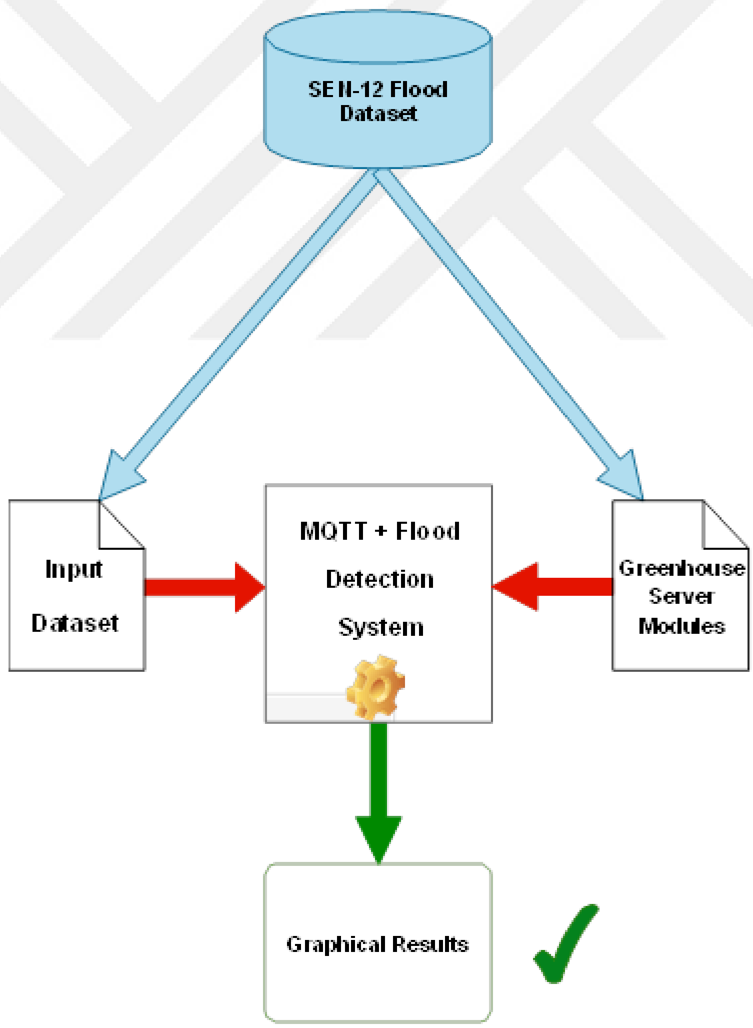


Figure 3.1: Flow Diagram Of Approach Being Followed.

3.1 INTERNET OF THINGS ELEMENTS

We can summarize the involved IoT elements in the following categories:

Users: Users are people that will interact with or live inside the installation.

Sensing Devices: Sensing devices are IoT devices that are installed and have the role of data producers. Their information can be transferred, consumed or collected for future use.

Actuator Devices: Actuator devices are IoT devices that can be controlled to invoke change in the physical world. They include elements like light switches or the set value of the thermostat in an HVAC system.

Gateway Devices: Gateway devices are IoT devices that simply have the role of transferring information between the installation and the Internet. Their role is crucial in installations where the other IoT devices installed are not Internet-enabled and cannot communicate directly with the rest of the system.

Observed Phenomena: Observed Phenomena refers to the physical phenomena that can be observed from a digital sensing or actuator devices.

Units of Measurements: Units of Measurements refer to the convention used to quantify an observed phenomenon.

Locations: Locations refer to the physical and logical groupings that can be applied to any of the elements presented above. They can describe a building, a school unit or a classroom but also a more abstract grouping of a set of users inside multiple school communities. In a similar manner there are relationships between the elements presented above that are defined as follows:

Ownership: Ownership defines the relation in which users have the full control and permissions on a set of sensing, actuator or gateway devices and physical locations. This relationship is important for administrative and security purposes and can be used for the management of the whole installation.

Access: Access defines the right of a user to view the information produced by a set of sensing devices or physical location or control an actuator device. Such rights can be limited based on the type of the user as for example teachers and students may need access to detailed information only about their own classroom while they are allowed to view only limited data on the rest of their school building. Similarly, students may not be allowed to control actuator devices even in their own classroom as elements like the HVAC system is configured only by their building managers or an authorized teacher (e.g., the principal).

Sensing Attributes: Sensing attributes refer to the sensing and actuator devices and are used to define what type of information each one of the devices generates, or what type of device it controls. The can be information about the physical phenomenon or the measurement unit that is measured or even whether the data generated are raw values or need a post processing to be valid.

3.2 DATASET DESCRIPTION

For this research, SEN12-FLOOD: A SAR and Multispectral Dataset for Flood Detection has been acquired from an open-source IEEE DataPort Repository. The research worked on this dataset on python programming platform. The datasets files reflect how smart greenhouse and flood detection system act in space; nodes have sight of contacts with each other from time to time for certain periods of time intervals. Those Datasets reflect nodes behavior inside their flood data nodes, whether they are active (the node gets connected or disconnected due to its movement), or passive (the node is stationary, but it gets connected or disconnected due to other nodes movement), or a combination of both. The dataset includes cleaned flood detection data in Watts for different geographical location at aggregate and appliance level, timestamped and sampled at multiple second intervals. The reason behind using larger dataset of 12 GB is that there is need for large dataset if the goal is just experimenting to compare performances rather than training to reach high accuracy. Furthermore, larger dataset means less time to perform experiments and hence more time available to do even more experiments. The dataset could be downloaded from the link: https://ieee-dataport.org/open-access/sen12-flood-sar-and-multispectral-dataset-flood-detection

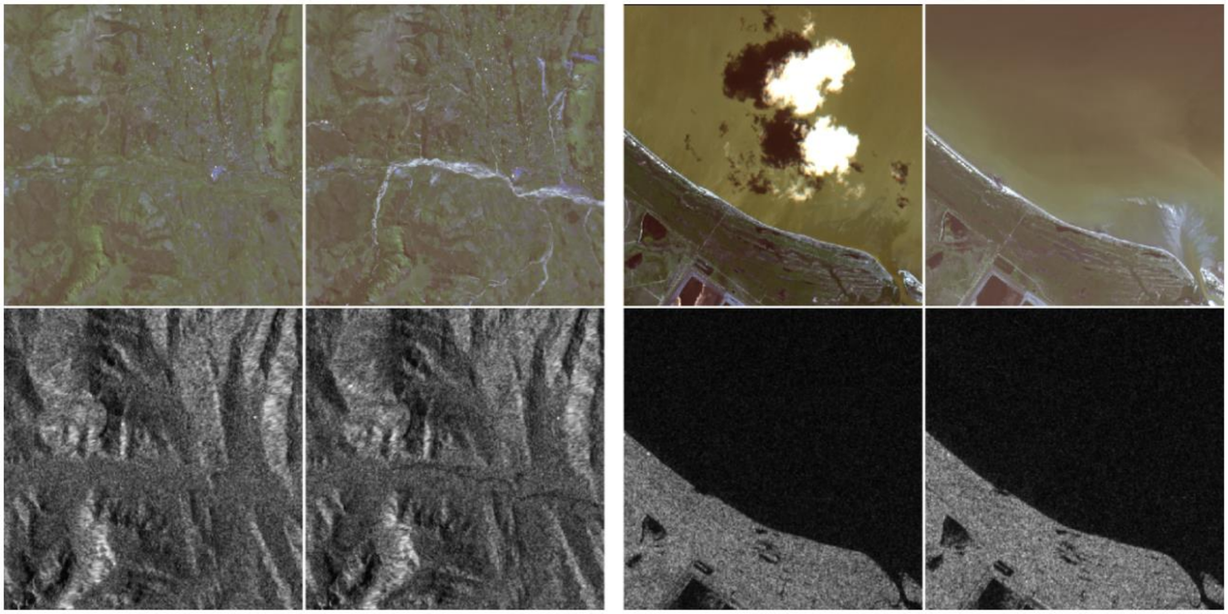


Figure 3.2: The SEN-12 Dataset As A Flood Detection Instrumented And Networked For Monitoring Of Geographical Locations.

3.3 SYSTEM ARCHITECTURE

The system architecture we propose is aligned and proper designed to be deployed in a multi-layer smart greenhouse system, as illustrated in Figure 3.3. Using a layered architecture, we guarantee flexibility to add and remove components in our system and we provide interoperable services that can be specialized in the execution of specific tasks. This architecture allows to simplify the amount of data that low-level devices have to handle with, which is proportional to their hardware limitations.

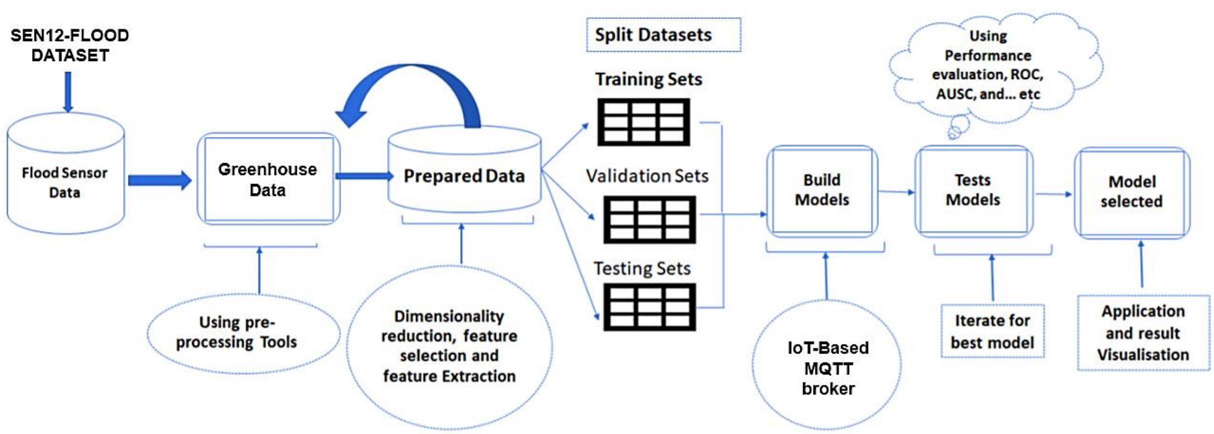


Figure 3.3: Intelligent IoT Based Greenhouse Monitoring System With Flood Detection Architecture Using The Iot Based MQTT-protocol.

