

Smart Gardening: A Solution to Your Gardening Issues

Niloy Chakraborty^{1,*}, Adrika Mukherjee², Mayuri Bhadra³,

¹Data Scientist, SenseGuard GmbH, Düsseldorf, Germany

²Data Scientist, Mercedes-Benz Tech Innovation GmbH, Stuttgart, Germany

³Digital Verification Engineer, Robert Bosch GmbH, Reutlingen, Germany

Abstract

The technology which could make our lives prosper within the four walls could also help to create our own corner of nature nourish. In this paper, we propose a smart gardening system that utilizes the concept of the Internet of Things (IoT) [1]. The major goal of this project is to reduce water consumption when gardening and to maintain the garden remotely. Important plant data, like temperature, relative humidity, and soil moisture, are continuously stored in a relational database in this gardening system. Artificial Intelligence (AI) based planning [2] is used for watering the plants at regular intervals and providing appropriate illumination in the garden area for aesthetics and overall plant growth. Our proposed system reduces the effort due to manual intervention by around 59.3%. The real-time sensor status can be monitored which in turn allows the end-users of the garden to control the surrounding conditions optimal for plant growth, using the Telegram application. A plant recognition model has been introduced in this system, where a Convolutional Neural Network (CNN) [3] based deep learning algorithm classifies the plant categories with 95% accuracy. Moreover, an 98% accurate, deep learning-based, plant health identification model integrated with this gardening system also informs the end-user about the health of the plant.

Received on 10 April 2022; accepted on 31 July 2022; published on 09 August 2022

Keywords: smart garden, AI planning, plant identification, CNN, smart lighting, Internet of Things, smart city

Copyright © 2022 Niloy Chakraborty *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [Creative Commons Attribution license](#), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi:10.4108/eetiot.v8i30.384

1. Introduction

With the rapidly growing popularity of the internet, multipurpose devices (laptops and smartphones) and the devices built to serve a specific purpose (say, an embedded system in a washing machine), the concept of the IoT [1] came into the picture. IoT is a shared network of objects or things that can interact with each other via the Internet connection [1]. In different domains, like, information and communication, business, social process, the things are expected to become active participants by using IoT. They must be able to interact with the environment as well as communicate with one another while exchanging and converting data and information

gathered from it. It is automatically influenced by the processes that create services and trigger actions with or without human intervention [4].

Smart and intelligent systems have become a trend of this era, and with the rapid growth of IoT-based technology and artificial intelligence (AI), many smart applications are being developed commercially. In this work, a gardening system is considered a use case. Maintaining a garden is a time-consuming and laborious activity in and of itself; additionally, watering plants with minimal waste has become a necessity to reduce energy usage and conserve water. In order to experience a greater sense of aesthetics and a breath of fresh air without necessarily being a propagator, there is a need for garden digitization.

In this context, a model which could efficiently take care of a garden on its own with almost no human intervention is proposed in this paper. The proposed gardening system is capable of watering the plants

*Corresponding author. Email: niloychakraborty17@gmail.com

whenever it is needed, by using the AI planning algorithms [2]. The system lets the user know the health situation of the plants. Moreover, upon a single-click request of the user, the associated camera captures the photos of the plants and sends them to the user via integrated Telegram application. Other features of the system include lighting the garden, subscribing to garden status, and knowing the local weather. Finally, there is a possibility of scaling the garden with the option to add new plants.

The rest of the paper is organized as follows: Section 2 notes a few related works, section 3 elaborates on the system design; Section 4 includes the implementation and setup. The experiment processes are briefed in Section 5, and finally, Section 6 concludes with some limitations faced and the future scope for this project.

2. Related Works

In recent years, the number of researches on smart commercial and home gardening has increased. There have been numerous launches in the commercial app-controlled irrigation system, e.g., Eva Aqua by Apple [5], cloudrain [6], hydrowise. [7], etc. These systems use IF-THEN decision systems for triggering the water pumps.

The smart garden management system [8], developed a system that epitomizes the supply of water nutrition and sunlight for nurturing the plants. However, the lack of details makes this project irreproducible.

