EAST WEST UNIVERSITY

PANI

An App for Dynamic Irrigation Scheduling

By

Md. Turza Ibna Muksud

ID: 2018-1-96-003

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science

2020

PANI - an App for Dynamic Irrigation Scheduling

Md. Turza Ibna Muksud

ID: 2018-1-96-003

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science

Department of Computer Science & Engineering

East West University
Dhaka-1212, Bangladesh

February, 2020

Declaration

I hereby declare that I have completed research on the topic entitled "PANI - an App for Dynamic Irrigation Scheduling" as well as prepared the research report under the supervision of Dr. Ahmed Wasif Reza, Associate Professor, Department of Computer Science Engineering. This report is submitted to the department of Computer Science Engineering, East West University in partial fulfillment of the requirement for the degree of MS in CSE, under the course "MS Thesis (CSE 599)".

I further assert that this report in question is based on my original exertion having never been produced fully and/or partially anywhere for any requirement.

Countersigned

…....................................

Dr. Ahmed Wasif Reza
Associate Professor,
Department of Computer Science &Engineering

Signature

..…..

Md. Turza Ibna Muksud
ID: 2018-1-96-003

Letter of Transmittal

February 13, 2020

Dr. Ahmed Wasif Reza

Associate Professor

Department of Computer Science & Engineering

East West University

Subject: Submission of Research Report

Dear Sir,

I am here by submitting my Research Paper, which is a part of MS in CSE under the Department of Computer Science & Engineering. It is a great achievement to work under your supervision. This report is based on "PANI - an App for Dynamic Irrigation Scheduling".

This project gave me both academic and practical exposures. In this project I have learned about Internet of Things (IoT), app development, Support Vector Machine (SVM), Artificial Intelligence (AI) etc.

I shall be highly obliged to you if you are kind enough to receive this report as well as provide your valuable judgment. It would be my immense pleasure if you find this report useful and informative to have an apparent perspective on this issue.

Sincerely Yours,

.…..

Md. Turza Ibna Muksud

ID: 2018-1-96-003

Letter of Acceptance

This research report "PANI - an App for Dynamic Irrigation Scheduling" is the outcome of the original work carried out by Md. Turza Ibna Muksud, ID: 2018-1-96-003, under my supervision to the Department of Computer Science & Engineering, East West University, Dhaka – 1212.

Supervisor

…..................................

Dr. Ahmed Wasif Reza

Associate Professor,

Department of Computer Science & Engineering,

East West University

Chairperson

…..................................

Dr. Taskeed Jabid

Associate Professor and Chairperson,

Department of Computer Science & Engineering,

East West University

Acknowledgement

First and foremost, with all my heartiest devotion I am grateful to almighty Allah for blessing me with such opportunity of learning and ability to successfully complete the research.

A special thanks with the honor to my respected supervisor Dr. Ahmed Wasif Reza who was kind enough to allocate his valuable time to provide me with his humble guidance, motivating thought and encouragement. And last but not least, I like to thanks my family, friends and everyone for their constant supports and helps. I will remember them in my heart and hope to find a more appropriate place to acknowledge them in the future.

Abstract

Large tracts of land in the delta region of Bangladesh are left fallow or are cultivated with low input crops during the dry winter months. Surface water based irrigation opens up new opportunities for sustainable intensification by enabling the production of a high yielding crop. Irrigation in Bangladesh is still managed in a very traditional manner without considering scientific data on soil and weather which increase cost of farming and also lower productivity. Farmers usually depends on their past experience. In a traditional way, they forecast the weather as well as the moisture of the land. So the process the irrigation depends more on their guess.

The simulation model has been integrated into a smart phone app called PANI (Program for Advanced Numerical Irrigation). PANI runs on a daily time step and uses forecasted weather data to predict irrigation needs one week in advance. I used the calibration of the Blainy-Criddle Equation so that only forecasted daily maximum and minimum temperatures are required. PANI addresses the needs of the irrigation service provider as well as of the farmer. Both will receive the field data, informing them as to whether a field needs to be irrigated or not. Since the water balance of PANI is based on the SVM algorithm, it can be used under most conditions, as long as weather and ground cover data are available.

Table of Contents

List of Figures

List of Abbreviations

IoT	-	Internet of Things
AI	-	Artificial Intelligence
BMD	-	Bangladesh Meteorological Department
/SVM	-	Support Vector Machine
API	-	Application Program Interface
GIS		Geographic Information Systems
GPS	-	Global Positioning System
I/O	-	Input and Output
IR	-	Inferred Ray
NIR	-	Near-infrared spectroscopy
MSI	-	Multispectral Instrument
GEO	-	Gene Expression Omnibus

Chapter 1

Introduction

A rising middle class has fueled demand for high quality agriculture products. Most agricultural production in Bangladesh is characterized by traditional subsistence farming. Bangladesh produces a variety of agricultural products such as rice, wheat, corn, legumes, fruits, vegetables, chicken meat, fish, and seafood.

Bangladesh agriculture has changed dramatically, especially since the end of Independent War. Crops productivity soared due to new technologies, mechanization, increased chemical use, specialization and government policies that favored maximizing production. These changes allowed fewer farmers with reduced labor demands to produce the majority of the crops in Bangladesh. Although these changes have many positive effects and reduced many risks in farming, there have also been significant increased costs viz. top soil depletion, ground and surface water contamination, continued neglect of the living and working conditions for farm labors, increasing costs of production, and the disintegration of economic and social conditions in rural communities.

A growing movement has emerged during the past two decades to question the role of the agricultural establishment in promoting practices that contribute to these social problems. Now a day the movement for sustainable agriculture is gaining increasing support and acceptance within mainstream agriculture. Not only sustainable agriculture address many environmental and social concerns, but it offers innovative and economically viable opportunities for growers, laborers, consumers, policy makers and many others in the entire food system with concerning congenial environment.

Irrigation is the artificial application of water for the success of crop production in the field. In the field of irrigation has reached a rapid development of mechanization. In modern times, irrigation efficiency has become important because of groundwater depletion. Therefore, adequate planning for irrigation is required. In Bangladesh, the flood irrigation method was selected mainly for its simplicity. To improve irrigation efficiency, an intelligent irrigation system has been introduced. In recent days, smart irrigation is the subject of popular discussion for researchers.

The irrigation system is the smart climate monitoring system, soil conditions, evaporation, using plant water and automatic irrigation program. Intelligent irrigation systems beautify watering schedules and automatically running times to meet the specific needs of the landscape. The controllers significantly improve the efficiency of outdoor water use. There are several options for smart irrigation controllers, such as climate-based soil moisture sensors (ETs) and on the site. The right solution depends on the geographical solution and the landscape environment.

Forecast evaporation data use four meteorological parameters such as temperature, wind, solar radiation and humidity. It is the most accurate way to calculate the climatic needs of the landscape. There are three basic forms of these ET drivers based on time. The user's signal-based controller uses weather data from an available audience. The source and value of ET are calculated for a grassy area at the site. The ET data is sent to the controller via a wireless connection. The difference is that the system will interrupt the next irrigation program when there is enough moisture in the ground. The automatic system of irrigation of plants or intelligent irrigation systems is one of the best examples of electronic technology in the field of agriculture that makes the best use of water in plants.

In Bangladesh, agricultural land decreases day by day due to population growth. That's why we need to increase food production every year, but our amount of land is not enough for this purpose. Smart technology is the best solution for this problem. By using intelligent technology, it is possible to increase the efficiency of each irrigation site and save the economy.

With the proposed model, various questions have raised to promote connectivity in between automation system and productivity of harvesting products. Regarding research goal the raised questions are mentioned below.

Q1 To manage the automation system in terms of run-time and design time efficiency, what should be the procedure?

Q2 How can standardized engineering methodologies for capturing process and technical knowledge be utilized to manage a cross-domain IoT automation system of systems?

Q3 What are the functional, performance and operational critical aspects of a successful migration of an industrial automation system to an IoT-based cloud solution?

1.1 OVERVIEW AND MOTIVATION

IoT is an overarching term describing networks of connected devices that are deployed in many different sectors and applications. While there are various ways in which these applications can be characterized, it is useful for the purposes of policymaking to separate them into industrial, public space and consumer applications. While these three categories are not entirely distinct, each category has broadly different communities of stakeholders and differing public expectations, legal contexts and governance arrangements. Each category also has varied desired outcomes upon which policymaking should focus. Policy for industrial applications should focus on realizing tangible productivity and efficiency gains, plus developing the secure provision of services through connected products or assets. Consumer IoT policy should focus on customer benefits such as reduced utility costs and improved quality of life, while proposing ways to reconcile the demands of security, privacy, cost and ease of use. Policy outcomes for public space applications share common elements with industrial and consumer applications for example, outcomes might include both public sector efficiencies and customer benefits and will need to balance the benefits between the various stakeholders. Focusing on these outcomes will help to ensure clarity in a complex landscape.

IoT and Big Data are interconnected with each other. IoT is going to generate huge amounts of data that must be analyzed if the IoT networks are going to operate accurately. The networks may generate some redundant data and that is why it becomes important for Big Data organizations to spend their analytics power on the data that is important. So, a new element of data categorization will be added so that the Big Data Analytics tools deliver better performance.

IoT devices are going to send in truckloads of data for analysis to Big Data organizations. Right now, Big Data companies are only just becoming capable of handling this immense amount of data in a highly secure manner. The change we are expecting on the Big Data front would be the adoption of flexible and scalable solutions to enhance security, data storing, and data analysis capabilities.

IoT is the new "thing" of this era and its wide adoption is sending signals to Big Data organizations to be ready to handle data of varied types coming in from different types of devices. IoT has burst onto the stage, interconnecting everyday objects over the Internet, which acts as everlasting sources of information. The occurrence has required a combination of three developments.

1.2 PROBLEM STATEMENT

The IoT allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems. Thanks to the huge addressing capabilities of IPv6, each thing is uniquely identifiable and is able to interoperate within the existing Internet infrastructure. Experts estimate that the IoT will consist of almost 50 billion objects by 2020.

So, in order to develop a smart system for better use of resources and implementation following factors need to be considered.

- ❖ Even though the automation using IoT is happening in all the sectors including Home, Business, Industry and agriculture, there is still need of predictive analysis to prevent some unexpected problems from before it happens.
- ❖ Google and Accurate weather is giving us weather forecasting using IoT and AI which is getting mature day by day. Now in many places in the world it is getting much more mature.
- ❖ IoT in agriculture can help us in to make our life easy but it cannot always adjust our expectation where AI can.

1.3 THESIS OBJECTIVES

In this paper, I have proposed a smart farming system that monitor weather as well the soil entity and suggest proper initiatives and implement them through SVM, , Mobile app also by analyzing the available data. The proposed system can optimize the usage of water and can play a key role to produce good productivity by introducing IoT in our daily life..

In the process of developing our proposed system I have set the following objectives:

- To design smart system by using low cost microcontroller, sensors and actuators
- To integrate existing devices into the system to improve efficiency
- To implement the system and analyze the resulted data

In this thesis paper, our aim is to make a prototype of a smart farming system which can be easily implemented by anyone or anywhere so that it can provide support for personal and industrial usage. The implementation of this system can change the view of traditional farming mechanism. In this paper, the concept of smart agriculture using IoT is briefly discussed. This paper gives an implementation of the proposed system. Wi-Fi sensor is used to collect real time data devices and other environmental API's. At last the paper discusses the result analysis of collected data through various sensors using decision tree classification algorithm.

1.4 THESIS ORGANIZATION

I have organized our rest of the paper as follows. In chapter 2, I have discussed about the PANI method and its possibility in future. In this section, I have also discussed about existing related works. In chapter 3, I have proposed our methodology. I discussed materials, sensors and devices that I used to develop the PANI system. I also discussed flow chart and algorithm I used in our proposed system. In this section, I have also discussed about experimental setup. In chapter 4, I have discussed about data collection, result analysis and discussion. Finally chapter 5 gives a conclusion of this paper.

Chapter 2

Literature Review

The goal of this section is to gather and include the knowledge required to understand the methodology of this work in an organized way. In that way, the problem we are working with can be understood as well as attempted to solve better.