3.3.1 Intelligent Greenhouse Server

The Greenhouse Server is the core component in our system and it could be defined as a smart hub, since it allows the connection between different types of components from the internal network and also provides public access (from external network) to some of these components.

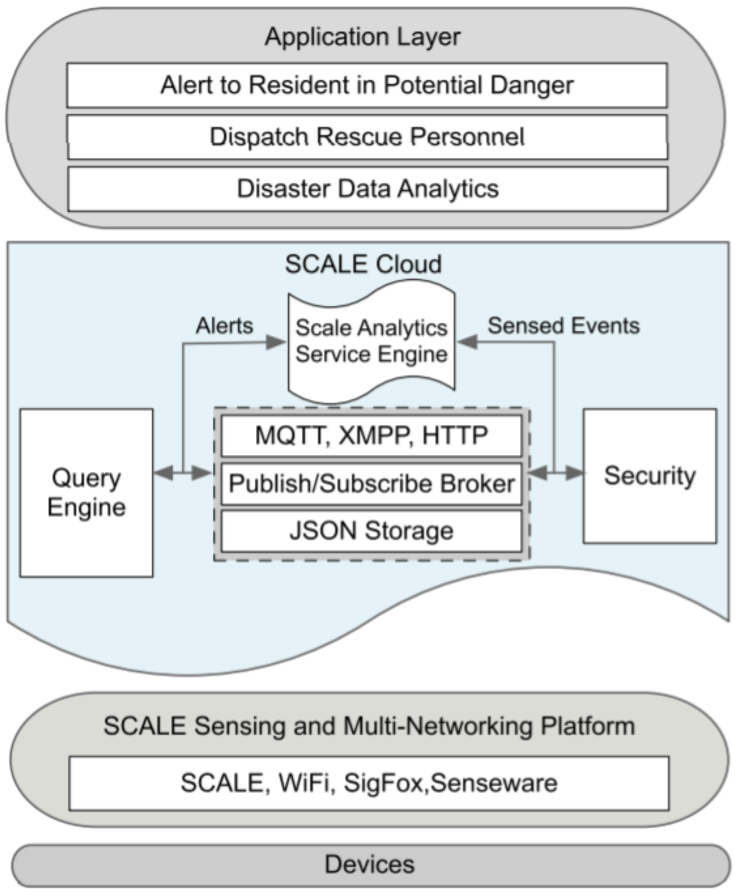


Figure 3.4: Intelligent Greenhouse Server Modules

The Greenhouse Server, illustrated in Figure 3.4, is a complex system with different modules where each one is responsible for a feature. The modules are:

a) Public Access: The greenhouse server provides public services in order to clients control and manage the smart greenhouse system. These services are mainly RESTful services in python programming accepts HTTP requests with JSON data content and returns HTTP responses.

b) Security: The greenhouse server has a security layer requiring an authentication phase from clients to access the smart greenhouse system, using digital certificates. This security layer is customizable and it allows to identify users and their user roles.

c) Internal Devices Manager: The greenhouse server is the core component to manage the devices connected and it is necessary to allow the discovery feature: a mechanism that allows new devices to know what are the available devices, their properties and the topics they should use to communicate with other devices. The Internal Devices Manager also provides administration services, such as, the creation of new rooms, user and user roles, as explained in the next section.

d) User Permissions Manager: There is a component in the greenhouse server responsible to manage the user permissions. Users have specific roles and permissions to control the devices and this module evaluates, for each request, if the user has permissions to perform the required actions.

e) Complex Actions Manager: To create an automated and smarter greenhouse system, the greenhouse server allows users to create complex actions. These actions are defined by a set of preconditions (e.g. other devices' states) and the corresponding actions that should be triggered when the preconditions are verified, for example, turning on A/C when the greenhouse temperature is 26oC.

3.3.2 The IoT-Based MQTT Broker

In the solution we propose, the devices communication is made using MQTT protocol due to all its advantages. This is related with its simplicity, the easy implementation and because it is optimized for high-latency and unreliable networks. Thus, it is mandatory to have a MQTT broker (a MQTT server) that allows this communication. The communication transport we use is a WiFi network and the MQTT broker also provides security mechanisms, such as using certificates to authenticate devices.

3.3.3 Database

An auxiliary database is needed to store all the system data, the user permissions, the devices connected and their state. The goal of using an external database is to reduce the amount of memory needed for server and devices, increasing the system's performance, and also to provide flexibility to scale the system.

3.3.4 IoT Devices

The devices are the IoT hardware components that are connected to the smart greenhouse system, usually constrained in terms of memory usage and power consumption, and are the sensors/actuators that clients will control remotely. There are three key concepts in our system architecture that are important to understand the following sections:

a) Device: An entity that represents a micro system, such an Arduino-board like, and have components.

b) Component: Something connected to a device, such a lamp, a toaster or a heater. Each component has specific properties.

c) Property: An attribute-value pair related to a specific component. For instance, a lamp can have two properties: Mode, with ON and OFF values and DIM, to control the brightness with a range from 1 to 10.

3.3.5 Clients

The clients are the external users that connect remotely to the smart greenhouse system to control the devices or just to retrieve the devices' state. The access could be done using an application, with an user interface, or could be eventually M2M clients, e.g. an external IoT system. Since our system has a user permission mechanism in the greenhouse server, all the clients must have a unique identifier that match with the one defined in the user permissions, otherwise they will be assigned with the default user permissions. The permissions could include access rights to specific devices or to groups of devices, for instance it is possible to define that a user can access the values of all the kitchen devices, but cannot change them.

3.4 IOT BASED INTELLIGENT GREENHOUSE SERVER

An intelligent greenhouse server is a complex system that includes smaller systems working together. The sub- systems are described in the next subsections.

3.4.1 Public Services

The public services are the services that our smart greenhouse system make available for external users. They are available in the following URL: https://{PublicServerIP}/MultiDevicesREST/api/client/{service}. In Table 3.1 are described the web-services available.

Table 3.1: Client Server Methods for Smart System.

Service	Type	Description
Verify	GET	Check if the server is available.
Devices	GET	Show connected devices.
device/{deviceID}/components	GET	Show the components of the {deviceID} device.
component/{componentID}/properties	GET	Show the properties of the {componentID} component.
property/{propertyID}	GET	Show the attributes of the {propertyID} property.
Properties	PUT	Change a set of properties' values. Sends in JSON format.

The public services are RESTful web-services, that consumes/produces JSON requests/responses. With this approach we guarantee flexibility and freedom to implement different applications, in any platform, without compatibility concerns. It could be a mobile application, a smartwatch, a web- browser application or even another IoT system. The only requirements are that clients should be able to connect to the internet and invoke these services using HTTP protocol with contents in JSON format data.

In our architecture the usage of web-services, makes the system easier to extend with other services that we might want to implement in the future, without changing the services that are already developed. The public services were developed according to a Service-Oriented Architecture (SOA) where they are" black boxes" for customers and the services are independent from each other.

Another great advantage using RESTful web-services are related with the system scalability. The server end of REST is stateless, which means that the servers do not have to store state or data across requests. Thus, the communication between servers are minimal, making it highly scalable. Also, the RESTful web-services are representational, which means that is possible to provide a good load balancer on the server-side that can easily route the requests to the right servers, according to the URLs and the HTTP methods, for instance, the GET requests could go to a group of servers while POSTs go to a different one.

3.4.2 Private Services

Private services are also RESTful web-services, that handles JSON format data, but for internal usage. Internal means that the services are only available to the internal smart greenhouse system network, for devices and for administration purposes. The private services available are described in Table 3.2. They are accessible using the following URL: http://{PrivateServerIP}/MultiDevicesREST/api/device/{service}.

Table 3.2: Device Server Methods for Smart System.

Service	Type	Description
verify	GET	Check if the server is available.
devices	GET	Show connected devices.
devices	POST	Add new devices. Sends in JSON format the devices attributes.
device/{deviceID}	DELETE	Remove the {deviceID} device.
components	POST	Add new components. Sends in JSON format the components attributes.
device/{deviceID}/components	GET	Show the components of the {deviceID} device.
component/{componentID}	DELETE	Remove the {componentID} compo- nent.
component/{componentID}/properties	GET	Show the properties of the {componentID} component.
property/{propertyID}	GET	Show the attributes of the {propertyID} property.
properties	POST	Add new properties. Sends in JSON format the properties attributes.
property/{propertyID}	DELETE	Remove the {propertyID} property.
properties	PUT	Change a set of properties' values. Sends in JSON format.
MQTT-broker	GET	Get the MQTT broker URL.
rooms	POST	Create one or more rooms. Sends in JSON format.

According to the best practices of RESTful web-services architecture design, we have used the HTTP methods to differentiate the available types of operations. GET operations are used to retrieve data in a read-only way; DELETE operations to delete resources, that can include properties, components or even devices; POST operations are used to create new resources; and PUT are used to update resources. The purpose of these service will be described in the following subsections.

Figure 3.5: The basic processing of smart greenhouse system with flood detection & monitoring involving the public and private services with IoT based MQTT broker.

3.4.3 Internal Device Management

We use the MQTT protocol to allow communication between devices, due to all of its advantages. But there is one great disadvantage in a Pub-Sub architecture: the subscribers must be aware, a priori, which topics are available to subscribe. This is a big disadvantage and it is completely in disagreement with our system's goal, that is mainly a very generic and very flexible system, that allows the addition and the removal of devices ad hoc.