In [9], the authors applied a smart system in order to monitor and manage paprika growth via collection and monitoring data of crops inside or outside of paprika greenhouses by WSN sensors and images from CCTV cameras. This system improved productivity by reducing costs and improving the convenience of users. [10] proposed the solution applied to the tomato plants for monitoring and watering of the plants through smart devices. Although this project is easily implementable, it was only restricted to the tomato plants, hence ending with a lack of generalization. Although these projects have their benefits of specializing in one particular plant, the lack of generalization makes them inefficient for general purpose gardening.

GreenIQ Smart Garden Hub in [11] implemented an automated commercial garden monitoring and irrigation system. In the aforementioned works, there is a need for the addition of more functionality; like automated planning for watering or tracking the health of the garden plants.

In [12], an IoT-based gardening system is developed using pressure, temperature, moisture, and humidity sensors. The LEDs added to the system can produce light that helps the plants grow faster. Also, the smartphone integration makes the system more convenient

for users. However, the system needs manual intervention.

3. Smart Garden System Design

The overall structure of our system can be described in detail using a layered architecture that includes both physical (sensors and actuators) and virtual characteristics (different services and communication protocols) [13]. The multiple-tier architecture where each tier works independently, helps to ease the complexity of the IoT systems. Besides, offloading tasks to the edges of IoT networks reduces latency, improves privacy, and lowers bandwidth costs in data-driven IoT applications.

Smart garden system design is based on this type of architecture where a static IoT gateway is used to produce and process the data before sending it to the cloud, where the data is generally stored. The whole structure of the smart garden system was modeled in a way to make it easy for the users to take care of the garden by being in any part of the world. The architecture was created with the goal of reducing water waste and enhancing light in the garden area. Conserving water is critical these days, and the architecture ensures that water and light energy are used efficiently for plant sustainability.

As shown in Figure 1, the architecture is divided into five layers. The **physical layer** consists of the *Aeotec MultiSensor 6* which uses Z-Wave wireless protocol [14] to transmit temperature, relative humidity, and luminescence values to the raspberry pi controller. The amount of water a plant needs can be controlled by keeping an eye on the soil moisture. Here, a *capacitive soil moisture sensor* with both analog and digital output pins is used. MQTT, a machine-to-machine messaging protocol [15] is used in the **middleware layer** to communicate the sensor data to the AWS- Relational Database Repository [16]. **IoT gateway** and **network layer** are comprised of raspberry pi and internet protocols. AI planning is performed by a Fast Forward planner [2], every 10 seconds to minimize resource wastage, regulate the watering of the plants, and control the LED grow lights in the garden area. Telegram [17] is a cloud-based instant messaging and VoIP service which is used as a user-level contact point to send messages to the pi. The Raspberry Pi sends the request to the Plugwise, which operates the garden's motor and lamp. Apart from this, this **application layer** allows camera usage(phone web camera set up for demo) to capture photos based on user request.

The design is robust in the sense it is capable of handling any error both at the hardware and software levels. The garden is intelligent enough to respond to user requests as well as environmental conditions.

The design is adaptable and open, allowing for future integration with other devices.

4. Smart Garden System Set-up & Implementation

Figure 2 shows the implementation diagram for the proposed work. The set up consists of two flower pots firmly rooted into the soil. *Aeon Multisensor 6* is calibrated using Domoticz software [18] for temperature, relative-humidity, luminescence, and is connected to the raspberry pi using Zwave stick. *Soil moisture sensor* along with *A-D converter(ADC) [MCP3008]* is used to convert the analog output to digital values.

Plugwise is used to control two *5V submersible pumps* for watering the plants and two *50W LED grow lights* for illumination during low light. *Plugwise* consists of USB stick that is connected to the raspberry pi, Circle+ , and Circle where the appliances are connected. Telegram bot application acts as the User Interface (UI) for the user to manually control the pump, monitor the local environmental and plant conditions.