2.1 HISTORICAL DEVELOPMENT OF AGRICULTURE IN BANGLADESH

Bangladesh agriculture has achieved significant structural changes over the past three and half decades. Despite many problems and constraints a quiet agricultural revolution has taken place which is still evolving in response to natural calamities, sociopolitical changes, population growth, urbanization, new technology in agriculture and new opportunities in rural nonagricultural sector, commercialization and changes in macro policy and sector policy reforms including market and trade liberalization and substantial reduction in public sector intervention in agriculture. From largely a peasant based subsistence activity, agriculture today is more of a commercial entrepreneurial activity than ever before. Cereal grain production increased three folds. Growth of non-cereals, mostly horticultural products accelerated. The great observation of the increasing phenomenon of diversification and intensification of Bangladesh agriculture. Technological change coupled with market forces greatly influenced Bangladesh agriculture in midsixties and seventies. From a relatively stagnant sector in the pre-green revolution period, Bangladesh agriculture emerged as a dynamic sector in the green revolution period. There was a significant growth in agriculture infrastructure as well as a shift towards liberalization from government control.

Long-term trend of rice production shows that the dominant factor in growth is rice yield, stimulated by high yielding varieties (HYV), fertilizer, and irrigation technology. Investment in research was small, yet there was technological progress, as evidenced by release of 58 HYV rice varieties by BRRI/BINA/Universities and 473 high yielding varieties of various crops by different research institutes. Extension too played a role although the much discussed, the lesson being that market should spread new technology, farmers should seek out extension with queries and problems and agents should be there to assist. Chemical fertilizer and pesticide use became popular across all groups of farmers form 1971 **[1]**. Integration of variety-fertilizer-irrigation development was a major factor behind spread of HYV and consequent yield increase. Other concurrent developments include increasing mechanization of agricultural operations, including power tillers for cultivation, pumps for irrigation and threshers and rice mills for converting paddy into rice. These technologies are now concern to use them an integrated manner get sustainability in agriculture.

"Sustainable agriculture means an integrated system of plant and animal production practices having a site-specific application that over the long term will: Satisfy human food and fiber need, enhance environmental quality and the natural resource base upon which the agricultural economy depends, make the most efficient use of non-renewable resources and on farm resources and integrate, where appropriate, natural biology cycles and controls, sustain the economic viability of farm operations, enhance the quality of life for farmers and society as a whole" **[2]**.

"Sustainable agriculture is one that produces abundant food with depleting the earth's resources or polluting its environment. It is agriculture that follows the principles of nature to form systems for raining crops and livestock that are, like nature, self-sustaining. Sustainable agriculture is also the agriculture of social values, one whose success is indistinguishable from vibrant rural communities, rich lives for families on the farm, and wholesome a food for everyone." **[1]**.

2.2 APPLICATION OF SVM

SVM has various applications, and many of these applications are in the field of precision agriculture. Precision agriculture refers to agricultural practices that aim at maximizing yield, minimizing inputs and/or improving the quality of the agricultural products. What is specific for precision agricultural is the use of technologies such as the global positioning system (GPS), geographic information systems (GIS), various sensors, remote sensing. Remotely sensed images have a variety of applications in precision agriculture – they are used for mapping soil moisture content, determining the phenological state of crops, monitoring crop growth, nutrient deficiency and evapotranspiration rate, identifying crop diseases, assessing weed and insect infestation. Based on remotely sensed images an estimate of the radiation reflected by the crops can be made which can indirectly give us insights into different bio-physical properties of the vegetation such as chlorophyll and nitrogen content, biomass, etc. Reflectance can also be used for calculating various vegetation indices which do not only relate to the bio-physical properties of crops but serve as inputs in agroecosystem models too.

One of the applications of SVM in agriculture is its use for the estimation of soil water evaporation. Soil water evaporation is negatively correlated to SVM because the lower the SVM the higher the evaporation. SVM is essential for the calculation of the crop coefficient used in the methodology for estimating crop water requirements of the Food and Agriculture Organization of the United Nations. In this methodology, SVM is originally referred to as ground cover. SVM has also been used in models for soil moisture estimation.

SVM can be calculated for post-harvest agricultural fields to quantify the amount of crop residues. The tillage performed in agricultural fields accelerates soil erosion and decreases the content of carbon in soil. Crop residues on agricultural land can prevent these processes, thus crop residue fraction cover is an important measure in conservation tillage systems. SVM has been widely used in soil erosion models not only for agricultural fields but for diverse land cover types.

Despite the wide use of SVM, the multiple terms being used to describe it are not yet standardized which makes the comparison of studies using this bio-physical parameter difficult. In literature, SVM can be often regarded to as (fractional/fraction of/percentage of) (green) foliage, ground, plant, vegetation or canopy cover. Depending on the purpose for which it is used, cover may refer only to certain types of vegetation such as tree and shrub canopies, only to green vegetation, to all types of vegetation including litter or in the case of ground cover also to non-vegetation material such as rocks. SVM can be expressed in different ways - as percentage, proportion or using a categorical scale. In this thesis, SVM is expressed as percentage and refers only to the photosynthetically active green parts of a plant.

2.3 METHODS FOR ESTIMATION SVM

Various methods exist for estimating SVM. Some of them rely on visual interpretation while other use remote sensing observations. In this thesis, the emphasis is put on methods using the latter approach.

Remote sensing observations do not directly provide insights into the biophysical or biochemical characteristics of the surface being sensed, they only quantify the radiation field reflected or emitted by it. An intermediate step is required to convert the measured radiation into vegetation products **[21]**. The methods used to perform this step can broadly be classified into three groups – physical, statistical, and hybrid.

i. **Statistical methods:**

Statistical methods rely mainly on vegetation indices. VIs are used for describing the relationship between the reflectance measured by a sensor and a parameter of interest. Ideally, they have high sensitivity to the parameter of interest and are insensitive to perturbing factors such as soil color change and atmospheric effects. Various VIs have been used to estimate SVM. While NDVI is the most popular of, it is not always the best choice. For example, NDVI is suitable for estimating SVM in savannas, pecan orchards and arid and semi-arid areas but for tropical landscapes the reduced simple ratio index is better suited.

Some VIs are based only on the red, green and blue spectral bands. These color indices have the advantage that they are more intuitive to humans as human eyes can actually “see” these colors. Color indices such as the Excess Green Index and the Normalized Difference Index are particularly suitable for estimating SVM because they allow to easily differentiate between plants and other background by using an appropriate threshold value. Color indices are less sensitive than NIR indices to the lighting conditions and can be adjusted to different backgrounds. Although vegetation is well distinguished in the NIR band due to the strong reflection in the latter, color indices still offer an alternative way to estimate SVM, when information in the NIR band is missing.

Statistical methods based on VIs are computationally efficient and suitable for application at a regional scale. However, as they are specific to the vegetation type, they are not always suitable for application at a larger scale.

The soil line concept is also a statistical method used to estimate bio-physical variables. Soil pixels plotted in the red and near infra-red spectral space are positively correlated and form a line referred to as soil line. When plotted together with vegetation, the soil line represents the base of a triangle with the points above it showing vegetation. The soil line depends on variety of factors such as soil type, moisture content, organic matter content and other. Thus, it is not possible to retrieve global soil line but it has to be calculated for each particular case. Another method that can be associated with the group of statistical methods is spectral unmixing. Spectral unmixing is a technique where the spectral reflectance of a scene is presented as a weighted sum of the reflectance spectra of its components. At least two components are needed – for example, plant and soil. But it is possible to include more endmembers – soil, green vegetation and senescent vegetation or plant, shaded soil and illuminated soil.

ii. Physical and hybrid models:

Physical models are based mainly on inverting canopy reflectance models but soil and snow reflectance models can be used as well. These models simulate the radiation field emitted or reflected by the surface for a given type of vegetation. To solve the opposite problem – i.e. to retrieve the biophysical variables from the remote sensing measurements of the radiation field, the model should be inverted. Traditional inversion methods rely on repeated model runs until the total deviation between actual data and model predictions is minimized. Whether the inversion is successful heavily depends on the initial values provided to the reflectance model. If these values differ substantially from the "true" values, iterating the model may not lead to convergence. Thus, it is often necessary to provide several sets with initial values before the best solution can be selected increasing even more the computational time needed.

The main advantage of physical models is that training datasets simulated by inverse models cover more possible combinations of conditions than training datasets based on empirical data. But they also have drawbacks – radiative models have model uncertainties due to the simplifications of the canopy structure and uncertainties originating from the radiometric measurements; the distribution of the variables in the radiative models is poorly known. These uncertainties make the inversion problem ill-posed – a set of different solutions can correspond to similar reflectance values, and the use of prior information is often required. However, as such information is not always available other approaches based on contextual and temporal information are gaining popularity. Physically based models are also computationally expensive and therefore not suitable for application on regional and global scale. This limitation can be overcome by employing a generic algorithm which mimics the natural selection during evolutionary processes when selecting suitable set of initial variables for the reflectance model. Another alternative are look-up tables which store precomputed radiative transfer functions and thus allow for fast and accurate radiometric processing. Hybrid methods are also computationally more efficient than physical models. In hybrid methods, a radiative transfer model is used for the creation of a training dataset and a non-parametric statistical model is applied for describing the relationship between the reflectance and the bio-physical variable. Machine learning techniques such as artificial neural networks are often used.

2.4 OTHER METHODS

Machine learning techniques can be applied not only in the context of radiative transfer models but also directly for retrieving bio-physical variables from reflectance. The estimation of SVM can be considered as a classification problem with two classes – green vegetation and background, or as a regression problem where SVM is a dependent variable. Machine learning techniques are well suited for addressing both types of problems. In machine learning, it can be differentiated between various learning types such as rote learning, learning from instruction, learning by analogy and learning from example. Learning from example encompasses supervised and unsupervised learning each of them trying to find a general rule, which can explain the data based on a sample of it. In supervised learning, a training sample is available, in which both the dependent and independent variables are known (input-output pairs). Based on the training sample, a function is derived which can map an input to an output minimizing the error. Depending on the type of the output, supervised learning can be approached as regression or classification problem. In a classification problem, the output is a discrete class or category, while in a regression problem it is a value of a continuous variable. Although machine learning algorithms are well suited to solve this type of problems, they are often considered to be black-box models.

Many of the methods mentioned so far and in particularly classification can be applied both on pixel and object level. When the estimation of bio-physical variables is carried out on a pixel level, information such as texture, context and shape is often neglected. Object based image analysis (GEOBIA) techniques make use of this information by exploring the characteristics of features in the images and the relationships between them. Such techniques have already been used for the estimation of various vegetation parameters, SVM being one of them. In GEOBIA, it is argued that pixels are not always the best spatial unit when the mapping of landscape elements is concerned. In a typical GEOBIA workflow, an image is first segmented and then classified. During the segmentation step, similar pixels are grouped together into objects or "objects candidates" – homogenous and semantically meaningful groups of pixels.

Segmentation techniques have been widely used in remote sensing long before GEOBIA emerged as a paradigm. They can be divided generally into four groups – point-based, edge-based, region-based, and combined, using the discontinuity or similarity of the pixel's grey level values. In the context of image analysis, point-based approaches use the homogeneity between pixels and by applying a threshold divide the image in two or more categories. They are not particularly suitable for remotely sensed data because the reflectance values of an object may vary depending on the location. In edge-based approaches, the segments are defined by their outlines. In region-based approaches, the similarity between the pixels or already-existing regions is compared resulting in splitting or merging existing segments.

The image objects created during the segmentation step may correspond to real world objects such as individual buildings and tree crowns but this is not always possible often due to the mismatch between the spatial resolution of the image and the size of the object of interest. Image segmentation can be performed on multiple levels and scales. The scale depends on the object of the interest which can heavily range in size - from vegetation patches to more generalized classes such as forest and agricultural area. At each scale different features might be revealed. After the image has been segmented, it can be further processed by considering additional object properties such as size and shape allowing the interpretation and classification of the image or in other words the derivation of "meaningful objects".

