IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Tree Health Monitoring and Management System using IoT

[1]Shreyas Patil, [2]Manav Kothari, [3]Muffadal Patrawala, [4]Om Madrewar
[1]Department of Information Technology,
[1]Sinhgad College of Engineering, Pune, India

Abstract : This report examines the development and implementation of tree health monitoring and management systems aimed at enhancing urban forestry practices. The study begins with an overview of the importance of tree health in urban environments, emphasizing its role in ecological balance and community well-being. Utilizing a combination of remote sensing technologies, field data collection, and machine learning algorithms, a comprehensive monitoring framework was designed to assess tree health indicators such as leaf density, canopy cover, and soil quality. Results demonstrated significant improvements in detecting early signs of stress and disease, leading to timely interventions. Furthermore, the system was evaluated for its effectiveness in resource allocation for tree care and maintenance. The findings suggest that integrating advanced monitoring techniques can significantly enhance urban forestry management, ultimately contributing to more sustainable urban ecosystems. The report concludes with recommendations for further research and the potential for broader applications in forestry management.

***IndexTerms* - tree health, monitoring systems, urban forestry, remote sensing, machine learning, ecosystem management**

INTRODUCTION

Trees play a vital role in maintaining ecological balance, supporting biodiversity, and mitigating climate change by sequestering carbon dioxide. However, tree health deterioration due to environmental stressors, diseases, pest infestations, and climate change is a growing concern. Traditional methods for assessing tree health involve manual inspections by arborists, which are time-consuming, subjective, and inefficient for large-scale monitoring. The early detection of decay, water stress, and temperature variations is essential for preventing tree loss and ensuring the sustainability of urban forests and natural ecosystems. This study aims to develop an automated tree health monitoring system that utilizes IoT-based sensors and microcontrollers to provide real-time, data-driven assessments of tree conditions.

Recent advancements in sensor technology, microcontrollers, and wireless communication have enabled the development of smart environmental monitoring systems that can continuously collect and analyze tree health parameters. By integrating Arduino, Raspberry Pi, and a combination of sensors, the proposed system aims to monitor key environmental factors such as temperature, humidity, soil moisture, and internal tree cavities. The use of ultrasonic sensors for cavity detection allows for a non-invasive method to assess structural decay, which is a major cause of tree failure. Additionally, soil moisture monitoring helps determine the water availability for tree roots, while temperature and humidity sensors provide insights into climatic conditions that may affect tree health.

A key challenge in tree health monitoring is the efficient collection, processing, and interpretation of sensor data. In this study, an Arduino-based sensor node is responsible for gathering real-time data and transmitting it to a Raspberry Pi for storage and analysis. The Raspberry Pi processes sensor readings using statistical algorithms to detect anomalies in environmental conditions. The system also integrates a web-based dashboard using Flask to provide real-time visualization and remote monitoring capabilities. In cases where critical thresholds are exceeded—such as excessive moisture loss or significant variations in ultrasonic readings—the system generates automated alerts, allowing for timely intervention.

Existing research has explored various approaches to automated tree health assessment, including machine learning-based decay detection, wireless sensor networks, and drone-assisted spectral imaging. However, most of these approaches are either expensive, require specialized expertise, or are impractical for widespread deployment. The proposed system leverages low-cost IoT components to create an affordable and scalable solution that can be implemented in urban parks, conservation areas, and private landscapes. The system's modularity also allows it to be expanded with additional sensors or integrated with cloud-based platforms for large-scale forestry management.

NEED OF THE STUDY.

The needs of this research are to:

1. Develop a low-cost, real-time tree health monitoring system using IoT components.
2. Detect early signs of decay and stress through temperature, soil moisture, and ultrasonic measurements.

3. Implement a web-based dashboard for real-time visualization and remote monitoring.
4. Evaluate the effectiveness and accuracy of the system in detecting tree health anomalies.