4.1. Integration of Sensors and Actuators with the Raspberry Pi:

The temperature, relative-humidity from the *Aeon* sensor and soil-moisture value together decides whether the plant needs water or not. The value from the soil moisture sensor is supplied to one of the analog channels on A-D converter that provides a digital input to the GPIO (General purpose input-output) pins of the raspberry pi. Furthermore, the data from sensors is also published on cloud MQTT and subscriber subscribes to this data for storing in the Amazon Web services database.

4.2. Artificial Intelligence Planning Algorithm:

AI Planning is a branch of Artificial Intelligence that studies how to solve planning and scheduling problems using autonomous strategies. Here, we have an initial starting state that we want to change into the desired goal state by performing a series of actions [19].

The problem file for planning is generated in accordance with the sensor values and the current weather conditions (temperature, humidity) to effectively estimate the sufficient moisture level. Sufficient moisture level is also determined depending on the time when the plant was last watered. To compensate for the delay in the reaction time of the system, the moisture level has to be maintained within the level where the condition of the plant is not deteriorated with the excessive supply of water. For this, the sufficient moisture level is then determined according to the lower and upper moisture limit for the respective plant. If the current moisture level is below the lower limit only

then it is considered to be insufficient. This prevents both wastage of water and water-logging. Depending on the declaratively described system based on the current moisture level and the weather conditions, the system's actions are then directed towards the water in various discrete amounts. To retain the simplicity of the system, weather forecast data is not used for the AI planning, and this could be a potential future enhancement of the watering system. Fast-forward (FF) planning [20], a fast state-space search algorithm is used here for efficient run-time performance.

Initially, the domain and problem formulation of AI planning algorithm was done assuming a system consisting of two plants and two motors without any resource constraints. These domain and problem files are generated automatically and are scalable for any number of plants with the same domain formulation. The moisture level ($p1_sufficient/p1_insufficient$) for each of the plant ($p1$ and $p2$) is determined according to the required moisture level considering both upper and lower moisture limits, the time when the plant was last watered and weather conditions. Also the current scenario whether the motor ($m1$ and $m2$) is already in use ($free_m1/ free_m2$) must be taken into account. To explain the planning algorithm, let us consider the state model:

1. $S = [(p1_sufficient, p2_insufficient, not(free_m1), not(free_m2)), \dots, (p1_insufficient, p2_insufficient, free_m1, not(free_m2)), \dots]$ in other terms the states are all possible combinations of four boolean variables: $|S| = 2^4$.

2. $s_0 = [p1_insufficient, p2_insufficient, free_m1, not(free_m2)]$ is the initial state

3. $S_G = [p1_sufficient, p2_sufficient, free_m1, free_m2]$ is the goal state.

4. $A = [start_watering, after_watering]$ is the set of actions.

5. $\gamma = [(p1_insufficient, p2_insufficient, free_m1, not(free_m2)) \bullet (start_watering\ p1\ m1) \rightarrow (p1_insufficient, p2_insufficient, not(free_m1), not(free_m2))]$ is the transition function. With the transition function it is possible to achieve an multiple intermediate state which facilitates the progression to the goal state. For this example, to achieve the goal state, the set of actions $A = [start_watering\ p1\ m1, after_watering\ p1\ m1, after_watering\ p2\ m2, start_watering\ p2\ m2, after_watering\ p2\ m2]$

In addition to the watering system, the lighting system of the garden is also planned to enhance the growth of the plants in case of low sunlight. The current luminescence level of the surrounding, as well