2.5 INSTRUMENTS AND PLATFORMS FOR ESTIMATING SVM

SVM estimates can be retrieved using various instruments and platforms ranging from low cost, consumer grade cameras to professional spectrometers and from unmanned aerial vehicles to satellites.

i. **Digital cameras and smartphones**

Digital cameras are often used to take in-situ ground pictures usually only in the visible bands of the spectrum. Both hemispherical and nadir looking photographs are suitable as they all provide similar SVM estimates. For taking nadir looking photographs simple consumer grade cameras can be used. Hemispherical photographs require fish eye lenses.

Smartphones can also be used for taking high quality photographs. They are useful for vegetation properties estimation not only because of their high resolution camera but also because of other embedded sensors collecting information about the position, motion and surrounding environment. Smartphone photos have been used in a variety of agricultural applications such as disease detection and diagnosis, soil study, determining ripeness of fruits, fertilizer application and irrigation. They are not only compact, accessible and computationally efficient but in combination with dedicated applications can also enable the direct estimation of bio-physical variables and even provide an all-round farm management information system. Vegetation properties such as the leaf area index (LAI) and chlorophyll content derived from smartphone photos are comparable to those estimated with specialized instruments. Smartphones are a promising tool that may replace traditional instruments for field measurements. This is possible both due to the high resolution camera that many of them have and to other built-in sensors such as gyroscopes and inclinometers which allow the measurement of the angular velocity and the angles of slope. However, more research is needed to evaluate the impact of the smartphone model and the type of vegetation when smartphone photos are used for the estimation of vegetation parameters. Another concern is the location accuracy provided by smartphone devices. Many of the agricultural activities in which smartphones are involved are location based. However, the actual locational accuracy of the smartphone devices has to be carefully tested before they can be used for making any agricultural decisions.

ii. Satellites and their instruments

Satellite and areal images are replacing costly field surveys for collecting crop status information. The increasing spatial and temporal resolution of space-borne and airborne remote sensors enables the global monitoring of various crop characteristics in a timely manner **[8, 9]**. Global estimate of SVM is already provided by Copernicus Global Land Service at 1 km resolution based on images from the Proba-V satellite which is equipped with the Vegetation imager instrument. Multispectral images with near a meter ground sampling distance are available upon payment from commercial satellite images providers such as Digital Globe. However, when validating remotely sensed data, field surveys are inevitable.

2.6 THE IMPORTANCE OF SCALE

The instruments and platforms listed in the previous sections allow the estimation of vegetation properties at various scales. Scale refers not only to the spatial resolution but also to the spatial extent and between the two of them there is often a trade-off when the extent increases, the resolution usually decreases.

It is often assumed that better spatial resolution contributes to the more accurate estimation of bio-physical variables. However, multiple factors have to be taken into account when choosing the best spatial resolution. Their method assesses the spatial structure of images by using local variance which is measured by the standard deviation averaged over a moving window. If the size of the pixel is similar or smaller to that of the object of interest, the variance is high. If on the other hand the object of interest is smaller than the pixel size, the variance is low. Thus, a rapid change in variance can show when the optimal scale is reached.

2.7 THE IMPORTANCE OF FIELD SAMPLING STRATEGY

Comparing vegetation properties across different scales resembles the problem of validating vegetation products derived from satellite imagery with field measurements. In each of the cases, a sampling strategy has to be proposed as the extents of the study area at the different scales usually do not overlap. However, when designing a sampling strategy, the spatial resolution of the final product has to be taken into account.

In this section of the paper, I analyzed some previous works done by other researchers and showed our results in a comparative study table.

2.8 AVAILABLE RESEARCH

A regular and periodic monitoring of crop health is essential in any cultivation. An important parameter which act as indices of crop health is the leaf chlorophyll measurement. In the Asian part of the world, Betel vine (Piper betle L., family Piperaceae) ranks second to coffee and tea in terms of daily consumption. Therefore, these important and highly productive crash crop is selected for the purpose of study. The experiment was conducted in an established pan betel vine crop field (Pan boroj). A small review of the popular method of leaf chlorophyll measurement is done and some of the drawbacks of the existing methods are reported. The review point out a need for fast and precise leaf chlorophyll measurement technique. Thus an image processing technique based on trichromatic colors i.e., red green and blue (RGB) model is proposed. For the purpose of analysis of the proposed model, the model outcome was compared with atLEAF+ chlorophyll meter reading. And a regression analysis was performed the result of regression analysis proof that there is a strong correlation between proposed image processing technique and chlorophyll meter reading. Thus, it appears that the proposed image processing technique of leaf chlorophyll measurement will be a good alternative for measuring leaf chlorophyll rapidly and with ease [9].

Chapter 3

Related Works

3.1 SVM ALGORITHM

At present a fast iterative algorithm for identifying the support vectors of a given set of points. This algorithm works by maintaining a candidate support vector set. It uses a greedy approach to pick points for inclusion in the candidate set. When the addition of a point to the candidate set is blocked because of other points already present in the set, we use a backtracking approach to prune away such points. To speed up convergence we initialize our algorithm with the nearest pair of points from opposite classes. We then use an optimization based approach to increase or prune the candidate support vector set. The algorithm makes repeated passes over the data to satisfy the KKT constraints.

Good generalization performance of the proposed parameter selection is demonstrated empirically using several low- and high-dimensional regression problems. Further, we point out the importance of Vapnik's ε-insensitive loss for regression problems with finite samples. To this end, we compare generalization performance of SVM regression (using proposed selection of ε-values) with regression using 'least-modulus' loss (ε=0) and standard squared loss. These comparisons indicate superior generalization performance of SVM regression under sparse sample settings, for various types of additive noise **[11, 13]**.

3.2 GROUND COVER APP

Large tracts of land in the delta region of Bangladesh are left fallow or are cultivated with low input crops during the dry winter months. Surface water based irrigation opens up new opportunities for sustainable intensification by enabling the production of a high yielding crop. To ensure an optimal use of irrigation pumps, CIMMYT (International Maize and Wheat Improvement Center) is currently developing an advisory system for irrigation scheduling, which calculates a daily soil water balance and estimates crop water use based on ground cover derived from satellite data. The project aims to replace satellite data by RGB photos taken with smartphones as these do not depend on weather conditions and are accessible to famers **[16]**.

3.3 TEMPERATURE & HUMIDITY MEASURE AND CONTROL SYSTEM

New digital temperature humidity sensor DHT11 has many advantages such as compactness, simple interface, fast response, and cheapness. This paper briefly introduces the concept of 1-wire bus, and expounds the basic principles and the application methods of DHT11, as well as its application in the temperature and humidity control system. The excellent advantages makes DHT11 valued in the fields of automation and consumer electronics products.

3.4 CLOUD BASED AUTOMATION

Recent technological and commercial developments make cloud computing an affordable, scalable, and highly-available platform technology. Meanwhile, precision agriculture is showing its potentials by improving agricultural operations through better data-driven decision making. Nevertheless, further development of precision agriculture requires better technology and tools to process data efficiently at a reasonable cost, and to translate the data to better decisions and actions in afield. We developed a framework for cloud-based Decision Support and Automation systems that can acquire data from various sources, synthesize application-specific decisions, and control field devices from the Cloud. A distinctive feature of our framework is its extensible software architecture: decision modules can be added and/or configured for a specific operation. The platform features a device-agnostic frontend that can process incoming data in different formats and semantics. Finally, the platform incorporates software-defined control, a new software design paradigm we proposed to enable versatile and safe control of field devices from a cloud computing platform. An early version of the system has been developed and tested with support from the USDA.

3.5 DISCUSSION

From the literature review some valuable and important conclusions can be drawn. First of all, there is not much research done for remote servers that are included in closed control loops. So far, cloud infrastructures are used for monitoring industrial machines, improving business intelligence and reporting. A growing movement has emerged during the past two decades to question the role of the agricultural establishment in promoting practices that contribute to these social problems. Now a day the movement for sustainable agriculture is gaining increasing support and acceptance within mainstream agriculture. Not only sustainable agriculture address many environmental and social concerns, but it offers innovative and economically viable opportunities for growers, laborers, consumers, policy makers and many others in the entire food system with concerning congenial environment. These technologies are not mature yet, and a clear separation line cannot be drawn between them. The local server schemes present great potential for the applications that require real-time behavior and fast responses. The amount of IoT devices is going to rise significantly in the foreseen future, bring more devices connected to the world and increase the intelligence of every day appliances. IoT is, according to researches, an important part of the 4th industrial revolution, known as Industry 4.0.

Chapter 4

Research Methodology

4.1 METHODOLOGY

In this section I will discuss the research techniques used to conduct this research work. This research is focused on implementing IoT in the field. This research is not based on a specific algorithm but the integration of Hardware, Sensor, Mobile Application, etc. The main focus of this thesis is to implement the IoT to suggest become guide of the user to make life easier and stop wasting of resources. The Methodology section discusses the techniques used to conduct this research work. The primary step was to aggregate training and testing data to analyze and learn from.

4.2 MATERIALS (HARDWARE & SENSORS)

In this thesis work I used total four types of sensor and hardware they are ESP8266 Nodemcu, humidity and temperature sensor (DHT11) and mini water pump.

a. DHT11 Temperature & Humidity Sensor:

The DHT11 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). It's fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds. 33

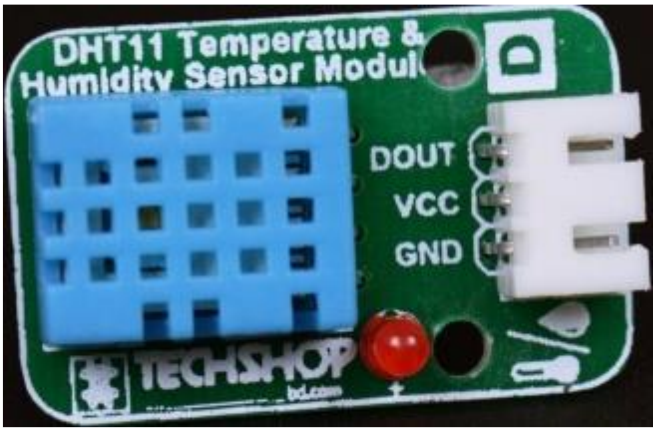

Figure 4.1: DHT11 Sensor (Temperature & Humidity Sensor)

b. ESP8266 WiFi 4 Channel IoT Smart Switch:

The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much Wi-Fi-ability as a Wi-Fi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community. This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts. There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support. In the *Documents* section below you will find many resources to aid you in using the ESP8266, even instructions on how to transforming this module into an IoT (Internet of Things) solution!

- Supports Wifi Board: ESP8266 NodeMCU Lua WiFi with CP2012
- DHT11 Temperature and Humidity Sensor (Not Included)
- AC 220V Supported
- 4 Independently Controlled Relays
- Operating Voltage: 5V (From AC 220V / USB 5V)
- Power Indication: Red LED
- Relay Output Status: Blue LED CH1, CH2, CH3, CH4
- Each relay output capacity: 220VAC / 10A
- Three Terminal Output AC-DC Control (NO COM NC)
- Single Pole Double Throw Relay Output (SPDT)
- Relay current consumption at 5V is 72mA each when switched on
- Back EMF protection
- Mounting Holes: 2.65 Inch

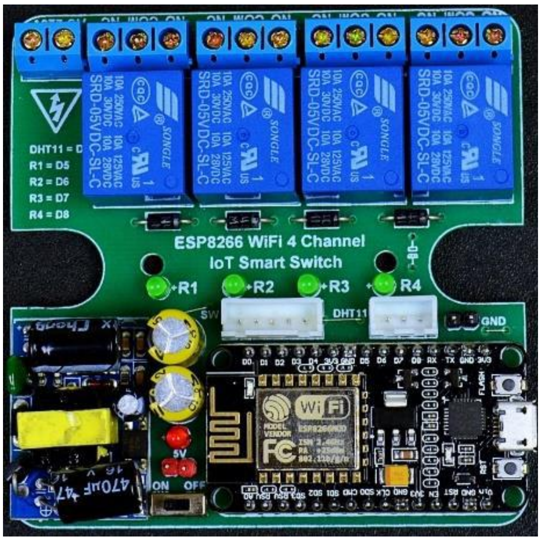

Figure 4.2: ESP8266 WiFi 4 C hannel IoT Smart Switch

4.3 MATERIALS (SOFTWARE & LANGUAGES)

I used the following languages to implement the demonstration for this project. This project has the following components:

1. Hardware Microcontroller programming using Arduino IDE in C (For ESP8266 Smart Switch)
2. PHP & MySQL (Web Server API system for saving the data)
3. Android Application (User to control the IoT Hardware used in Home or Industry)

4.4 FLOW CHART

The flow chart given below describes the functionality of the system clearly. In the flow chart (Figure: 3.5), first the system collects data of the sensors, also collect data of the temperature and humidity of the environment, user information, currently used devices status.