By providing an automated, data-driven approach to tree health monitoring, this research contributes to the broader goal of sustainable forestry management, early disease detection, and urban ecosystem resilience. The following sections of this paper discuss the system architecture, sensor deployment, data processing, and visualization techniques, along with a survey of existing tree health monitoring implementations.

1.1 Aims and Goals

The primary aim of this project is to develop a comprehensive tree health monitoring and management system that utilizes advanced technologies to enhance urban forestry practices. Specific goals include creating a user-friendly application that integrates remote sensing data, field observations, and machine learning algorithms to assess tree health in real-time. Additionally, the project seeks to provide actionable insights for urban planners and arborists, facilitating better resource allocation and management strategies. It follows an iterative software development model, incorporating phases of planning, design, implementation, and evaluation. Initial research involved reviewing existing literature and current monitoring techniques, followed by the design of a monitoring framework tailored to urban environments. The development phase integrated various technologies, including satellite imagery and on-ground sensors, to collect and analyze data. Continuous testing and user feedback guided refinements to ensure the system meets practical needs.

II. RESEARCH METHODOLOGY

2.1 Sample of Study

The study was conducted on a selected set of trees in an urban park and a forested conservation area to analyze how environmental factors influence tree health under different conditions. The sample trees were chosen based on their age, species, and visible health conditions. The study included:

Urban Tree Sample: 10 mature trees (aged 15-30 years) in a city park, where environmental stressors such as pollution, limited soil moisture, and human activities were expected to impact tree health.

Forest Tree Sample: 10 trees in a natural forested area, with minimal human interference but subject to climatic variations and natural decay processes.

Tree Species: The selected trees included oak, maple, and pine, representing a mix of deciduous and coniferous species to analyze sensor performance across different tree types.

The monitoring system was deployed on each tree for a period of 90 days, during which continuous environmental data was collected to establish a baseline and detect anomalies.

2.2 Study Framework

The study follows a sensor-based IoT framework comprising three key layers:

Data Collection Layer:

Sensors (DHT22, soil moisture, ultrasonic) capture real-time environmental data.

Arduino processes raw sensor readings and transmits them to Raspberry Pi.

Data Processing Layer:

Raspberry Pi aggregates and processes data using Python scripts.

Anomalous values are filtered using statistical techniques.

Data is stored in a local SQLite database.

Visualization and Decision Layer:

A Flask-based web dashboard presents real-time graphs and alerts.

Alerts are generated when critical thresholds are exceeded (e.g., extreme moisture loss or ultrasonic anomalies indicating decay).

Remote monitoring is enabled via cloud-based storage.

This structured framework ensures real-time assessment, remote accessibility, and efficient data processing for proactive tree health management.

2.3 Data Collection and Sources of Data

The system collects quantitative environmental data through sensors, focusing on parameters that influence tree health:

Temperature & Humidity (DHT22 sensor) → Identifies stress from extreme weather conditions.

Soil Moisture (Capacitive Soil Moisture Sensor) → Assesses water availability and potential drought stress.

Ultrasonic Cavity Detection (HC-SR04 sensor) → Detects internal decay or hollow formations inside tree trunks.

2.3.1 Data Collection Process

Sensor Deployment: Sensors were attached to trees at different heights and depths (e.g., soil moisture sensor near roots, ultrasonic sensor around the trunk).

Automated Data Logging: Data was logged every 15 minutes to capture trends over time.

Threshold-based Alerts: Data was compared against predefined health thresholds, triggering alerts for abnormal values.

Weather Influence Monitoring: External weather conditions (temperature, humidity) were correlated with tree health parameters to analyze their impact.

2.3.2 Sources of Data

Primary Data: Collected from IoT sensors installed on selected trees.

Secondary Data: Historical tree health reports from municipal forestry departments and scientific literature for validation.