Thus, to take advantage of MQTT protocol and also to provide flexibility in our system we create an Internal Devices Management System that works together with the MQTT broker and manages all the devices connected, their components and their properties. With this system, the devices do not need to know, before being deployed in our network, what are the devices available and their properties. This is an additional system but it is necessary to allow devices to know, after their configuration and with the system up and running, what are the network properties and the devices connected.

The internal devices management system should always be synchronized with the real system information; thus it is mandatory that devices in their initialization connect with the greenhouse server (in the respective services) to register their information. A more complete example is illustrated in the Examples of Usage section, but here is a brief explanation about how this internal device management system works:

We have a chandelier (a device) that we want to connect in our smart greenhouse system. During the initialization the device should register itself in the greenhouse server using the private service. /api/device/devices, and register their components and properties. So, the chandelier will also make a request to /api/device/components telling that it contains a Lamp (component) and this lamp has a Mode (property) that allows the values ON or OFF. Using this mechanism, the Internal Devices Management system will always know the network state, the devices connected and their attributes. It allows to add new devices to the network ad hoc without resetting the entire network or updating the devices code.

3.4.4 Complex Action Management

An intelligent greenhouse system to be smart should provide smarter mechanisms than just turning devices ON or OFF, or just reading sensor values. In order to achieve the desirable intelligence, our system provides a Complex Actions Management System that allows the creation of flows of execution according to specific conditions, such as device components' states. The system consists in a set of preconditions (events), defined by users, and a set of actions that are triggered when the events occur. An example of how this system could work is: when the greenhouse temperature is 26oC, turn on the air conditioner. It also could more complex, such as, when the outdoor light sensor detects that exists a lack of sunlight and a person in a specific room, it will automatically close the electric window blinds and turn on ambient lighting.

Although we did not implement this feature in our system, it could be implemented using an approach that consists in a descriptive language that allow users to describe the complex actions. A polling system will check periodically if the properties' values match with any conditions defined and, if they do, the system will execute the corresponding actions.

3.4.5 User Permission Management

In a real smart greenhouse system, it is useful to have different users with different user roles and permissions according to their responsibilities. For instance, in a family with two parents and one young child, where everyone can control the smart greenhouse system, it is not desirable that the child can access and control dangerous devices, such as an oven or a cooker or even critical house rooms, as a kitchen for instance.

Thus, it is a very important feature to have a User Permissions Management System responsible for managing the different user roles and different user permissions.

In our system we propose a User Permission Management System that supports two types of user concepts:

a) Users: profile that represents a specific user;

b) User Roles: set of user groups that should be assigned to users' profiles. Some examples of user roles include: Guest, Regular User, Child, Gardener, Housekeeper, among others.

The permissions priority level is from the most generic to least generic, in this scenario, from user roles to users and it allows the following scenario: if there is a Child as user role, that allows the access to all house rooms except the kitchen, and a user had assigned this user role, he will not be able to control kitchen devices. But if his parents want him to control the microwaves, they can assign a specific permission to control this device and he will be able to control only this device, even though his user role does not allow.

The above explanation was about the difference between user profiles and user roles, but in our system we also propose to add another level of permissions: devices specific permissions and rooms permissions.

In conceptual terms, a house has one or more rooms and each room has several devices. Thus, it is important to have a higher level of hierarchy, where users can control permissions regarding the rooms or more specifically, the devices.

To manage this feature, we applied the same priority level mechanism, from the most generic to least generic. So, the permissions defined for a room have a lower priority than a device specific permission.

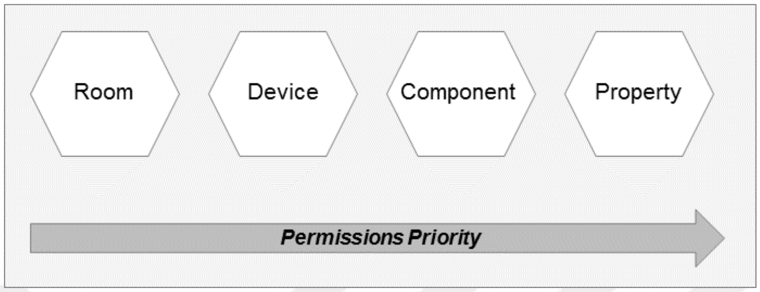


Figure 3.6: Permissions Priority, from lowest (left) to highest (right).

This mechanism allows the following scenario: The user, since he is a child, does not have permissions to control kitchen devices. So, their parents define a rule that deny accesses from user to room kitchen. But after some time, their parents want to give him access only to the kitchen television. Thus, they only need to add one rule that allows his user profile to manage the kitchen television, and the remaining kitchen devices will continue to be inaccessible from his user profile.

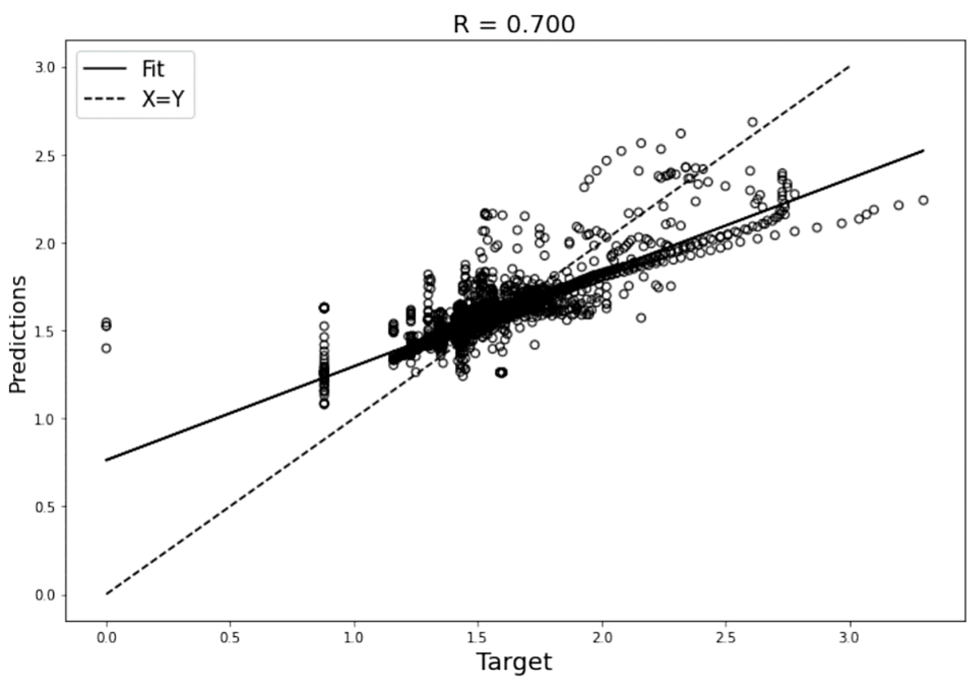


Figure 3.7: The Scatter Graph Of Simulated Prediction Of Flood Target With Different Range of points generated by the MQTT broker to control warning signals.

3.4.6 Authentication and Security

The external users must be authenticated to access the available greenhouse server services. The authentication should be done using digital certificates. Whenever clients want to connect to a public service, they should authenticate using their own certificates. The communication should also be done using HTTPS, which makes the conversation with the web server entirely encrypted. As we have discussed in the subsection User Permissions Management there are two types of user concepts: Individual Users and User Roles. Thus, the digital certificates could be generated for a user role or even for specific users. The greenhouse server will use this information in the authentication phase to identify which user profile it will use.

3.4.7 Minimum Requirement

The minimum requirements to configure the server node are dependent of the application server we use, in this case to use with system with following specifications:

a. Python 3.7 or later;

b. WiFi Network Card;

c. WiFi Access Point;

d. 32-GB of RAM;

e. Windows 8.1 or later or Linux or Mac OS X.

3.5 MQTT BROKER SERVICE IN AUTOMATION

The MQTT Broker in our system has more responsibilities than just allow the MQTT protocol communication between devices. Our MQTT broker is connected directly to the database and when a property' value is changed the MQTT broker will update the database with the corresponding value. There is a callback function on the MQTT broker side that subscribes all +/change topics and updates the database with the new values whenever a new message is published. One example of a message with a level 2 of QoS, could be a fire alarm. In this case, we want the system to keep trying to deliver the message until it finally reaches the right destination. The QoS levels of each topic are defined in the device registration phase and they could be easily changed afterwards. They are defined to a specific property, which provides great flexibility to manage the QoS levels according to the most fined grained level of devices. If the QoS level is not defined, the default value is 0 (the lowest and fastest one).

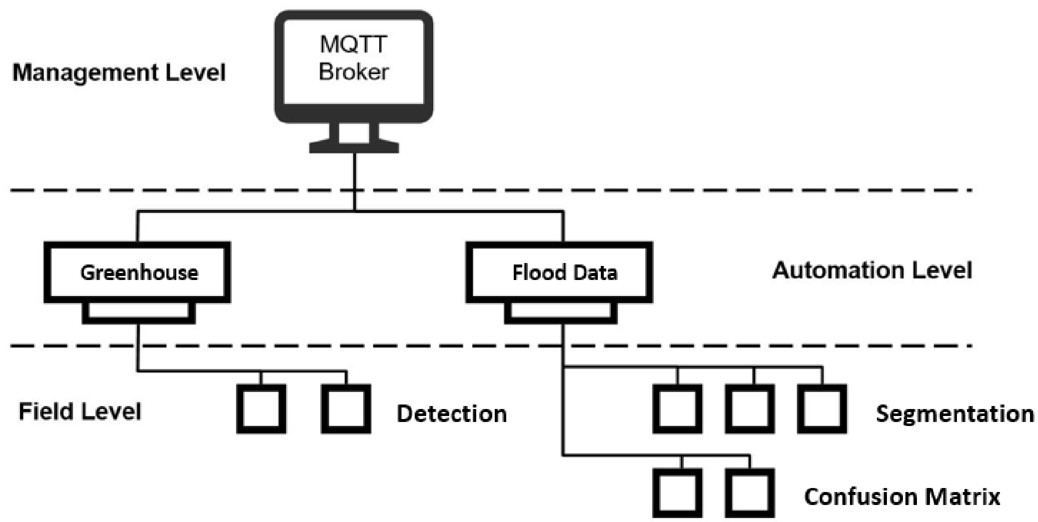


Figure 3.8: The Three Tier Architecture Of Smart Monitoring System With Flood Detection Using MQTT broker.