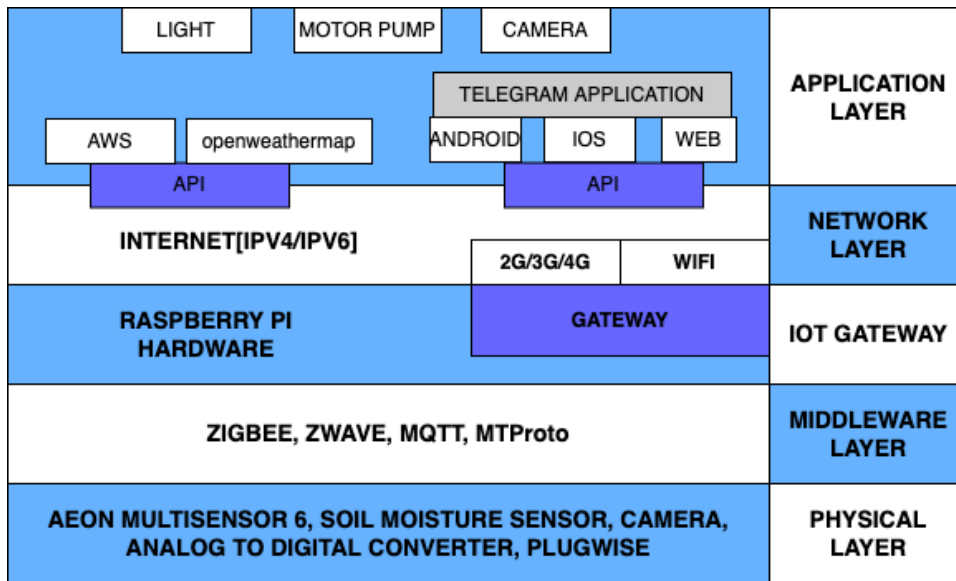


Figure 1. Smart Garden System Architecture

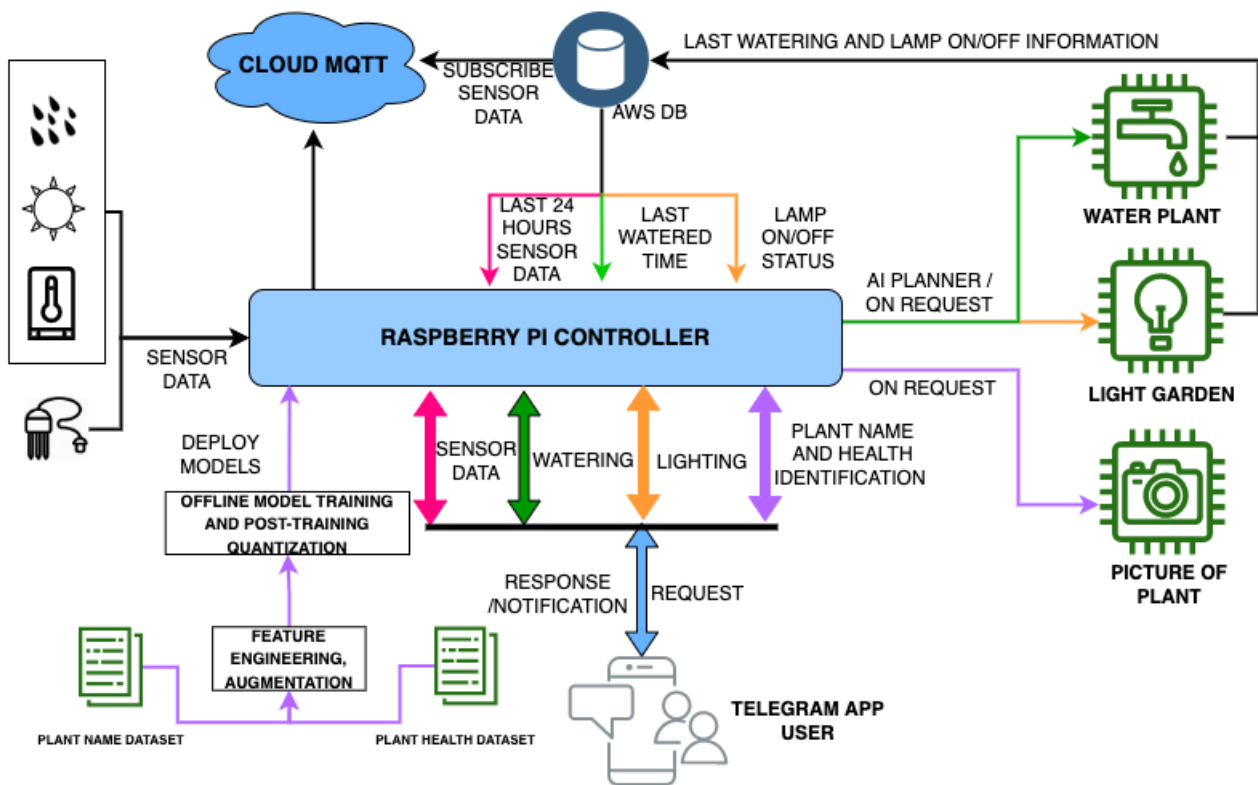


Figure 2. Implementation diagram

as the current state of both LED grow lights (on/off), are considered, and the purpose is to equalize the needed and present sun luminescence levels. This can be accomplished by turning on and off the lamps specified in the domain file. The AI planner remains active for 14 hours everyday. Here as well, the domain

as well as the problem files are created automatically and FF planner is used to generating the plan.

4.3. CNN Based Plant Name and Health Identification:

Convolutional Neural Network (CNN) [3] based architecture has been developed to allow the user to know the plant name and plant health. In this prototype, a total of 6 plant classes, i.e. *rose*, *angelonia*, *spider plant*, *orchids*, *cactus*, and *palm tree* are used to train the model. Plant and flower images are collected from Global Biodiversity Information Facility [21], as well as web scraping.

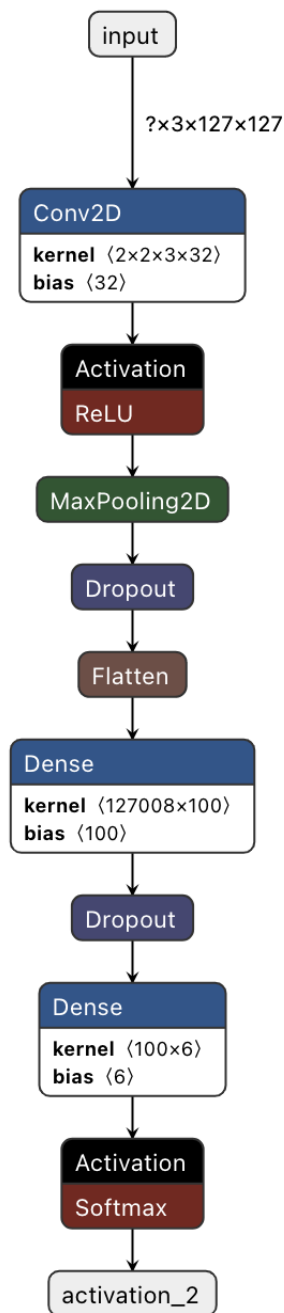


Figure 3. Architecture of Plant Identification model

Figure 3 shows the deep learning model architecture, where *CNN2D* layer is used before *relu* activation function and *MaxPooling* layer. Finally, 2 dense layers, followed by *softmax* activation function are implemented to classify the plants into 6 classes. To reduce overfitting [22], dropout of 30% is used in between the layers.

The model is trained extensively and 95% validation accuracy and 0.02% loss are achieved. Finally, the model size is reduced by post-training quantisation using TensorFlow Lite Converter, and the generated model is then deployed in the Raspberry pi.

For plant health identification, the aim is to detect if the plant is healthy or not. *Dry leaf*, *black-spot*, *rust*, and *fire-blight* conditions are considered non-healthy plant conditions. The data is collected from the internet using web-scraping and the image augmentation technique [23] is used to increase the amount of image data. Finally, a two-class classification (healthy/non-healthy) problem is solved. The architecture is identical to that illustrated in Figure 3, except in the final layer, *Binary Cross Entropy* activation function is employed to determine whether the plant is healthy or not. After rigorous training, we gained validation accuracy of 98% and validation loss of 0.017%.