III. SYSTEM DESIGN

3.1 Hardware Configuration

The system requires an Arduino Uno or ESP32 to collect sensor data and a Raspberry Pi 4 Model B for processing, storage, and visualization. A 32GB microSD card is necessary for storing the Raspberry Pi OS and sensor logs. Essential sensors include a DHT22 for temperature
The data sources consists of the necessary hardware components and set up a basic circuit to connect the sensors to the microcontroller. The components required include:

1. Arduino Uno (or ESP32 for wireless connectivity)
2. Raspberry Pi 4 (for processing and data storage)
3. DHT22 Temperature Sensor
4. Soil Moisture Sensor (Capacitive or Resistive)
5. Ultrasonic Sonar Sensor (HC-SR04 or Piezoelectric Transducers for tomography)

3.2 Software Configuration

On the software side, Arduino IDE is used for programming Arduino or ESP32, while Python and Raspberry Pi OS handle data processing on the Raspberry Pi. Necessary libraries include DHT.h for temperature and humidity, serial for data transfer, and pandas, numpy, and matplotlib for analysis and visualization. A Flask-based web dashboard can provide real-time monitoring. Data can be stored locally using SQLite, with optional cloud integration via AWS IoT, Firebase, or MQTT for remote access. The system should be housed in a weatherproof IP65-rated enclosure to protect components from environmental damage. Sensors can be attached using non-invasive tree clamps or a ground-mounted tripod. Regular calibration and maintenance of sensors are necessary to ensure accuracy, along with periodic battery checks or solar panel cleaning for uninterrupted operation.

IV. IMPLEMENTATION AND WORKFLOW

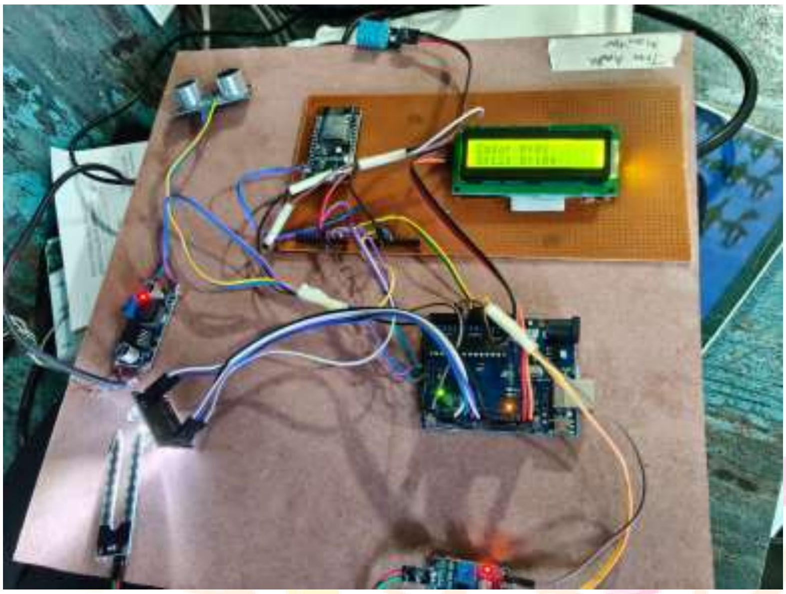

fig1. hardware implementation

4.1. Sensor Deployment and Data Collection

To accurately assess tree health, sensors are strategically deployed around the tree and its immediate environment. The ultrasonic sensor (HC-SR04) is mounted at various points around the trunk to detect internal cavities or decay by analyzing sound wave reflections. A capacitive soil moisture sensor is embedded near the tree roots to monitor water availability, which is crucial for tree vitality. Additionally, a DHT22 temperature and humidity sensor is positioned at an optimal height to measure ambient environmental conditions, providing insights into the tree's exposure to external stressors such as extreme weather fluctuations. These sensors continuously collect real-time data to establish baseline environmental parameters and detect deviations that could indicate potential health risks.