After taking these inputs, it sends the data to the server. Server stores the data sent from Hardware and request a real time feedback. After that the server (web application) sends the prediction data to the mobile apps as form of notification.

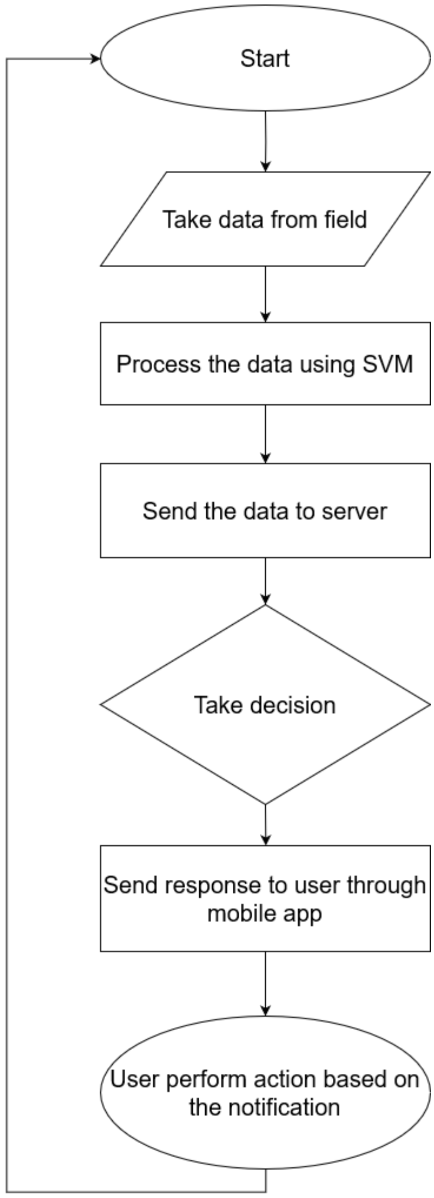

Figure 4.3: Flow chart of the System

4.5 EXPERIMENTAL SETUP

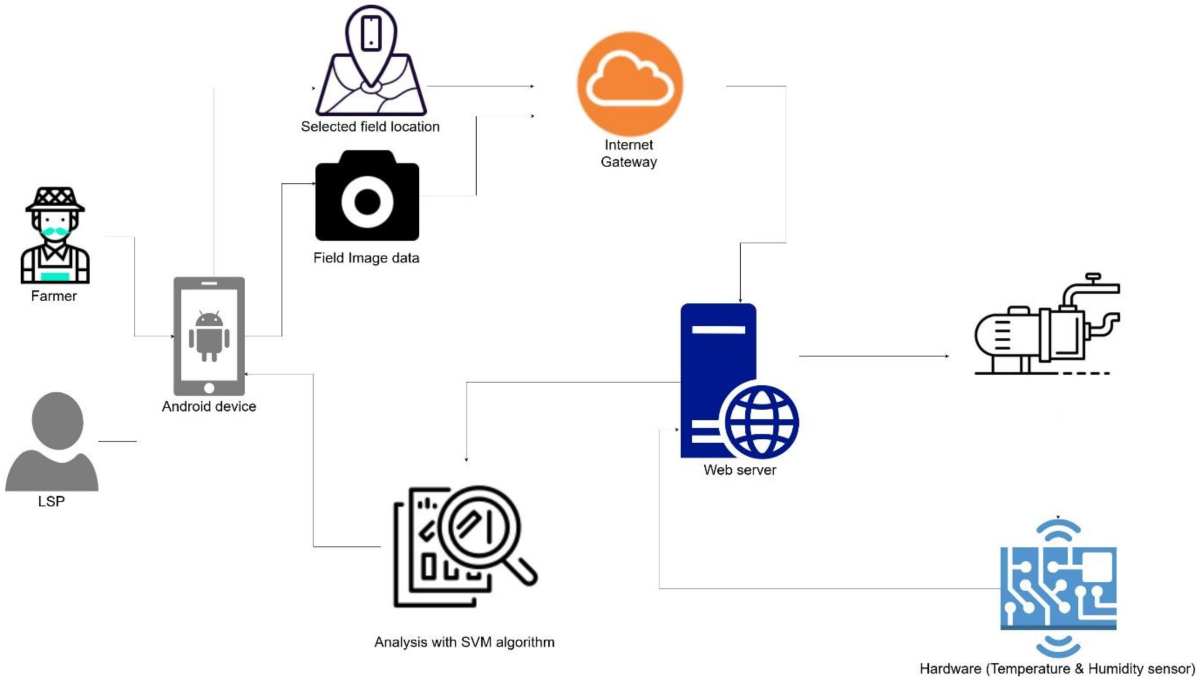

Figure 4.4: Picture of Functional Diagram

The above picture describes how all the technologies software and hardware are communicating with each other.

- The sensor and microcontroller which I define set top box is connected to the internet through WIFI network. They send data to the application server.
- The web server stores the data and applies SVM algorithm to get the prediction analysis result. Based on that it sends notification to the mobile app using real time API.
- Mobile app communicates with the web application server for validating user and get the hardware information. With this information mobile app can get the real time data through API and server.

4.6 STUDY AREA

The study area is located in Bangladesh which is characterized by sub-tropical humid climate. There are four seasons – dry winter season lasting from December to February, pre-monsoon hot summer season lasting from March to May, rainy monsoon season lasting from June to September, and post-monsoon autumn season lasting from October to November. Maize and wheat are the most popular crops after rice but while maize can be grown both during the winter as well as the summer season, wheat can be grown only during the dry winter season. Mung beans on the other hand are usually grown in the post-monsoon dry season. Various crops were sampled during the field surveys such as wheat, maize and mung beans. The field surveys were conducted during the cropping seasons of 2014, 2015 and 2016 **[29]**. For the purposes of this Master's thesis only data from 2016 was used. The field surveys took place in Barisal, Kalapara and Patuakhali located in the south part of the country (Fig 3.1).

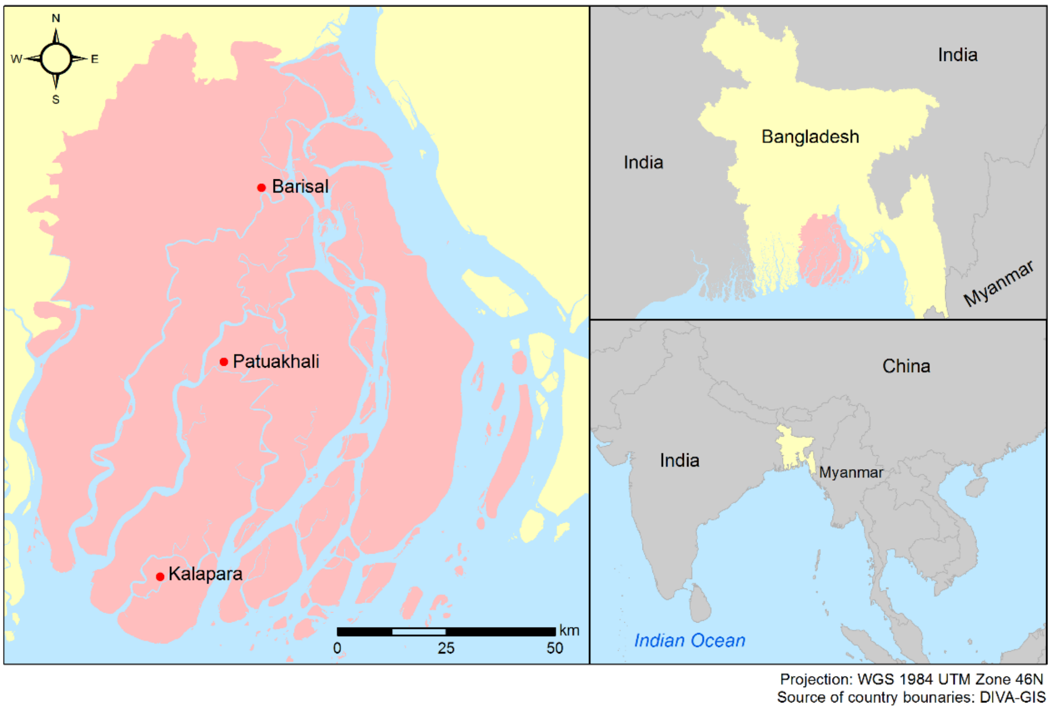

Fig 4.5: Study area

4.7 DATA

Several fields were visited and photos were collected using smartphones. During the field campaign in 2014, only one smartphone model was used to collect RGB images. However, some of the photos were taken only with certain devices and not with all smartphones available. These photos were excluded from the dataset so that each device type is represented by an equal number of photos. Each of the pictures was taken vertically above ground and covered approximately 1 m2 as measured by a square frame placed on the ground during the field surveys. The sample datasets are given below:

This images was taken from above at around 2 to 2.5 feet height.

Fig 4.6: Image sample with crop

This image was taken from 1 to 1.5 feet distance to learn soil moisture.

Fig 4.7: Image sample of soil

To get temperature and humidity data, here DHT11 is used.