4.4. Telegram Bot application

Telegram Bot are special accounts that serve as an interface for code running on the pi [17]. Different functionalities offered by the bot are noted below:

1. **Take picture:** The mobile phone is used as a webcam of the pi to capture a glimpse of the garden.
2. **Request for picture:** Upon user request, the integrated camera captures a picture of the garden and sends it to the Telegram application of the user.
3. **Water plants:** The last watered time of all the plants in the database is checked. The AI planner takes the moisture threshold of the plant into account and starts the water pumps if the plant is watered less than a specified time. After the watering is finished, the latest watered time is updated in the database.
4. **Know Your Plant and Plant Health:** When a user requests plant identification online, the CNN-based plant's name and health identification system is triggered, and the user is given the plant's name, picture, as well as its health status.
5. **LED growth light on/off :** The light is turned on/off using another AI planner, depending on the present status of the light and current

luminescence level. The changed status of the lights is updated in the database.

6. **Subscribe to Garden Status** : The user is provided with the graphical representation of the temperature and relative humidity during the day. This data is obtained from the database which has subscribed to the cloudMQTT.
7. **Local Weather** : This functionality uses openweathermap [24] to get the current weather condition in the locality of the garden.
8. **Add/Delete a Plant**: The maximum temperature withstanding capability, humidity required for sustainability and the moisture threshold of new plant could be inserted into the present plant database through this function. The function of the delete plant option is self explanatory.

5. Experiments and Results

The experimental setup consists of two plants, i.e. angelonia plant and spider plant. Angelonia is a summer plant that lives in moderate temperatures, and the preferred Substrate Moisture Content (SMC) for this plant is around 40% [25]. The spider plant is a tropical plant, which requires 40%-60% of soil moisture and prefers a temperature between 13°C to 27°C [26].

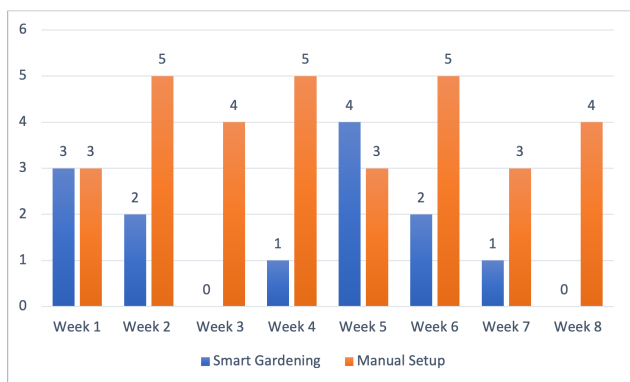


Figure 4. Plot of no. of manual intervention required for smart gardening vs the manual setup

For this experiment, two similar-sized setups were built with the aforementioned plants of similar size. One setup was fully integrated with the proposed gardening system, i.e. the setup is equipped with the raspberry pi based hardware setup with a camera, 2 LED growth lights, and 2 water pumps. The other setup just had 2 LED lights and 2 water pumps, a human attendant must be present physically to turn on the light or water the plants based on the sensor readings.

Both the setups were separately maintained in a similar environment for 8 weeks, and the number of times each setup (one or both the plants), needed

manual intervention, was noted down. Figure 4 shows the plot for number of physical interventions per week required by the proposed setup vs. the number of physical interventions required per week in the manual setup for a period of 8 weeks. The manual setup required average 4 interventions every week, mainly due to watering and changing the position of the plant tub for sufficient sunlight. With the smart gardening setup, this number was reduced to 59.37%. The setup was required to provide sufficient light in case of bad daylight conditions, as 1 LED grow light can provide approximately 1000lx only. Week 5 required some additional attention because of a technical error due to a loose circuit connection.

6. Conclusions and Outlook

A rational gardening system prototype is proposed in this paper, which can control the water and light supply to plants. The system boasts of efficient resource usage and low waste. The deep-learning algorithm has effectively managed the plant health-care department. Along with this, the garden can connect people from different parts of the globe, which is one of its unique selling propositions.

One of the primary limitations of this project was the lack of a mobile robotic arm that could take images of the garden from various angles or irrigate several plants with a single motor. Further extension on this project would be to connect different gardens located physically apart, "Connecting gardens for optimized learning". This project is reproducible in its entirety [27].