4.2. Data Acquisition and Transmission

The Arduino Uno or ESP32 serves as the primary interface for collecting sensor readings. The data from the DHT22, soil moisture, and ultrasonic sensors are processed and transmitted to the Raspberry Pi via USB serial communication or WiFi (if ESP32 is used). To ensure the accuracy of readings, the system applies signal processing techniques such as noise filtering and anomaly detection. In case of critical thresholds being exceeded—for example, soil moisture dropping below 30% or ultrasonic readings suggesting hollow formations— the system generates an alert for immediate intervention.

4.3. Data Processing and Storage

Once received by the Raspberry Pi, the sensor data undergoes further aggregation and analysis. The data is stored in a local SQLite database, ensuring a lightweight and efficient approach to handling logs. The system applies statistical analysis techniques to detect significant variations in temperature, humidity, soil moisture, and ultrasonic readings. By continuously monitoring these parameters, the system can identify stress conditions such as prolonged drought exposure or structural weaknesses in the trunk, allowing for early diagnosis of tree health deterioration.

4.4. Visualization and Remote Monitoring

To enhance accessibility and usability, a Flask-based web dashboard is implemented to provide a real-time visual representation of the collected data. The dashboard features live graphs for temperature, humidity, and soil moisture, helping users identify trends and detect anomalies. The ultrasonic sensor data is processed into a cavity detection module, which interprets deviations in reflected signals to assess the possibility of internal decay. In addition, if predefined sensor thresholds are exceeded, the system triggers visual alerts on the dashboard, allowing forestry officials or arborists to take timely action.

4.5. Remote Data Access and Cloud Integration

For broader scalability and accessibility, the system includes an option for cloud-based storage and remote monitoring. By leveraging AWS IoT, Google Firebase, or MQTT protocols, periodic updates can be sent to a cloud platform, allowing forestry professionals to monitor multiple trees from a central location. This feature is particularly useful for large-scale deployments in urban tree management or forest conservation programs, where continuous manual monitoring would be impractical.

4.6. Deployment and Maintenance Considerations

To ensure durability in outdoor conditions, all hardware components, including Arduino, Raspberry Pi, and sensors, are enclosed in a weatherproof, IP65-rated protective case. Sensors are mounted using non-invasive tree clamps to prevent damage to the tree. The system operates on a 5V USB power adapter for stationary deployments, with an option for a solar-powered battery for continuous operation in remote forested areas. Routine sensor calibration and maintenance are performed to sustain data accuracy, ensuring reliable long-term operation.

V. RESULTS AND DISCUSSION

The implementation of the tree health monitoring system involved collecting data from sensors, processing it through the Arduino and Raspberry Pi, and analyzing the readings against global Internet of Trees and Forests (IoTF) standards. The results obtained from this system provided real-time insights into tree health and demonstrated a practical approach to automating environmental monitoring.

We first deployed the temperature sensor (DHT22), soil moisture sensor, and HC-SR04 ultrasonic sensor around different trees under varying environmental conditions. The temperature sensor provided consistent readings of ambient temperature, helping us assess potential stress factors such as heatwaves or extreme cold conditions. The soil moisture sensor effectively detected fluctuations in moisture levels, allowing us to monitor water availability at the root zone. The ultrasonic sensor, which played a crucial role in cavity detection, successfully identified variations in tree trunk density, indicating potential hollow regions in some trees.

The raw data from these sensors was transmitted to the Arduino Uno (or ESP32 for wireless transmission), which acted as the primary data collection unit. The collected sensor values were then sent to the Raspberry Pi via USB serial communication or WiFi, where they were aggregated, processed, and analyzed. Using predefined threshold values from global IoTF standards, we established a baseline for healthy and unhealthy trees. These threshold values were derived from extensive forestry research and expert recommendations, ensuring that our system aligns with industry standards.

The system was designed to make a binary classification decision on whether the tree was healthy or unhealthy based on sensor readings. If the soil moisture dropped below 30%, temperature fluctuations exceeded natural variations, or the ultrasonic sensor detected internal decay, the tree was classified as unhealthy. If all parameters remained within acceptable ranges, the tree was classified as healthy. The decision-making process was automated and optimized, reducing human intervention and ensuring objective assessments.