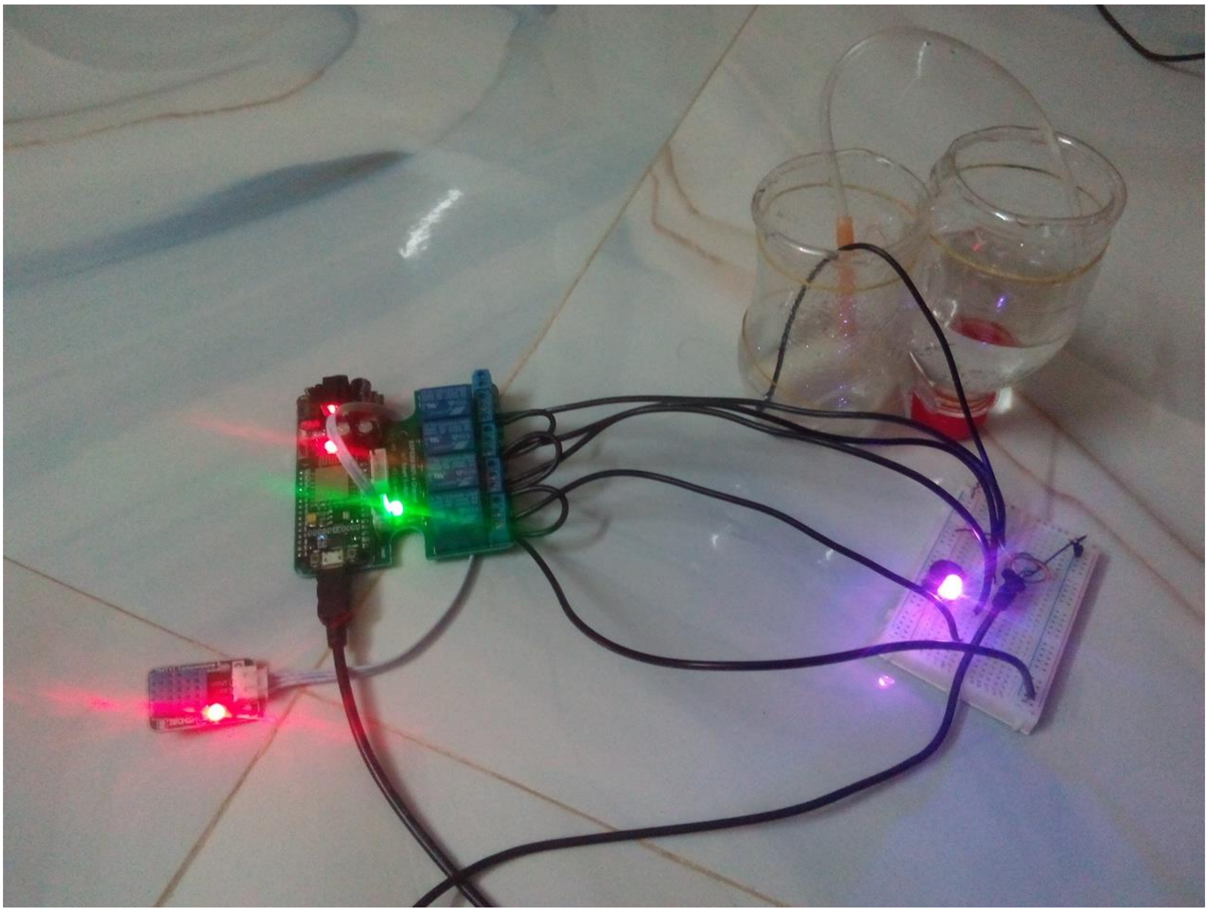

Fig 4.8: Working hardware

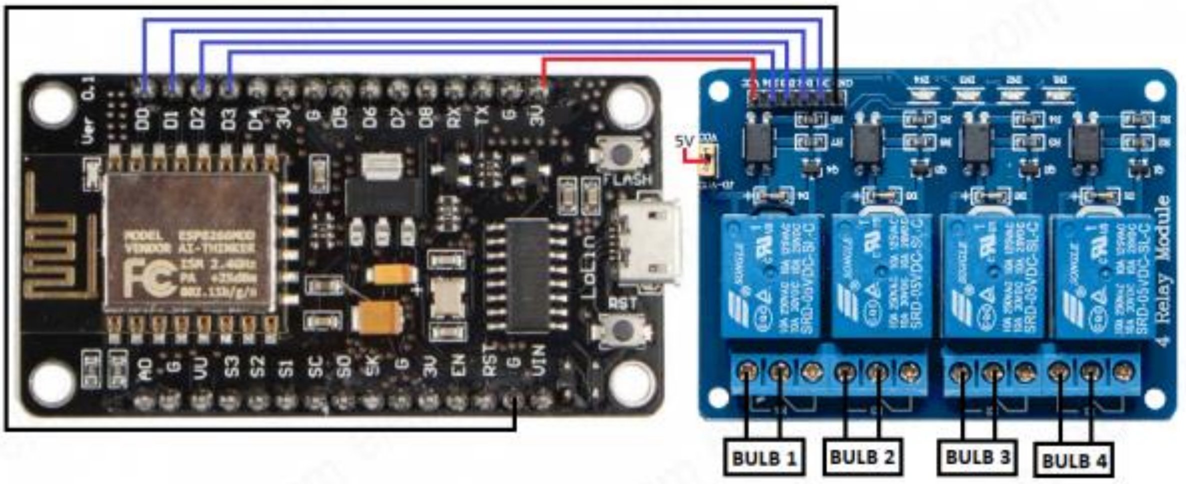

Fig 4.9: Hardware diagram

4.8 PROPOSED MODEL

From the figure (Fig 4.4), we can see that there will be two end user. One will get services (farmer) and another one will provide (Local Service Provider) the services. Both will use a smart platform, by which they can post images and those data along with temperature and humidity data from hardware will be stored the database. After analyzing those data with SVM algorithm, the result will be displayed to the end users.

4.9 PROPOSED SYSTEM

For fulfilling the objective, a dynamic android irrigation scheduling application is proposed to be developed.

Fig 4.10: Intro interface of app

Fig 4.11: Available user option

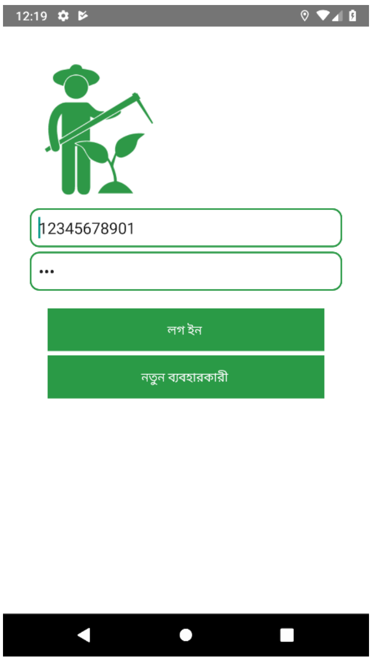

Fig 4.12: Login page

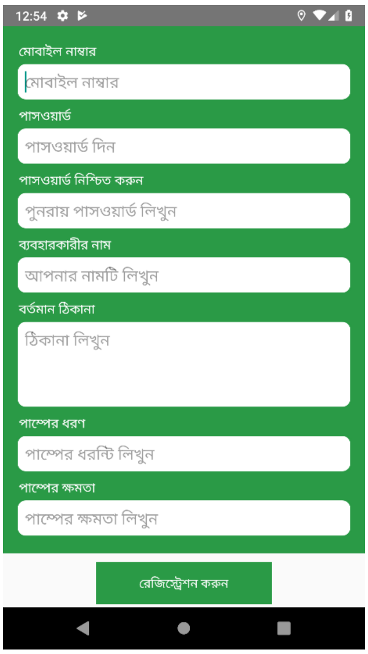

Fig 4.13: Registration page

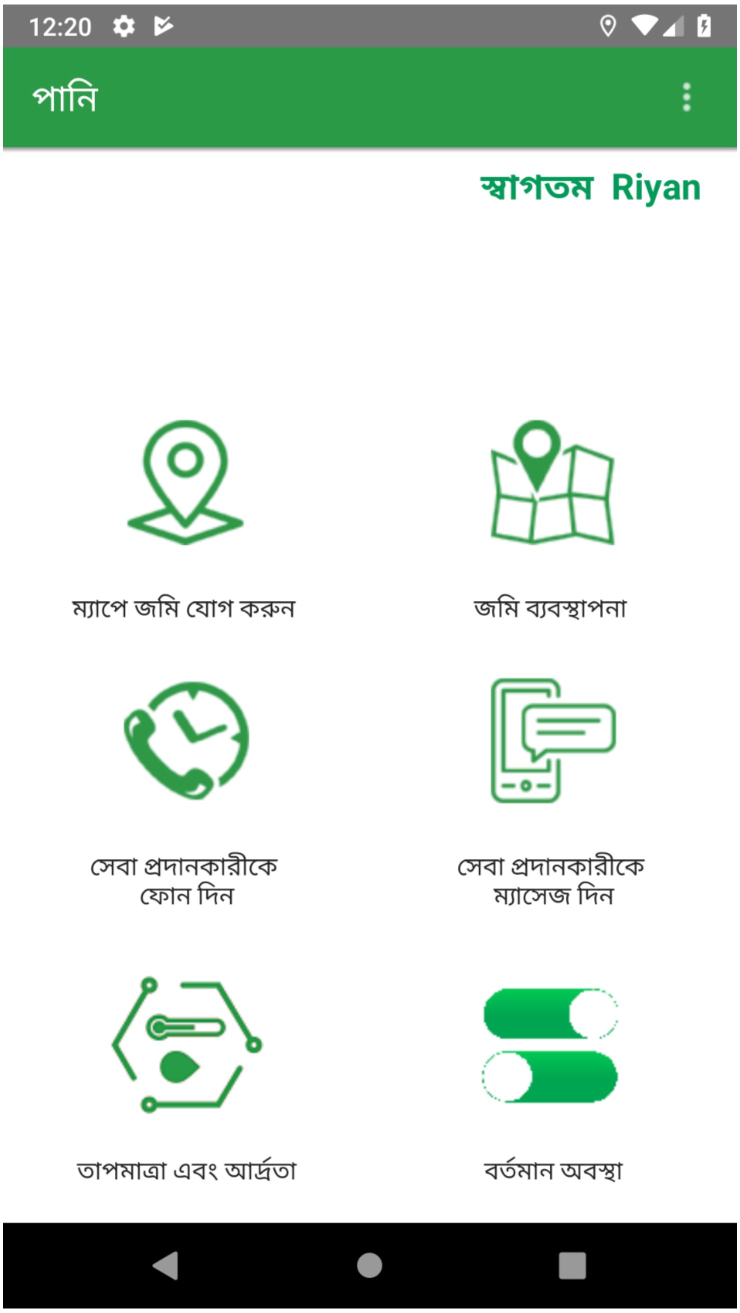

Fig 4.14: Dashboard

4.10 DATA UPLOAD & ANALYSING

Through this application, three kinds of data can be uploaded. GEO data from google map, image data of the field and weather data from the hardware.