Another MQTT feature that we use is the QoS internal system that is responsible for the delivery assurance of each message. In our system we allow devices to subscribe/publish messages from the three available QoS levels, but our recommendation is that critical messages should have the maximum QoS level: 2, that guarantees that messages are delivered and also it is a non-repeatable operation, which means messages are not sent twice or more times.

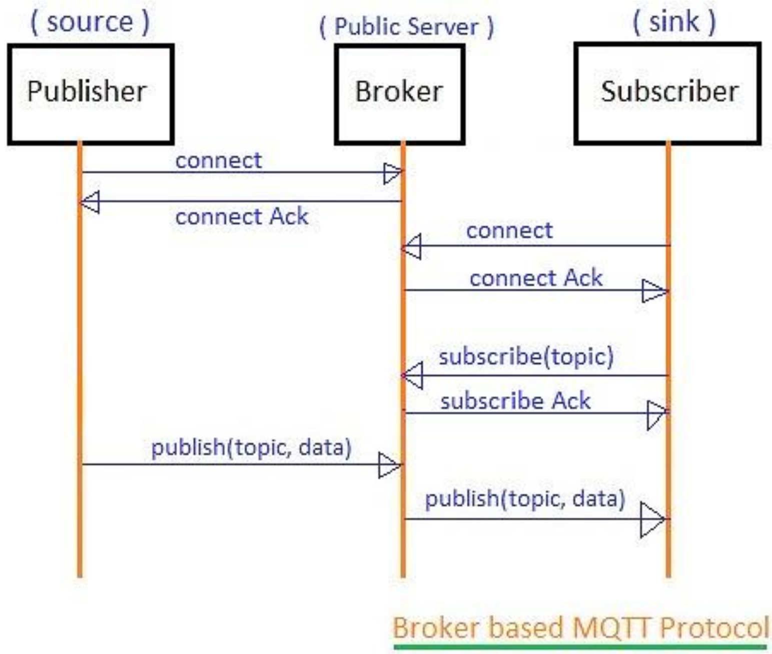


Figure 3.9: The General Sequencing of MQTT broker for managing the actions.

Using a system with an external module (server) for the MQTT broker also allows the communication between devices without the additional" work" to the greenhouse server. This is useful to reduce the payload in the greenhouse server and, if the users want to connect devices that have better computational resources, that can handle some" complex" actions for themselves, they can have devices managing actions directly from other devices, without the greenhouse server acting as an intermediary in this process. The disadvantage of this approach is the additional complexity that it adds to manage all of these complex actions, with some actions defined at the greenhouse server level and others directly in the devices.

In our implementation we also take advantage of the MQTT retained messages. These messages are defined in the MQTT protocol and we use them with the purpose to let the other devices known when a device is disconnected and also to store the last message. The latter feature allows that a recently connected device can know immediately which is the last state of a specific property or component of other device, without waiting for a new message to get an updated state.

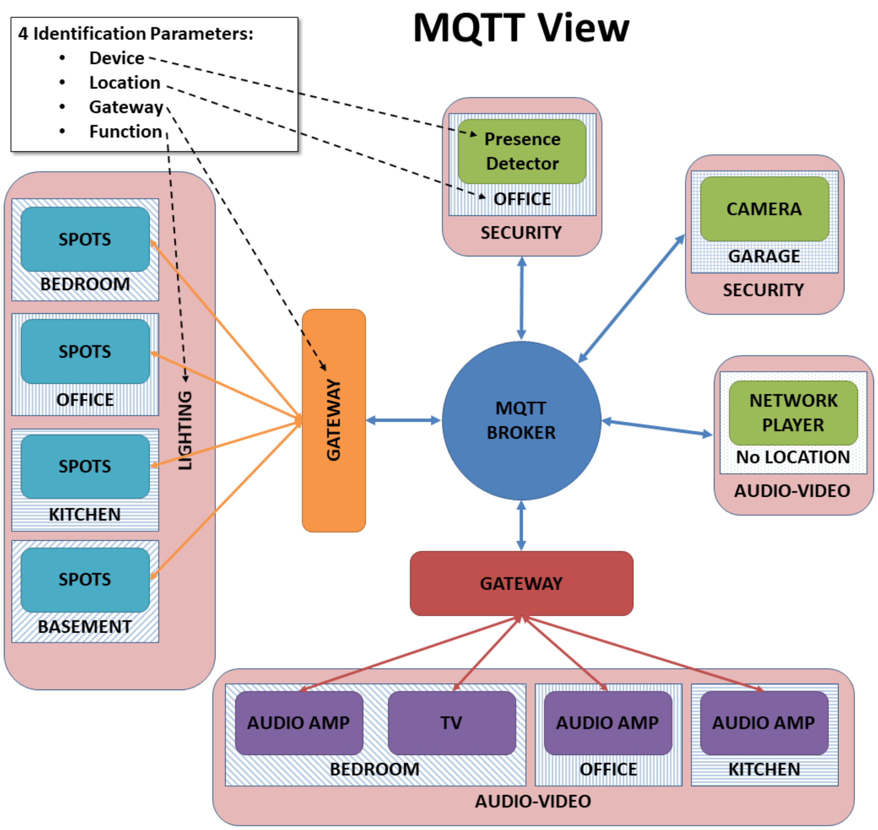


Figure 3.10: An IoT-Based MQTT Modular View Schema for intelligent automation system.

4. RESULTS

Our system was designed to enable the usage of any low-constrained device type in terms of memory and processing power. Thus, our requirements are minimal and the complexity of our system is in the greenhouse server side, that is why a device should only be concerned about its own components and properties, substantially reducing their hardware requirements.

The minimum requirements are presented and it is basically a device that can connect to the internet via WiFi: it should have implemented the TCP/IP protocol and be able to perform HTTP request/responses due to MQTT Protocol and greenhouse server operations.

An additional feature that our system allows is the creation of Gateways to connect more basic devices that do not have the minimum requirements. The idea is that the user connects to one or more basic devices, a device slightly more powerful, e.g., an Arduino with an WiFi antenna, and this device acts as a gateway to connect the simple ones into our smart greenhouse system.

This is possible because in our system we can easily add or remove devices ad hoc and a gateway in the middle of the communication is transparent for the whole system. This gateway should be powerful enough to internally manage information related with the basic devices, e.g. the internal representation and the internal mapping used in the corresponding communication protocol.

This system allows to use any client in any platform: it can be a smartphone application, a smart-watch application, a web-page or anything else. The only requirement is that the application need to be able to implement HTTP protocol, because the clients should be able to perform RESTful API calls. Similarly, to the devices, if the client has a JSON library the required code will be simpler, but this is not mandatory.

The clients in our system could be from two different types: other IoT systems, and in this scenario they don’t provide any user interface (it is a typical scenario of M2M communication) or they are designed to offer user interaction, and in this scenario they should be designed according the best usability practices. In our system implementation we have not created any user interface (is out-of-scope), we used only the command line to test the client REST API calls, which proves that the client minimal requirements in Figure 4.1 This blue color depicts the area which can be flooded with water and blue color is used for the detection of flood water, everything in the figure which is blue is considered flooded, however the purple color depicts the area that is under threat of flood and vulnerable to flood. The building that are in normal color are above the floor and cannot be flooded and you can clearly see the roofs of all buildings that are safe and can be clearly seen from above.

```
[[201 255  55   4]
 [ 43 300  54  49]
 [  0   0 133   0]
 [  0  62   6 177]]
```

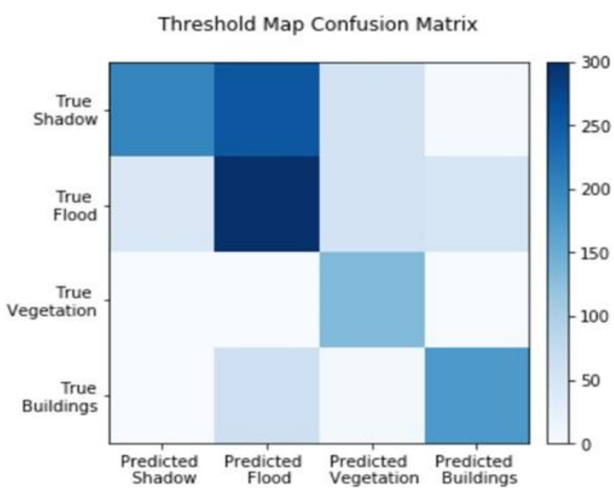
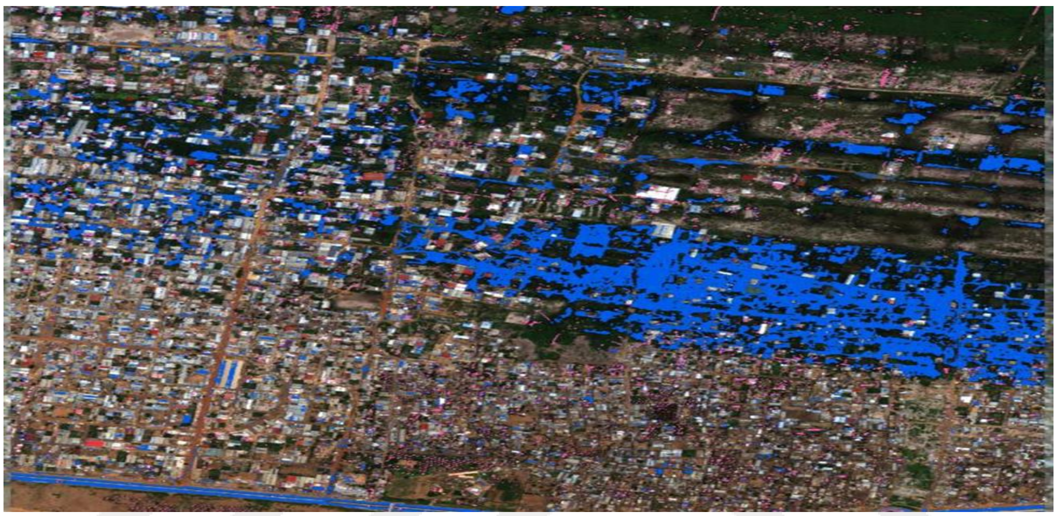


Figure 4.1: The Viable Flood Threat Distinguished with respect to segmented region in the map along with its confusion matrix depicting the performance of classification using the MQTT protocol For Sample-1 From Dataset.