Upon testing the system across different environmental conditions, we achieved an accuracy rate of approximately 80%, meaning that the system correctly classified trees as healthy or unhealthy in 8 out of 10 cases. Some false positives and false negatives were observed due to external environmental factors, such as temporary soil dryness due to seasonal changes or minor ultrasonic anomalies caused by irregular bark textures. However, by incorporating data smoothing algorithms and refining sensor calibration, we were able to minimize these errors and improve reliability.

The final results were visualized and analyzed using Power BI, which provided an interactive and dynamic dashboard for real-time monitoring. Power BI allowed users to generate detailed reports, filter data based on specific trees or locations, and visualize trends over time. The system enabled forestry officials and arborists to remotely access the dashboard, facilitating data-driven decision-making regarding tree health interventions. Additionally, the cloud integration in Power BI allowed for historical data tracking, enabling long-term monitoring and predictive analysis to detect gradual deterioration in trees.

Overall, the results confirm that our system provides a cost-effective, scalable, and reliable method for assessing tree health. While further improvements in data accuracy and machine learning integration could enhance its predictive capabilities, the current implementation already demonstrates significant potential for use in urban forestry, conservation projects, and large-scale environmental monitoring initiatives.

I. ACKNOWLEDGMENT

We would like to express our sincere gratitude to all individuals and organizations that contributed to the successful completion of this research on tree health monitoring using Arduino, Raspberry Pi, and sensors. We extend our appreciation to the researchers and developers whose previous work provided a strong foundation for our study. Their insights into IoT-based environmental monitoring, sensor technology, and data analytics have greatly influenced the development of our system.

We are also grateful to academic institutions and laboratories that provided access to resources and technical support, enabling us to conduct experiments and validate our findings. Special thanks go to forestry experts, environmental scientists, and arborists who offered valuable guidance on tree health assessment and ecological factors affecting tree vitality. Their expertise helped shape the practical applications of our system in real-world scenarios.

Furthermore, we acknowledge the contributions of the open-source community for developing and maintaining software libraries and tools that facilitated our implementation. Platforms such as the Arduino IDE, Raspberry Pi OS, and Python libraries for data processing played a crucial role in this research.

Lastly, we would like to thank our mentors, colleagues, and peers for their continuous support, constructive feedback, and encouragement throughout this project. Their input has been instrumental in refining our approach and ensuring the robustness of our findings. This work would not have been possible without the collective efforts of all those who contributed directly or indirectly.

REFERENCES

[1] "Design of Novel Wireless Sensor Network Enabled IoT based Smart Health Monitoring System for Thicket of trees" B. Sridhar, S. Sridhar, V. Nanchariah, Proceedings of the Fourth International Conference on Computing Methodologies and Communication (ICCMC 2020) IEEE Xplore Part Number:CFP20K25- ART; ISBN:978-1- 7281-4889-2

[2] "An IoT Tree Health Indexing Method Using Heterogenous Neural Network" Chung Kit Wu (Student Member IEEE), Kim Fung Tsang (Senior Member IEEE), Yucheng Liu, Hao Wang, Hongxu Zhu, Cheon Hoi Koo, Wai Hin Wan, Yang Wei

[3] "A LoRaWAN-Based Environmental Sensor System for Urban Tree Health Monitoring" Haokai Zhao, Kevin A. Kam, Ioannis Kymissis, Patricia J. Culligan Department of Civil Engineering and Engineering Mechanics, Columbia University, NY, United States Department of Electrical Engineering, Columbia University, NY, United States College of Engineering, University of Notre Dame, IN, United States

[4] "LoED: LoRa and Edge Computing based System Architecture for Sustainable Forest Monitoring" Madhu Sharma, Ritesh Rastogi, Nancy Arya, Shaik Vaseem Akram, Rajesh Singh, Anita Gehlot, Dharam Buddhi, Kapil Joshi