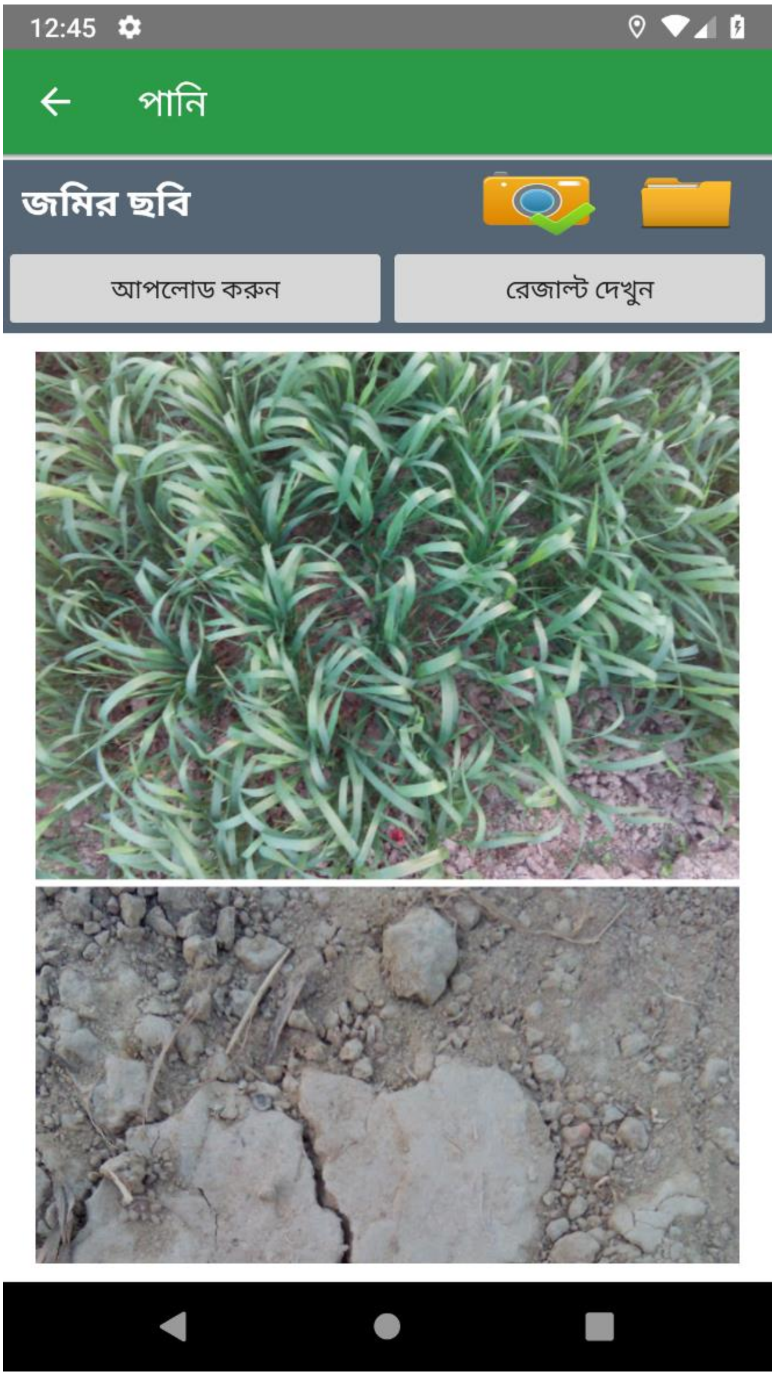

Fig 4.15: Upload field image in the server

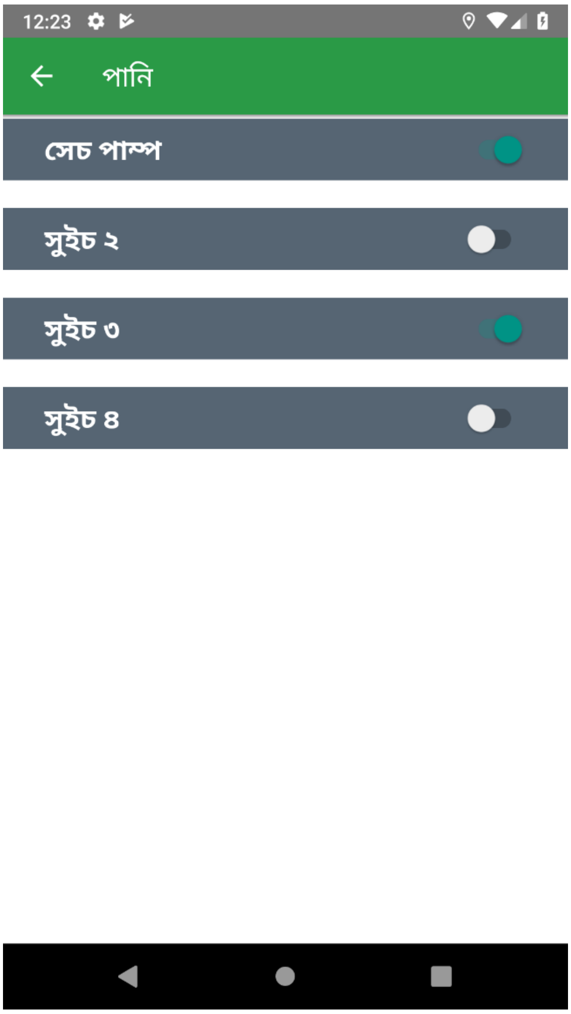

Fig 4.16: Control field pump & devices remotely **Fig 4.17: Select field using google map**

+ Options

←T→	id	temperature	humidity	moisture	time
Edit Copy Delete	1	24.43	14.76	6.87	2019-06-25 01:38:49
Edit Copy Delete	2	34.43	24.76	13.87	2019-06-25 01:40:15
Edit Copy Delete	3	37	93.8	46.87	2019-07-10 15:20:48
Edit Copy Delete	4	27	73.8	26.87	2019-08-24 14:41:32
Edit Copy Delete	5	29	44	24	2019-09-05 02:00:09
Edit Copy Delete	6	29.60	86.00	30	2019-09-10 01:24:16
Edit Copy Delete	7	29.50	86.00	30	2019-09-10 01:24:26
Edit Copy Delete	8	29.40	87.00	30	2019-09-10 01:24:37
Edit Copy Delete	9	29.50	87.00	30	2019-09-10 01:24:46
Edit Copy Delete	10	29.50	86.00	30	2019-09-10 01:24:56
Edit Copy Delete	11	29.50	87.00	30	2019-09-10 01:25:08
Edit Copy Delete	12	29.50	86.00	30	2019-09-10 01:25:16
Edit Copy Delete	13	29.50	86.00	30	2019-09-10 01:25:26
Edit Copy Delete	14	29.50	86.00	30	2019-09-10 01:25:37
Edit Copy Delete	15	29.50	87.00	30	2019-09-10 01:25:46
Edit Copy Delete	16	29.50	86.00	30	2019-09-10 01:25:56
Edit Copy Delete	17	29.50	87.00	30	2019-09-10 01:26:07
Edit Copy Delete	18	29.50	87.00	30	2019-09-10 01:26:16
Edit Copy Delete	19	29.50	87.00	30	2019-09-10 01:26:26

Fig 4.18: Server data collected from DHT11

4.11 OVERALL APPLICATION STRUCTURE

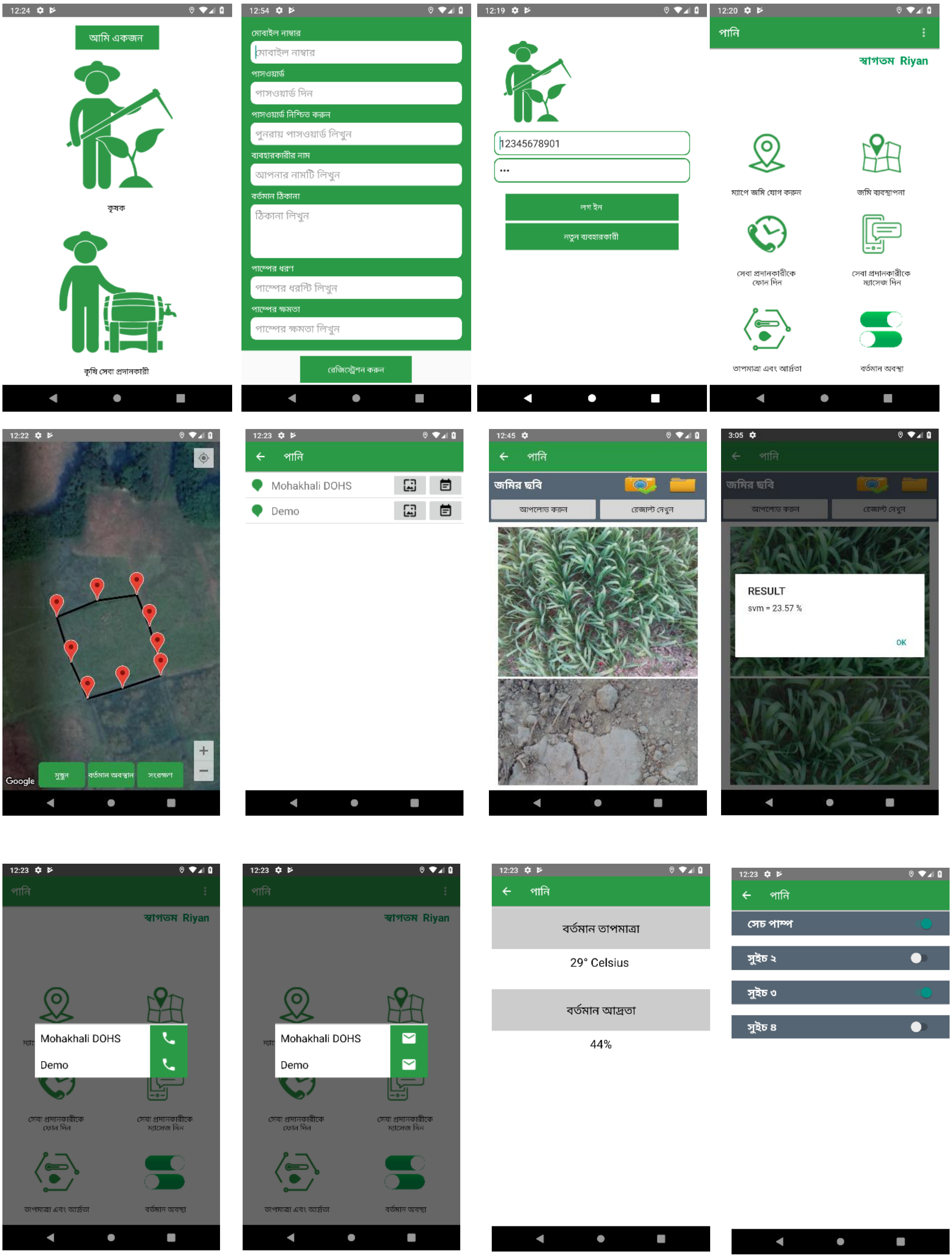

Fig 3.16: Overall application structure

Chapter 5

Results and Discussion

5.1 INTRODUCTION

In this part of the paper I discussed about results and solution that I prepared so far. The results and plotting's are based on the data, training and validation.

This section contains the result found by SVM algorithm simulation. There are two phase of results. One is to clarify the percentage of SVM and the other one is to show temperature and humidity data collected from the field. Through this application, three kinds of data can be uploaded. GEO data from google map, image data of the field and weather data from the hardware.

O one smartphone model was used to collect RGB images. However, some of the photos were taken only with certain devices and not with all smartphones available. These photos were excluded from the dataset so that each device type is represented by an equal number of photos. Each of the pictures was taken vertically above ground and covered approximately 1 m2 as measured by a square frame placed on the ground during the field surveys. The sample datasets are given below:

5.2 RESULT

This section contains the result found by analysing the field data.

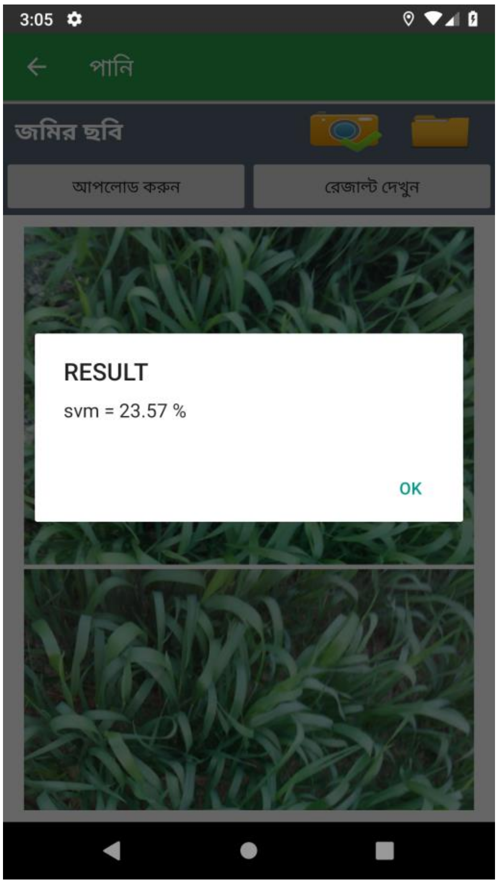

Fig 5.1: SMV result

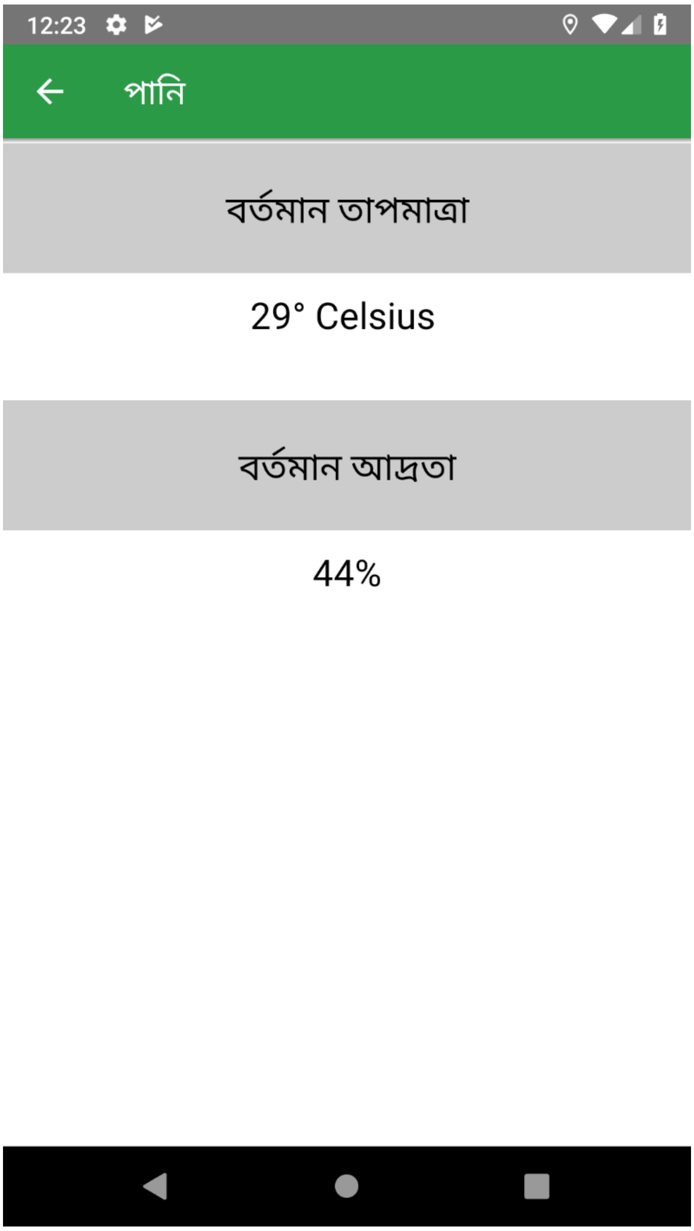

Fig 5.2: Data reading from DHT11 hardware

From the figure (Fig 5.1), we can see that SVM is 23.67%. To know the result, whether we need to give water in the field or not, the below conditions are applied.