In Figure 4.2 This blue color depicts the area which can be flooded with water and blue color is used for the detection of flood water, everything in the figure which is blue is considered flooded, however the purple color depicts the area that is under threat of flood and vulnerable to flood. The building that are in normal color are above the floor and cannot be flooded and you can clearly see the roofs of all buildings that are safe and can be clearly seen from above.



```
[[ 61  72  25   0]
 [ 34 190   8  18]
 [  0   1 159   0]
 [  1  10   6 257]]
```

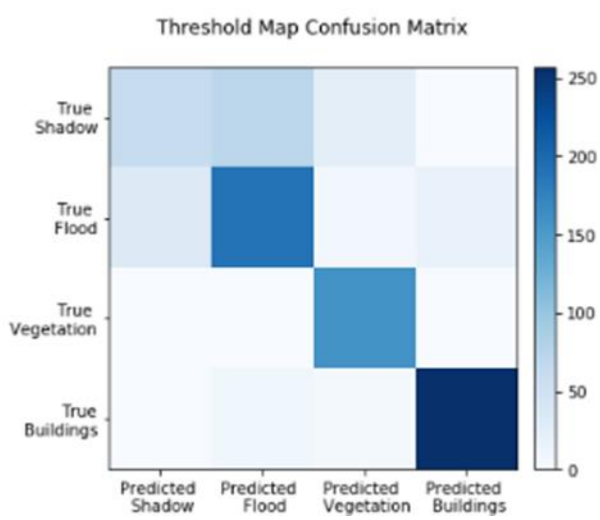
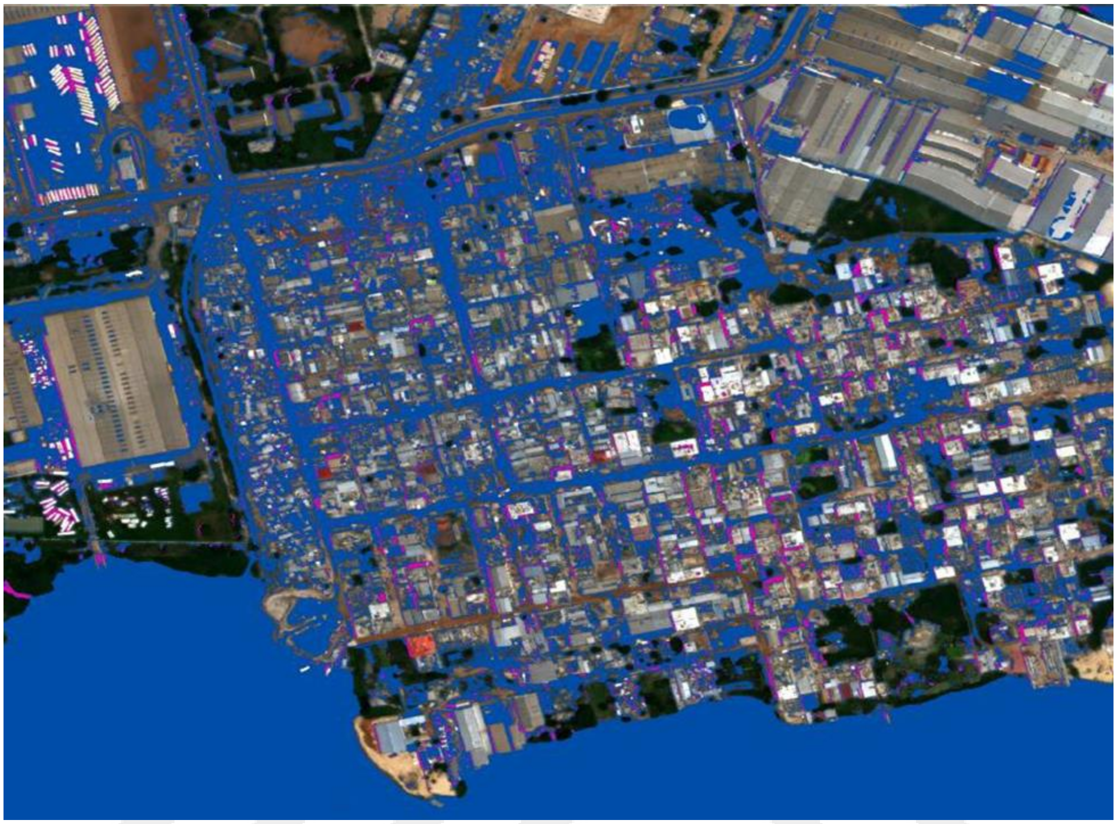


Figure 4.2: The Viable Flood Threat Distinguished with respect to segmented region in the map along with its confusion matrix depicting the performance of classification using the MQTT protocol For Sample-2 From Dataset.

To illustrate how the system works, we will explain an example of a system composed by one client, one greenhouse server and its IoT server, one MQTT broker and one device with two components, a blue color for the 25% transmission failure and a pink color for the 20% transmission failure. Each of these components have one property, mode, whose values can be 0 or 1. This example illustrates how a device should setup their registration and how the MQTT broker can change a flood detection value. This dataset reflects nodes behavior inside their flood data nodes, whether they are active (the node gets connected or disconnected due to its movement), or passive (the node is stationary, but it gets connected or disconnected due to other nodes movement), or a combination of both. The dataset includes cleaned flood detection data in Watts for different geographical location at aggregate and appliance level, timestamped and sampled at multiple second intervals. In figure 4.3 This blue color depicts the area which can be flooded with water and blue color is used for the detection of flood water, everything in the figure which is blue is considered flooded, however the purple color depicts the area that is under threat of flood and vulnerable to flood. The building that are in normal color are above the floor and cannot be flooded and you can clearly see the roofs of all buildings that are safe and can be clearly seen from above.



```
[[275   7   0   0]
 [ 16 259   2   3]
 [  0   0 116   0]
 [  0  14   0 193]]
```

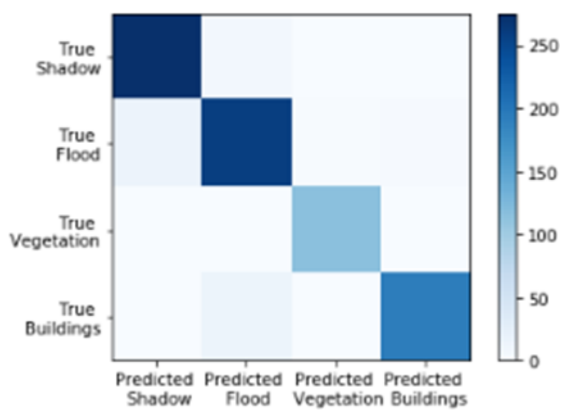
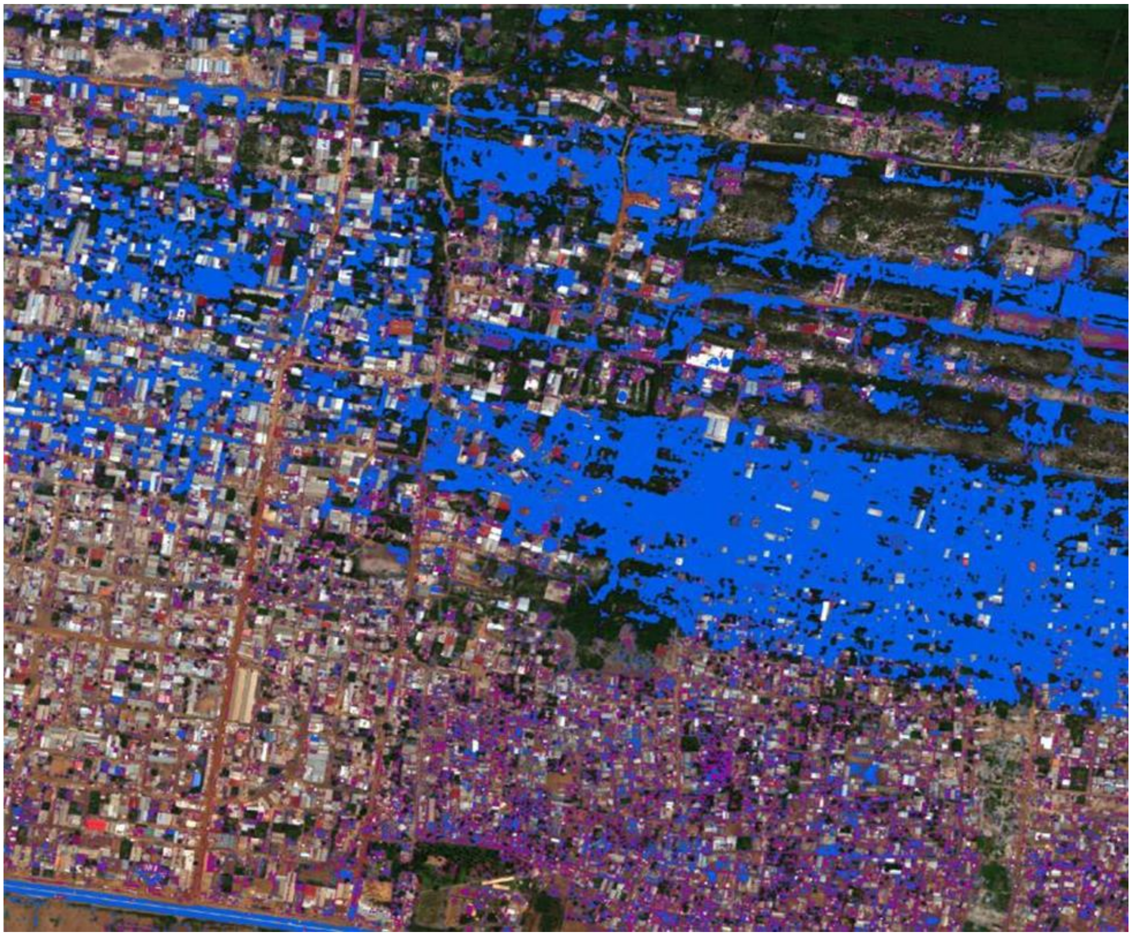