References

- [1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of things for smart cities. *IEEE Internet of Things Journal*, pages 22–32, Feb 2014.
- [2] J. Hoffmann and B. Nebel. The ff planning system: Fast plan generation through heuristic search. In *Journal of Artificial Intelligence Research 14 (2001) 253-302*, May 2001.
- [3] S. Albawi, T. A. Mohammed, and S. Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, Aug 2017.
- [4] Smart city technology: It's all about the internet of things. <https://www.information-age.com/smart-city-technology-123473905/>. Accessed: 2022-07-15.
- [5] Eve aqua: Smart water controller. https://www.evehome.com/en/eve-aqua?utm_source=Aqua&utm_medium=ApplePage&utm_campaign=Apple, 2022. Accessed: 2022-07-15.
- [6] Cloudrain: Your smart garden integration. <https://cloudrain.com/>, 2022. Accessed: 2022-07-15.
- [7] Hydrowise. <https://www.hydrowise.com/>, 2022. Accessed: 2022-07-15.

- [8] Nishath Shireen and Banita Devi. Smart garden management system. 2018.
- [9] Jeonghwan Hwang, Changsun Shin, and Hyun Yoe. A wireless sensor network-based ubiquitous paprika growth management system. *Sensors*, 10(12):11566–11589, 2010.
- [10] Rudy Gunawan, Tegas Andhika, S –, and Fadil Hibatulloh. Monitoring system for soil moisture, temperature, ph and automatic watering of tomato plants based on internet of things. *Telekontran : Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, 2019.
- [11] Greeniq. greeniq, 2017. <https://greeniq.com/>, 2018. Accessed: 2022-07-15.
- [12] Norakmar binti Arbain Sulaiman and Muhamad Dan Darrawi bin Sadli. An iot-based smart garden with weather station system. In *2019 IEEE 9th Symposium on Computer Applications Industrial Electronics (ISCAIE)*, pages 38–43, 2019.
- [13] C. Yu. Cloudthings: a common architecture for integrating the internet of things with cloud computing. In *International Conference on Computer Supported Cooperative Work in Design*, 2013.
- [14] Behrang Fouladi and Sahand Ghanoun. Security evaluation of the z-wave wireless protocol. 2013.
- [15] U. Hunkeler, H. L. Truong, and A. Stanford-Clark. Mqtt-s - a publish/subscribe protocol for wireless sensor networks. In *COMSWARE '08*, Jan 2008.
- [16] S. Narula, A. Jain, and Prachi. Cloud computing security: Amazon web service. In *2015 Fifth International Conference on Advanced Computing Communication Technologies*, pages 501–505, Feb 2015.
- [17] python-telegram-bot. <https://python-telegram-bot.readthedocs.io/en/stable/>, 2022. Accessed: 2022-07-15.
- [18] Domoticz: Control at your finger tips. <https://domoticz.com/>, 2022. Accessed: 2022-07-15.
- [19] Daniel S. Weld. Recent advances in ai planning. *AI Magazine*, 20(2):93, Jun. 1999.
- [20] J. Hoffmann and B. Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, may 2001.
- [21] Gbif | global biodiversity information facility. <https://www.gbif.org/>, 2022. Accessed: 2022-07-15.
- [22] Igor V. Tetko, David J. Livingstone, and Alexander I. Luik. Neural network studies. 1. comparison of overfitting and overtraining. *Journal of Chemical Information and Computer Sciences*, 35(5):826–833, Sep 1995.
- [23] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, Jul 2019.
- [24] Weather api. <https://openweathermap.org/api>, 2022. Accessed: 2022-07-15.
- [25] Alison Bingham Jacobson, Terri W. Starman, and Leonardo Lombardini. Substrate moisture content effects on growth and shelf life of angelonia angustifolia. *HortScience horts*, 50(2):272 – 278, 2015.
- [26] <https://www.gardenerreport.com/must-know-tips-for-watering-spider-plants/>.
- [27] Github project link. <https://github.com/Niloy-Chakraborty/Smart-Gardening>.