- If the result go above 50%, then it is necessary to give water in the field.
- More the percentage is displayed, more the field needs water.
- Sometimes the field needs water, but there is possibility of raining. That time no need of additional water, so no need to start the water pump.
- The water pump can be operated remotely through the mobile app.

If SVM ≥ 50%, Then time for irrigation

If SVM < 50%, Then no need of irrigation

SVM ∝ irrigation

Chapter 6

Conclusion

6.1 CONCLUSION

The proposed research work provides a dynamic smart irrigation system using IoT. Experiments have been conducted with the system. Various data are recorded through cloud server using web platform and from hardware input. The system used environment temperature, humidity, Sensor reading and actual temperature, humidity and user activity values to analysis. After learning the system can predict the needs of plants and can suggest in advance what the user might be expecting based on the activities it recorded so far. As the real-time sensor data and activity data are uploaded to cloud server, it populates random large amount of user data with activity and sensor reading which can be used later for data mining.

In this paper, I have designed and implemented a dynamic, smart irrigation system (PANI) with a view to make an efficient use of all the resources in the field. This is the era of industry 4.0 where every device not only needs to work efficiently also need to behave intelligent. This thesis intended to integrate almost all the devices that I used in our daily life. So if the system perform well enough as expectation then this system will know us like hardware will interact with us.

6.2 FUTURE WORK

Since the proposed system is interacting with many people and learning their behavior over time it can assist many people to get the benefits of the AI and can become their daily assistant. This thesis work is basically a concept of integrating all our daily used devices to interact with us. So the scope of work for this thesis is literally of limits. But the future work focus areas can be divided into some categories like integration of new hardware's and enhancing the system. Big data analysis can be done based on the data it will collect to find out some new pattern and update the model and train to adopt more accurate behavior predictions. I can just think of all the devices and hardware's around us will respond according to our behavior.

Bibliography

1. M. Baten and R. Titumir, "Environmental challenges of trans-boundary water resources management: the case of Bangladesh", Sustainable Water Resources Management, vol. 2, no. 1, pp. 13-27, 2015.
2. K. Morgan, "Smart irrigation: Agricultural water savings with improved irrigation scheduling", Agrotechnology, vol. 06, no. 04, 2017.
3. M. Faroque, M. Kashem and S. Bilkis, "Sustainable Agriculture: A Challenge in Bangladesh", International Journal of Agricultural Research, Innovation and Technology, vol. 1, no. 1-2, pp. 1-8, 2013.
4. Akter, Sharmin & Mahanta, Pinkirani & Mim, Maliha & Hasan, Rakib & Ahmed, Raziun & Billah, Mostasim & Sultana, Arifin. (2018). Developing a Smart Irrigation System Using Arduino. 6. 31-39.
5. Caetano, F., Pitarma, R., & Reis, P. (2015). Advanced System for Garden Irrigation Management. In New Contributions in Information Systems and Technologies (pp. 565–574).
6. Chavan, C. H., & Karande, P. V. (2014). Wireless Monitoring of Soil Moisture, Temperature & Humidity Using Zigbee in Agriculture. International Journal of Engineering Trends and Technology, 11(10), 493–497.
7. Dobbs, N. A., Migliaccio, K. W., Li, Y., Dukes, M. D., & Morgan, K. T. (2014). Evaluating irrigation applied and nitrogen leached using different smart irrigation technologies on bahiagrass (Paspalum notatum). Irrigation Science, 32, 193–203.
8. Gutiérrez, J., Villa-medina, J. F., Nieto-garibay, A., & Porta-gándara, M. Á. (2013). Automated Irrigation System Using a Wireless Sensor Network and GPRS Module. In IEEE transactions on instrumentation and measurement (pp. 1–11).
9. Houstis, E., Nasiakou, A., & Vavalis, M. (2017). Linking Smart Energy and Smart Irrigation: Integration, System Architecture, Prototype Implementation and Experimentation. In 3rd International Congress on Energy Efficiency and Energy Related Materials (ENEFM2015) (pp. 143–149). Springer International Publishing AG.
10. Keswani, B., Mohapatra, A. G., Mohanty, A., Khanna, A., Rodrigues, J. J. P. C., Gupta, D., & Hugo, V. (2018). Adapting weather conditions based IoT enabled smart irrigation technique in precision agriculture mechanisms. Neural Computing and Applications, 30(6), 1–16.
11. Kinjal, A. R., Patel, B. S., & Bhatt, C. C. (2018). Smart Irrigation: Towards Next Generation Agriculture. In Internet of Things and Big Data Analytics Toward Next-Generation Intelligence (pp. 265–282).
12. Ososanya, E. T. (2015). Design and Implementation of a Solar-Powered Smart Irrigation System. In 122nd ASEE Annual Conference & Exposition (pp. 1–15).

13. Parameswaran, G., &K.Sivaprasath. (2016). Arduino Based Smart Drip Irrigation System Using Internet of Things. International Journal of Engineering Science and Computing, 6(5), 5518–5521.
14. Reddy, A. M., & Rao, K. R. (2016). An Android based Automatic Irrigation System using a WSN and GPRS Module. Indian Journal of Science and Technology, 9(29), 1–6.
15. Addink, E.A., Van Coillie, F.M.B. & de Jong, S.M. 2012, "Introduction to the GEOBIA 2010 special issue: From pixels to geographic objects in remote sensing image analysis", International Journal of Applied Earth Observation and Geoinformation, vol. 15, no. 1, pp. 1-6.
16. Adhikari, H., Heiskanen, J., Maeda, E.E. & Pellikka, P.K.E. 2016, "The effect of topographic normalization on fractional tree cover mapping in tropical mountains: An assessment based on seasonal Landsat time series", International Journal of Applied Earth Observation and Geoinformation, vol. 52, pp. 20-31.
17. Afzal, M. A., Bakr, M. A., Hamid, A., Haque, M. M., & Aktar, M. S. 2006, "Adoption and seed production mechanisms of modern varieties of mungbean in Bangladesh", Improving income and nutrition by incorporating mungbean in cereal fallows in the Indo-Gangetic Plains of South Asia, DFID Mungbean Project for 2002–2004.
18. AgiSoft PhotoScan, 2014, St. Petersburg, Russia: Agisoft LLC.
19. Agüera, F., Carvajal, F. & Pérez, M. 2011, "Measuring sunflower nitrogen status from an unmanned aerial vehicle-based system and an on the ground device", International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, pp. 33-37.
20. Anderson, K., & Gaston, K. J. 2013, "Lightweight unmanned aerial vehicles will revolutionize spatial ecology", Frontiers in Ecology and the Environment, vol. 11, no. 3, pp. 138-146.
21. Aquino, A., Millan, B., Gaston, D., Diago, M. & Tardaguila, J. 2015, "vitisFlower®: Development and testing of a novel android-smartphone application for assessing the number of grapevine flowers per inflorescence using artificial vision techniques", Sensors (Switzerland), vol. 15, no. 9, pp. 21204-21218.
22. Balota, M. & Oakes, J. 2016, "Exploratory use of a UAV platform for variety selection in peanut", Proceedings of SPIE - The International Society for Optical Engineering.
23. Bannari, A., Staenz, K. & Khurshid, K.S. 2007, "Remote sensing of crop residue using Hyperion (EO-1) data", International Geoscience and Remote Sensing Symposium (IGARSS), pp. 2795-2799.
24. Baret, F. & Buis, S. 2008, "Estimating canopy characteristics from remote sensing observations: Review of methods and associated problems", Advances in Land Remote Sensing: System, Modeling, Inversion and Application, pp. 173-201.

25. Baret, F. & Guyot, G. 1991, "Potentials and limits of vegetation indices for LAI and APAR assessment", Remote Sensing of Environment, vol. 35, no. 2-3, pp. 161-173.
26. Baret, F., Guyot, G. & Major, D.J. 1989, "TSAVI: A vegetation index which minimizes soil brightness effects on LAI and APAR estimation", Digest - International Geoscience and Remote Sensing Symposium (IGARSS), pp. 1355-1358.
27. Baret, F., Hagolle, O., Geiger, B., Bicheron, P., Miras, B., Huc, M., Berthelot, B., Niño, F., Weiss, M., Samain, O., Roujean, J.L. & Leroy, M. 2007, "LAI, fAPAR and SVM CYCLOPES global products derived from VEGETATION. Part 1: Principles of the algorithm", Remote Sensing of Environment, vol. 110, no. 3, pp. 275-286.
28. Baret, F., Weiss, M., Allard, D., Garrigues, S., Leroy, M., Jeanjean, H., ... & Bohbot, H. 2005, "VALERI: a network of sites and a methodology for the validation of medium spatial resolution land satellite products", Remote Sensing of Environment, vol. 76, no. 3, pp. 36-39.
29. Bastiaanssen, W.G., Molden, D.J., & Makin, I.W. 2000, "Remote sensing for irrigated agriculture: examples from research and possible applications", Agricultural water management, vol. 46, no. 2, pp. 137-155.
30. Bauer, J., Siegmann, B., Jarmer, T. & Aschenbruck, N. 2016, "Smart fLAIr: A smartphone application for fast LAI retrieval using Ambient Light Sensors", SAS 2016 - Sensors Applications Symposium, Proceedings, pp. 401-406.
31. Bauer, T. & Strauss, P. 2014, "A rule-based image analysis approach for calculating residues and vegetation cover under field conditions", Catena, vol. 113, pp. 363-369.
32. Bendig, J., Yu, K., Aasen, H., Bolten, A., Bennertz, S., Broscheit, J., Gnyp, M.L. & Bareth, G. 2015, "Combining UAV-based plant height from crop surface models, visible, and near infrared vegetation indices for biomass monitoring in barley", International Journal of Applied Earth Observation and Geoinformation, vol. 39, pp. 79-87.
33. Benz, U.C., Hofmann, P., Willhauck, G., Lingenfelder, I. & Heynen, M. 2004, "Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information", ISPRS Journal of Photogrammetry and Remote Sensing, vol. 58, no. 3-4, pp. 239-258.