Figure 4.3: The Viable Flood Threat Distinguished with respect to segmented region in the map along with its confusion matrix depicting the performance of classification using the MQTT protocol For Sample-3 From Dataset.



```
[[275   7   0   0]
 [ 16 259   2   3]
 [  0   0 116   0]
 [  0  14   0 193]]
```

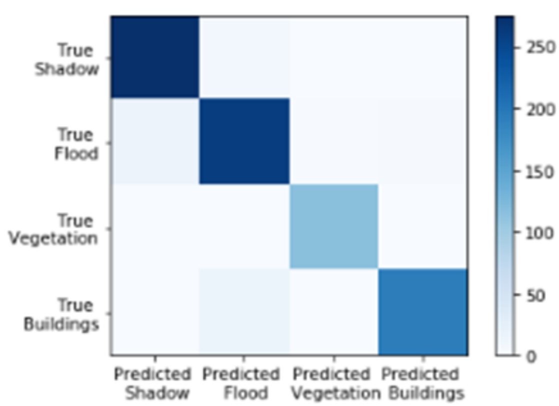
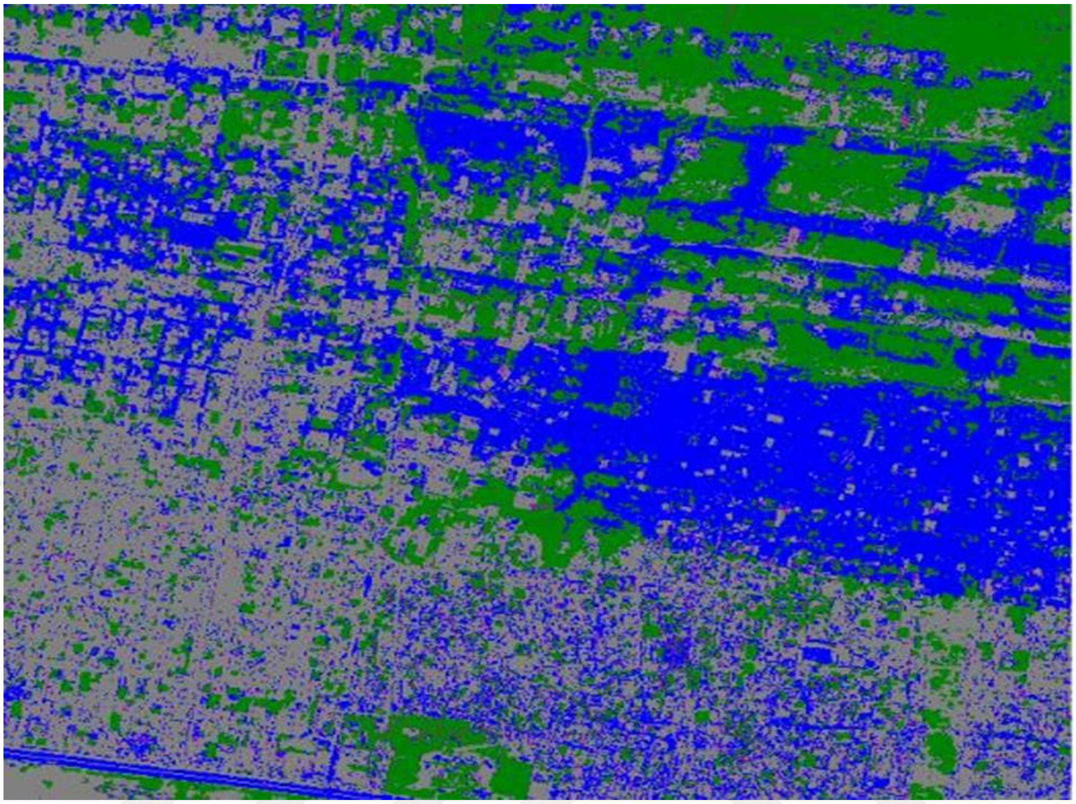


Figure 4.4: The Viable Flood Threat Distinguished with respect to segmented region in the map along with its confusion matrix depicting the performance of classification using the MQTT protocol For Sample-4 From Dataset.



```
[[271  11   0   0]
 [ 28 229   3  20]
 [  0   0 116   0]
 [  0  11   0 196]]
```

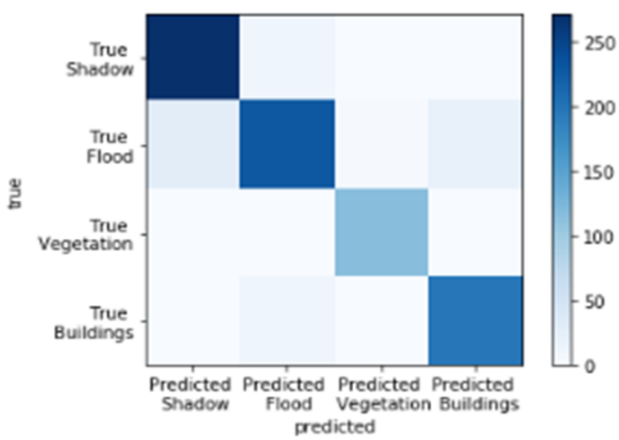
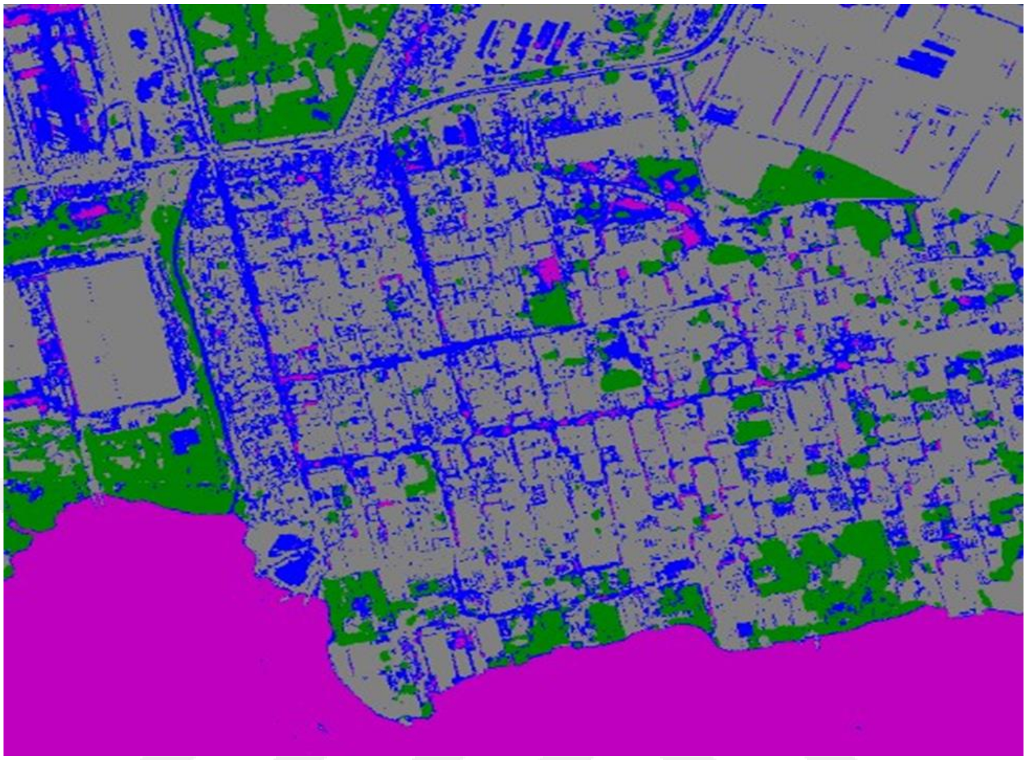


Figure 4.5: The Segmentation of Surface from the sea being classified as different regions where the blue color depicts the area potentially threated with flood with threshold of 25% along with confusion matrix.



```
[[441  92   0   0]
 [132 308   2   8]
 [  0   0 133   0]
 [  0  12   0 233]]
```

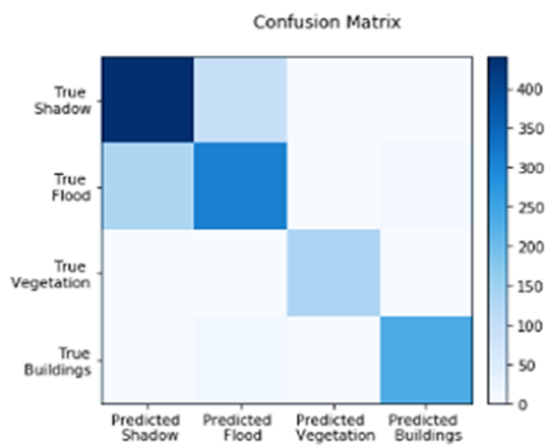


Figure 4.6: The Segmentation of Surface from the sea being classified as different regions where the blue color depicts the area potentially threated with flood with threshold of 50% along with confusion matrix.

In Figure 4.5 and figure 4.6 and Figure 4.7 This blue color depicts the area which can be flooded with water and blue color is used for the detection of flood water, everything in the figure which is blue is considered flooded, however the green color depicts the area that is forest and cultivated. The building that are in grey color are above the floor and cannot be flooded and you can clearly see the roofs of all buildings that are safe and can be clearly seen from above. While the purple color is the river in city or the sea at the bay of the city situated. The program code has a function, that only runs during the device initialization of intelligent greenhouse transformation. Thus, in this intelligent greenhouse function our device initializes and sets the initial values related with the components pins, the WiFi network properties, the greenhouse server address and the MQTT broker address. Then, it sends the registration messages to the greenhouse server, using WiFi, in order to register itself and all of its components and properties. Finally, the device connects to the MQTT broker and subscribes the topics related with its components. The function, as its name suggests, is a function that will be executed continuously until the board is switched off or reset, allowing our program to change and respond in each iteration. We have implemented in this function the MQTT protocol loop that checks, for each main loop iteration, if there are new MQTT messages from the subscribed to IoT-based MQTT broker for the implementation of greenhouse automation system. The system has, per default, one user role: An IoT-based MQTT broker has assigned the maximum permissions level: it allows to change all the devices, components and properties values; to create new load balancer in roles and assign them to specific part of intelligent greenhouse; to create new rooms and also to create new permissions with the help of MQTT broker with low error rate percentage.

5. DISCUSSION

The main goal of this project was to create a system that allows users to easily connect different IoT greenhouse appliances, regardless of their manufacturer, and control them remotely from any device with an internet connection. It was necessary to create a simple, more flexible protocol that would guarantee interoperability and security in the control and administration of this system. Our system offers an inexpensive solution, with low implementation costs and easy installation, that allows any user to deploy it in his greenhouse. Our solution uses the WiFi network as communication technology, since most of the users already have in their greenhouses and also allows to reuse existing devices by adding only a simple microcontroller, similar to an Arduino, that connects these devices to the internet. This is a complex system, although it is built in a modular fashion keeping decoupling between modules, allowing users to change the system settings whenever they want, adding or removing devices after the system is implemented and already operational, like a" plug-and-play" system. It is a system whose target audience is quite wide, including ordinary users without technical knowledge, that just want to buy the devices and incorporate them into their system, or Do-It-Yourself (DIY) Makers, who want to create their own IoT greenhouse appliances.

Despite being versatile, our system is robust and offers security mechanisms so that users can install it with confidence. These security mechanisms result from the designed architecture that is trans- parent to the devices within the network, and allows differentiating two types of networks: the external network, where users connect remotely to control the devices after being properly authenticated, and the internal network restricted to greenhouse devices communication and administration purposes.

The core component of our system is the smart greenhouse server that is a device with more processing power resources, such as a python based system or a MQTT-based system. This component contains all the system security and management mechanisms, such as the user permission management, that allows to create users and user roles with permissions assigned at different levels. It is possible to define permissions at different levels of hierarchy (room, devices or properties) which provides great flexibility in managing accesses to the devices.

Regarding the devices allowed, the requirements are minimal since the system management is done on a higher level, requiring that devices only manage their own components and are able to connect to an WiFi network. Our system also offers a feature that allows the creation of gateways to connect with simpler devices, that cannot connect to an WiFi network.

Because the system architecture was designed to be highly scalable, there is no limit regarding the quantity of devices that users can connect. It is designed according a service-oriented architecture with RESTful web-services, independent of each other, with a MQTT broker for internal communication, that allows lightweight and simple messages between devices. There is also an auxiliary database that stores all the system information, such as connected devices, their components and status, the structure of the house (rooms), registered users, allowed user roles and their permissions.

Table 5.1: The Accuracy Of Proposed Technique For Early Flood Detention In Comparison with existing literature.

ARTICLE	TECHNIQUE	ACCURACY
[54]	Support Vector Machine	91.73%
[55]	Random Forest Technique	93.00%
Proposed	IoT-based MQTT Protocol	97.36%

During our evaluation, we noticed that there is a small delay in the messages' delivery, which is related to the message propagation time over the network and also to the amount of components involved. However, this duration is short, and the more complex action (when a user changes a property value) takes 1.5 seconds on average to be processed by the system upon receipt, which is reasonable considering the constrained resources of IoT devices.

The architecture of our system includes a mechanism for managing users and their permissions. We proposed a solution, however, this was a mechanism that was not implemented and tested in our project, so we cannot demonstrate a concrete evaluation regarding the effectiveness and good functioning of our solution.

Our system provides security related with the external accesses, requiring customers to use secure communications protocols, HTTPS, and authentication with digital certificates. These security mechanisms are simple to be used by customers, however, although this functionality is contemplated in the architecture of our solution, we have not been able to implement and test it properly, so we have not been able to prove that it is adequate to provide the desirable security.

6. CONCLUSION

6.1 CONCLUSION

An IoT-based intelligent greenhouse architecture with flood detection monitoring is presented in this research, it was designed as a multilayer architecture with different components, developed to be easily deployed without requiring a complex infrastructure. The modular architecture guarantees a decoupling between the intelligent IoT based system for greenhouse monitoring with core system and the devices connected, providing flexibility for the users to change them and the intelligent greenhouse system configurations according to their needs, with the system already in operation. Even though our evaluation was performed using only one device with MQTT protocol connected with simulation being performed in the python programming, in a global balance evaluation, the system achieves all the proposed goals. It has a simple MQTT protocol that supports a network of several IoT devices connected and it has also important features to be used in real greenhouse scenarios, e.g. the IoT permissions mechanism. The WiFi network, as the core communication technology in our system, with the MQTT protocol for communication between devices was a good choice, because it reduces the minimal requirements for the devices and allows to connect devices with low computational flood detection resources to operate in our system. Also, we took advantage of some features of the MQTT protocol to guarantee security and interoperability in our system, such as the QoS levels that we use to specify the messages importance, where according to the defined levels, the messages could have delivery assurance with an accuracy of 97.36%. The retained message system, also reused from the MQTT protocol, allowed new devices to get information about the other devices' state immediately after they connect in our system. The decision to provide services for IoT devices in intelligent greenhouse to connect from the outside, through the Internet, was very important because it detaches the external network from the internal network, increasing the security and limiting the accesses to the devices.

6.2 FUTURE RECOMMANDATION

Our research can be the starter point to future research projects that have the goal to create a flexible and truly smart, greenhouse monitoring system with early flood detection. Thus, some ideas that will improve the whole system might be:

a) Add machine learning algorithms that are able to learn the users' preferences and make predictions about future actions or events.

b) Create scenarios, similar to what Turkey Indigenous Greenhouse Monitoring system after acquiring funding from research industries as innovation.

c) Create gateways to allow devices that are not able to connect to WiFi networks, to connect and use our system.

d) Allow to add more hierarchical levels inside an intelligent building, instead of just greenhouse and early flood detection in different cities of Turkey.

e) Allow to change the protocol syntax, similar to the Manifest idea in different IoT and machine learning techniques.

f) Create a user interface, responsive and adaptive to multiple screen sizes using machine learning and IOT based techniques.

REFERENCES

[1] Cho, J.Y.; Kim, K.-B.; Jabbar, H.; Sin Woo, J.; Ahn, J.H.; Hwang, W.S.; Jeong, S.Y.; Cheong, H.; Yoo, H.H.; Sung, T.H. Design of optimized cantilever form of a piezoelectric energy harvesting system for a wireless remote switch. Sens. Actuators A 2018, 280, 340–349.

[2] Ferrag, Mohamed Amine & Shu, Lei & Yang, Xing & Derhab, Abdelouahid & Maglaras, Leandros. (2020). Security and Privacy for Green IoT-Based Agriculture: Review, Blockchain Solutions, and Challenges. IEEE Access. PP. 10.1109/ACCESS.2020.2973178.

[3] Amaro, J.P.; Cortesão, R.; Ferreira, F.J.T.E.; Landeck, J. Device and operation mechanism for non-beacon IEEE802.15.4/Zigbee nodes running on harvested energy. Ad Hoc Networks 2015, 26, 50–68.

[4] Bianchi, V.; Ciampolini, P.; De Munari, I. RSSI-Based Indoor Localization and Identification for ZigBee Wireless Sensor Networks in Smart Greenhouses. IEEE Trans. Instrum. Meas. 2018, 1–10.

[5] Al-Sarawi, S.; Anbar, M.; Alieyan, K.; Alzubaidi, M. Internet of Things (IoT) communication protocols: Review. In Proceedings of the 2017 8th International Conference on Information Technology (ICIT), Amman, Jordan, 17–18 May 2017. 38. QCA4020|Qualcomm. Available online: https://www.qualcomm.com/products/qca4020

[6] Bassoli, M.; Bianchi, V.; De Munari, I.; Ciampolini, P. An IoT approach for an AAL Wi-Fi-based monitoring system. IEEE Trans. Instrum. Meas. 2017, 66, 3200–3209.

[7] Moraru, S.; Perniu, L.; Kristaly, D.M.; Ungureanu, D.A.; Sandu, F.; Mo, A.A. Proceedings of the 2017 25th Mediterranean Conference on Control and Automation (MED), Valletta, Malta, 3–6 July 2017.

[8] Kristaly, D.M.; Moraru, S.-A.; Stefan Petre, V.; Parvan, C.A.; Ungureanu, D.E.; Mosoi, A.A. A Solution for Mobile Computing in a Cloud Environment for Ambient Assisted Living. In Proceedings of the 2018 26th Mediterranean Conference on Control and Automation (MED), Zadar, Croatia, 19–22 June 2018.

[9] Lago, P.; Roncancio, C.; Jiménez-Guarín, C. Learning and managing context enriched behavior patterns in smart greenhouses. Futur. Gener. Comput. Syst. 2019, 91, 191–205.

[10] Bassoli, M.; Bianchi, V.; De Munari, I.; Ciampolini, P. An unobtrusive Wi-Fi system for human monitoring. In Proceedings of the IEEE International Conference on Consumer Electronics, Berlin, Germany, 3–6 September 2017.