Appendix

SVM computer:

```java
package turza.ewu.com.pani_200;

import android.content.res.Resources;
import android.os.Environment;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;

import libsvm.svm;
import libsvm.svm_model;
import libsvm.svm_node;

public class SvmComputer {

  private svm_model model;
  private final int bucketsize;
  private String model_file_name = "";
  private boolean valid = false;
  private Double [] histogramMin;
  private Double [] histogramMax;

  public static List<Integer> additionalModels() {
    List<Integer> result = new ArrayList<Integer>();
    File sdcard = Environment.getExternalStorageDirectory();
    for (int bucketsize = 16; bucketsize < 256; bucketsize *= 2)
    {
      String model_file_name = "svm_trained_" + bucketsize + ".model";
      File file = new File(sdcard, BuildConfig.APPLICATION_ID + "/" + model_file_name);
      if (file.exists())
        result.add(bucketsize);
    }
    return result;
  }

  public SvmComputer(int bucketsize, Resources resources) {
    this.bucketsize = bucketsize;
    this.model = null;
    model_file_name = "svm_trained_" + bucketsize + ".model";
    InputStream is = null;
    InputStream isMin = null;
    InputStream isMax = null;
    switch (bucketsize) {
      case 2:
        is = resources.openRawResource(R.raw.svm_trained_2);
        isMin = resources.openRawResource(R.raw.min_2);
        isMax = resources.openRawResource(R.raw.max_2);
        break;
      case 4:
        is = resources.openRawResource(R.raw.svm_trained_4);
        isMin = resources.openRawResource(R.raw.min_4);
        isMax = resources.openRawResource(R.raw.max_4);
        break;
      case 8:
        is = resources.openRawResource(R.raw.svm_trained_8);
        isMin = resources.openRawResource(R.raw.min_8);
```

```
        isMax = resources.openRawResource(R.raw.max_8);
        break;
    }
    if (is != null && isMin != null && isMax != null) {
      try {
        model = svm.svm_load_model(new BufferedReader(new InputStreamReader(is)));
        histogramMin = readHistogramMinMax(bucketsize, isMin);
        histogramMax = readHistogramMinMax(bucketsize, isMax);
        valid = true;
      } catch (IOException e) {
      }
    } else {
      File sdcard = Environment.getExternalStorageDirectory();
      File file = new File(sdcard, BuildConfig.APPLICATION_ID + "/" + model_file_name);
      File fileMin = new File(sdcard, BuildConfig.APPLICATION_ID + "/min_" + bucketsize + ".histogram");
      File fileMax = new File(sdcard, BuildConfig.APPLICATION_ID + "/max_" + bucketsize + ".histogram");
      if (file.exists() && fileMin.exists() && fileMax.exists()) {
        try {
          String model_path = String.valueOf(file);
          model = svm.svm_load_model(model_path);
          histogramMin = readHistogramMinMax(bucketsize, new FileInputStream((fileMin)));
          histogramMax = readHistogramMinMax(bucketsize, new FileInputStream(fileMax));
          valid = true;
        } catch (IOException e) {
        }
      }
    }
  }

  private Double [] readHistogramMinMax(final int bucketsize, InputStream is) {
    final int elements = bucketsize * bucketsize * bucketsize;
    try {
      BufferedReader reader = new BufferedReader(new InputStreamReader(is));
      String line = reader.readLine();
      String [] sValues = line.split(",");
      Double [] values = new Double[elements];
      for (int i = 0; i < elements; ++i) {
        String sVal = sValues[i];
        values[i] = Double.parseDouble(sVal);
      }
      return values;
    } catch (IOException e) {
      e.printStackTrace();
    }
    return null;
  }

  private void scaleHistogram(List<Double> histogramValues) {
    final int elements = bucketsize * bucketsize * bucketsize;
    for (int i = 0; i < elements; ++i) {
      Double newVal = (histogramValues.get(i) - histogramMin[i]) / (histogramMax[i] - histogramMin[i]);
      if (newVal.isNaN())
        newVal = histogramMin[i];
      else if (newVal.isInfinite())
        newVal = histogramMax[i];
      histogramValues.set(i, newVal);
    }
  }

  public String computeSVM (String fileName, Map<String, Double> results, ProgressMonitor progressMonitor) {
    progressMonitor.initSubJob(3, 30);
    if (valid) {
      progressMonitor.workedSubJob();
      List<Double> histogram = Histogram.getHistogram(fileName, bucketsize);
      progressMonitor.workedSubJob();
      if (histogram != null) {
        scaleHistogram(histogram);
```

```
          double SVM  = predictSVM(histogram);
          progressMonitor.workedSubJob();
          results.put(fileName, SVM );
          progressMonitor.endSubJob();
          return "";
        } else {
          progressMonitor.endSubJob();
          return "SVM: Failed to compute histogram of '" + fileName + "'";
        }
      } else {
        progressMonitor.endSubJob();
        return "SVM: File '" + model_file_name + "' is missing.";
      }
    }

    private double predictSVM(List<Double> histogram) {
      int m = histogram.size();
      svm_node[] x = new svm_node[m];
      for (int j = 0; j < m; j++) {
        x[j] = new svm_node();
        x[j].index = j;
        x[j].value = histogram.get(j);
      }
      double value = svm.svm_predict(model, x);
      return value;
    }
  }
```

Temperature and Humidity data:

```
package turza.ewu.com.pani_200;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.widget.TextView;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

import turza.ewu.com.pani_200.exceptions.BaseException;
import turza.ewu.com.pani_200.utils.Util;

public class temphum extends AsyncTask<Void,Void,Void> {

    String data = "";
    String dataParsed = "";
    String singleParsed = "";

    @Override
```

```
    protected Void doInBackground(Void... voids) {
        try {
            URL url = new URL("http://sme.com.bd/pani/post_temp.php");
            HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
            InputStream inputStream = httpURLConnection.getInputStream();
            BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));

            String line = "";
            while (line != null){
                line = bufferedReader.readLine();
                data = data + line;
            }

            JSONObject jsonObject = new JSONObject(data);
            JSONArray array = jsonObject.getJSONArray("result");

            for (int i=0; i<array.length(); i++){
                JSONObject JO = (JSONObject) array.getJSONObject(i);

                singleParsed = JO.get("temperature") + "° Celsius ";

                dataParsed = JO.get("humidity") + "%";

            }

        } catch (MalformedURLException e){
            e.printStackTrace();
        } catch (IOException e){
            e.printStackTrace();
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(Void aVoid) {
        super.onPostExecute(aVoid);

        ImageAnalysisInternalValue.data.setText(this.singleParsed);
        ImageAnalysisInternalValue.data2.setText(this.dataParsed);
    }
}
```

Remote access activity:

```
package turza.ewu.com.pani_200;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.widget.CompoundButton;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.FileInputStream;
```

```java
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import turza.ewu.com.pani_200.exceptions.BaseException;
import turza.ewu.com.pani_200.utils.PumpStatusPojo;
import turza.ewu.com.pani_200.utils.Util;

public class switch_statusActivity extends AppCompatActivity {

    private static final int PERMISSION_REQUEST_STORAGE = 1000;
    private TextView textView2;
    private Switch pumpSwitch, switchTwo, switchThree, switchFour;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_switch_status);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        textView2 = (TextView) findViewById(R.id.textView2);

        pumpSwitch = (Switch) findViewById(R.id.switch1);
        switchTwo = (Switch) findViewById(R.id.switch2);
        switchThree = (Switch) findViewById(R.id.switch3);
        switchFour = (Switch) findViewById(R.id.switch4);

        pumpSwitch.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
                if (b) {
                    Toast.makeText(switch_statusActivity.this, "Pump is on", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(switch_statusActivity.this, "Pump is off", Toast.LENGTH_SHORT).show();
                }
            }
        });


        switchTwo.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
                if (b) {
                    Toast.makeText(switch_statusActivity.this, "Switch 2 is on", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(switch_statusActivity.this, "Switch 2 is off", Toast.LENGTH_SHORT).show();
                }
            }
        });


        switchThree.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
                if (b) {
                    Toast.makeText(switch_statusActivity.this, "Switch 3 is on", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(switch_statusActivity.this, "Switch 3 is off", Toast.LENGTH_SHORT).show();
                }
```

```java
        }
    });


    switchFour.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
            if (b) {
                Toast.makeText(switch_statusActivity.this, "Switch 4 is on", Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(switch_statusActivity.this, "Switch 4 is off", Toast.LENGTH_SHORT).show();
            }
        }
    });

    PumpStatus pumpStatus = new PumpStatus();
    pumpStatus.execute();

}

@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {

}

private class PumpStatus extends AsyncTask<Void, Void, List<PumpStatusPojo> > {

    String data = "";
    String dataParsed = "";
    String singleParsed = "";
    String switchNo;
    String switchVal;
    List<PumpStatusPojo> pumpStatusList = new ArrayList<>();

    @Override
    protected List<PumpStatusPojo> doInBackground(Void... voids) {
        try {
            URL url = new URL("http://sme.com.bd/pani/get_switch.php");
            HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
            InputStream inputStream = httpURLConnection.getInputStream();
            BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inputStream));

            String line = "";
            while (line != null) {
                line = bufferedReader.readLine();
                data = data + line;
            }

            JSONObject jsonObject = new JSONObject(data);
            JSONArray array = jsonObject.getJSONArray("result");

            for (int i = 0; i < array.length(); i++) {
                JSONObject JO = (JSONObject) array.getJSONObject(i);

                switchNo = JO.get("switch_no") + "";

                switchVal = JO.get("status") + "";

                pumpStatusList.add(new PumpStatusPojo(switchNo, switchVal));

            }

        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } catch (JSONException e) {
```

```
            e.printStackTrace();
        }
        return pumpStatusList;
    }

    @Override
    protected void onPostExecute(List<PumpStatusPojo> aVoid) {
        super.onPostExecute(aVoid);

        for(PumpStatusPojo obj : aVoid){
            String tempPumpNO = obj.getSwitchNo();
            String tempPumpVal = obj.getSwitchValue();
            if(tempPumpNO.equals("1")){
                if(tempPumpVal.equals("1")){
                    pumpSwitch.setChecked(true);
                }
            }else if(tempPumpNO.equals("2")){
                if(tempPumpVal.equals("1")){
                    switchTwo.setChecked(true);
                }
            }else if(tempPumpNO.equals("3")){
                if(tempPumpVal.equals("1")){
                    switchThree.setChecked(true);
                }
            }else if(tempPumpNO.equals("4")){
                if(tempPumpVal.equals("1")){
                    switchFour.setChecked(true);
                }
            }
        }
    }
  }
}
```

:

Arduino Code:

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <Hash.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>

const char* ssid = "*****";
const char* password = "*****";

#define DHTPIN 5     // Digital pin connected to the DHT sensor

#define DHTTYPE    DHT22     // DHT 22 (AM2302)

DHT dht(DHTPIN, DHTTYPE);

float t = 0.0;
float h = 0.0;

AsyncWebServer server(80);

unsigned long previousMillis = 0;    // will store last time DHT was updated

const long interval = 10000;
```

```
const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384-
fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous">
  <style>
    html {
     font-family: Arial;
     display: inline-block;
     margin: 0px auto;
     text-align: center;
    }
    h2 { font-size: 3.0rem; }
    p { font-size: 3.0rem; }
    .units { font-size: 1.2rem; }
    .dht-labels{
      font-size: 1.5rem;
      vertical-align:middle;
      padding-bottom: 15px;
    }
  </style>
</head>
<body>
  <h2>ESP8266 DHT Server</h2>
  <p>
    <i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
    <span class="dht-labels">Temperature</span>
    <span id="temperature">%TEMPERATURE%</span>
    <sup class="units">&deg;C</sup>
  </p>
  <p>
    <i class="fas fa-tint" style="color:#00add6;"></i>
    <span class="dht-labels">Humidity</span>
    <span id="humidity">%HUMIDITY%</span>
    <sup class="units">%</sup>
  </p>
</body>
<script>
setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("temperature").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "/temperature", true);
  xhttp.send();
}, 10000 ) ;

setInterval(function ( ) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("humidity").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "/humidity", true);
  xhttp.send();
}, 10000 ) ;
</script>
</html>)rawliteral";

String processor(const String& var){
  if(var == "TEMPERATURE"){
    return String(t);
```

```
  }
  else if(var == "HUMIDITY"){
    return String(h);
  }
  return String();
}

void setup(){
  Serial.begin(115200);
  dht.begin();

  WiFi.begin(ssid, password);
  Serial.println("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println(".");
  }

  Serial.println(WiFi.localIP());

  server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/html", index_html, processor);
  });
  server.on("/temperature", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", String(t).c_str());
  });
  server.on("/humidity", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", String(h).c_str());
  });

  // Start server
  server.begin();
}

void loop(){
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    float newT = dht.readTemperature();

    if (isnan(newT)) {
      Serial.println("Failed to read from DHT sensor!");
    }
    else {
      t = newT;
      Serial.println(t);
    }

    float newH = dht.readHumidity();
    if (isnan(newH)) {
      Serial.println("Failed to read from DHT sensor!");
    }
    else {
      h = newH;
      Serial.println(h);
    }
  }
}
```