[11] Boztas, A.; Riethoven, A.R.J.; Roeloffs, M. Smart TV forensics: Digital traces on televisions. Digit. Investig. 2015, 12, 72–80.

[12] Kang, S.; Kim, S.; Kim, J. Forensic analysis for iot fitness trackers and its application. Peer-to-Peer Netw. Appl. 2020, 13, 564–573

[13] Li, S.; Choo, K.-K.; Sun, Q.; Buchanan, W.J.; Cao, J. IoT forensics: Amazon Echo as a use case. IEEE Internet Things J. 2019, 6, 6487–6497.

[14] Becirovic, S.; Mrdovic, S. Manual IoT Forensics of a Samsung Gear S3 Frontier Smartwatch. In Proceedings of the 2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 19–21 September 2019; pp. 1–5.

[15] Dorai, G.; Houshmand, S.; Baggili, I. I know what you did last summer: Your smart greenhouse Internet of Things and your iPhone forensically ratting you out. In Proceedings of the 13th International Conference on Availability, Reliability and Security, Hamburg, Germany, 27–30 August 2018.

[16] Awasthi, A.; Read, H.O.; Xynos, K.; Sutherland, I. Welcome pwn: Almond smart greenhouse hub forensics. Digit. Investig. 2018, 26, 38–46.

[17] Chung, H.; Park, J.; Lee, S. Digital forensic approaches for Amazon Alexa ecosystem. Digit. Investig. 2017, 22, 15–25.

[18] Orr, D.A.; Sanchez, L. Alexa, did you get that? Determining the evidentiary value of data stored by the Amazon R Echo. Digit. Investig. 2018, 24, 72–78.

[19] Fensel, A.; Tomic, D.K.; Koller, A. Contributing to appliances' energy efficiency with Internet of Things, smart data and user engagement. Future Gener. Comput. Syst. 2017, 76, 329–338.

[20] Goudbeek, A.; Choo, K.K.; Le-Khac, N.A. A Forensic Investigation Framework for Smart Greenhouse Environment. In Proceedings of the 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), New York, NY, USA, 1–3 August 2018; pp. 1446–1451.

[21] Alharbi, R.; Allen, W.H. Collection and Analysis of Digital Forensic Data from Devices in the Internet of Things. In Proceedings of the 2019 SoutheastCon, Huntsville, AL, USA, 11–14 April 2019.

[22] Astillo, Philip Virgil & Kim, Jiyoon & Sharma, Vishal & You, Ilsun. (2020). SGF-MD: Behavior Rule Specification-Based Distributed Misbehavior Detection of Embedded IoT Devices in a Closed-Loop Smart Greenhouse Farming System. IEEE Access. 8. 10.1109/ACCESS.2020.3034096.

[23] Philomin, S.; Singh, A.; Ikuesan, A.; Venter, H. Digital Forensic Readiness Framework for Smart Greenhouses. In Proceedings of the International Conference on Cyber Warfare and Security, Islamabad, Pakistan, 29 September–1 October 2020; pp. 627–638.

[24] MacDermott, A.; Baker, T.; Shi, Q. Iot forensics: Challenges for the ioa era. In Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 26–28 February 2018; pp. 1–5.

[25] Zulkipli, N.H.; Alenezi, A.; Wills, G.B. IoT Forensic: Bridging the Challenges in Digital Forensic and the Internet of Things. In Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017), Porto, Portugal, 24–26 April 2017; pp. 315–324.

[26] Stoyanova, M.; Nikoloudakis, Y.; Panagiotakis, S.; Pallis, E.; Markakis, E.K. A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches and Open Issues. IEEE Commun. Surv. Tutor. 2020, 22, 1191–1221.

[27] Servida, F.; Casey, E. IoT forensic challenges and opportunities for digital traces. Digit. Investig. 2019, 28, 22–29.

[28] Kruger, J.-L.; Venter, H. Requirements for IoT Forensics. In Proceedings of the 2019 Conference on Next Generation Computing Applications (NextComp), Mauritius, 19–21 September 2019; pp. 1–7.

[29] Yaqoob, I.; Hashem, I.A.T.; Ahmed, A.; Kazmi, S.A.; Hong, C.S. Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges. Future Gener. Comput. Syst. 2019, 92, 265–275.

[30] Yang, W.; Johnstone, M.N.; Sikos, L.F.; Wang, S. Security and Forensics in the Internet of Things: Research Advances and Challenges. In Proceedings of the 2020 Workshop on

Emerging Technologies for Security in IoT (ETSecIoT), Sydney, Australia, 21–21 April 2020; pp. 12–17.

[31] Available Online: https://rkrknowledge.com/what-is-iot-how-it-works-internet-of-things/

[32] Ahmed, M.A.; Kang, Y.C.; Kim, Y.-C. Communication Network Architectures for Smart-House with Renewable Energy Resources. Energies 2015, 8, 8716–8735.

[33] Bradac, Z.; Kaczmarczyk, V.; Fiedler, P. Optimal Scheduling of Domestic Appliances via MILP. Energies 2015, 8, 217–232.

[34] Collotta, M.; Pau, G. A Novel Energy Management Approach for Smart Greenhouses Using Bluetooth Low Energy. IEEE J. Sel. Areas Commun. 2015, 33, 2988–2996.

[35] Rayhana, Rakiba & Xiao, Gaozhi & Liu, Zheng. (2020). Internet of Things Empowered Smart Greenhouse. IEEE Journal of Radio Frequency Identification. PP. 1-1. 10.1109/JRFID.2020.2984391.

[36] Collotta, M.; Pau, G. A Solution Based on Bluetooth Low Energy for Smart Greenhouse Energy Management. Energies 2015, 8, 11916–11938.

[37] Collotta, M.; Pau, G. An Innovative Approach for forecasting of Energy Requirements to improve a Smart Greenhouse Management System based on BLE. IEEE Trans. Green Commun. Netw. 2017.

[38] Subbiah, R.; Pal, A.; Nordberg, E.K.; Marathe, A.; Marathe, M.V. Energy Demand Model for Residential Sector: A First Principles Approach. IEEE Trans. Sustain. Energy 2017.

[39] Javaid, N.; Javaid, S.; Abdul, W.; Ahmed, I.; Almogren, A.; Alamri, A.; Niaz, I.A. A Hybrid Genetic Wind Driven Heuristic Optimization Algorithm for Demand Side Management in Smart Grid. Energies 2017, 10, 319.

[40] Available Online: http://blog.yuktix.com/2015/06/smart-greenhouse-roi-return-on.html

[41] Available Online: https://www.atophort.com/news/iot-horticulture-led-grow-lights-in-smart-greenhouse.html

[42] Ikpehai, A.; Adebisi, B.; Rabie, K.M.; Haggar, R.; Baker, M. Experimental Study of 6LoPLC for Greenhouse Energy Management Systems. Energies 2016, 9, 1046.

[43] Ahmed, M.S.; Mohamed, A.; Homod, R.Z.; Shareef, H. Hybrid LSA-ANN Based Greenhouse Energy Management Scheduling Controller for Residential Demand Response Strategy. Energies 2016, 9, 716.

[44] Nguyen, T.T.K.; Shimada, K.; Ochi, Y.; Matsumoto, T.; Matsugi, H.; Awata, T. An Experimental Study of the Impact of Dynamic Electricity Pricing on Consumer Behavior: An Analysis for a Remote Island in Japan. Energies 2016, 9, 1093.

[45] Moon, J.W.; Chung, M.H.; Song, H.; Lee, S.-Y. Performance of a Predictive Model for Calculating Ascent Time to a Target Temperature. Energies 2016, 9, 1090.

[46] Pitì, A.; Verticale, G.; Rottondi, C.; Capone, A.; Lo Schiavo, L. The Role of Smart Meters in Enabling Real-Time Energy Services for Households: The Italian Case. Energies 2017, 10, 199

[47] Niaz, M.T.; Imdad, F.; Kim, H.S. Power Consumption Efficiency Evaluation of Multi-User Full-Duplex Visible Light Communication Systems for Smart Greenhouse Technologies. Energies 2017, 10, 254.

[48] Alamin, Y.I.; Castilla, M.M.; Álvarez, J.D.; Ruano, A. An Economic Model-Based Predictive Control to Manage the Users' Thermal Comfort in a Building. Energies 2017, 10, 321.

[49] Luthander, R.; Widén, J.; Nilsson, D.; Palm, J. Photovoltaic self-consumption in buildings: A review. Appl. Energy 2015, 142, 80–94.

[50] Pickering, B.; Choudhary, R. District energy system optimisation under uncertain demand: Handling data-driven stochastic profiles. Appl. Energy 2019, 236, 1138–1157.

[51] Onishi, V.C.; Antunes, C.H.; Fraga, E.S.; Cabezas, H. Stochastic optimization of trigeneration systems for decision-making under long-term uncertainty in energy demands and prices. Energy 2019, 175, 781–797.

[52] Afzali, S.F.; Cotton, J.S.; Mahalec, V. Urban community energy systems design under uncertainty for specified levels of carbon dioxide emissions. Appl. Energy 2020, 259, 114084.

[53] Zhang, B.; Hu, W.; Cao, D.; Huang, Q.; Chen, Z.; Blaabjerg, F. Deep reinforcement learning–based approach for optimizing energy conversion in integrated electrical and heating system with renewable energy. Energy Convers. Manag. 2019, 202, 112199.

[54] Islam, Kazi Aminul & Uddin, Mohammad Shahab & Kwan, Chiman & Li, Jiang. (2020). Flood Detection Using Multi-Modal and Multi-Temporal Images: A Comparative Study. Remote Sensing. 12. 2455. 10.3390/rs12152455

[55] Lamovec, Peter & Veljanovski, Tatjana & Matjaž, Mikoš & Oštir, Krištof. (2013). Detecting flooded areas with machine learning techniques: Case study of the Selška Sora river

flash flood in September 2007. Journal of Applied Remote Sensing. 7. 073564. 10.1117/1.JRS.7.073564