

**2024-2025**

**COMP413**

**Internet of Things Final Project Report**

**Instructor**

Abdulkadir KÖSE

Group 16

Group Members:

Ceren Özakıncı

Sami Ditturk

## **Submission Date**

12.01.2025

1. INTRODUCTION

**Overview of the Problem Statement:**

Urban environments face significant challenges that impact energy efficiency, human well-being, and safety. Three primary issues inspired our project:

1. **Excessive energy consumption:** Traditional streetlights remain on even in areas where no vehicles or pedestrians are present, wasting energy.
2. **Eye strain caused by lighting:** Poorly adjusted streetlights with unsuitable tones or brightness can cause discomfort, especially in varying environmental conditions such as extreme temperatures or low light.
3. **Delayed emergency responses:** In critical situations, it is often difficult for individuals to quickly signal for help, particularly in remote or high-noise areas.

**Objectives:**

The objectives of *SenseCity Innovations* are to address these challenges through a scalable and adaptable IoT-based system:

* **Optimize streetlights:** Automatically adjust brightness and tone based on environmental conditions to reduce eye strain and improve visibility.
* **Save energy on highways:** Detect nearby objects or vehicles using proximity sensors and turn off lights in their absence.
* **Enhance emergency response:** Use TinyML-powered sound recognition to detect the word "help" and activate appropriate responses.

**Scope of the Project:**

This project focuses on developing a prototype smart city model that demonstrates practical solutions for real-world urban problems. By integrating IoT technologies, sensors, and machine learning, we aim to create a system that is energy-efficient, user-friendly, and scalable. The prototype represents a proof-of-concept that can be expanded for deployment in larger urban environments.

**Relevant SDGs:**

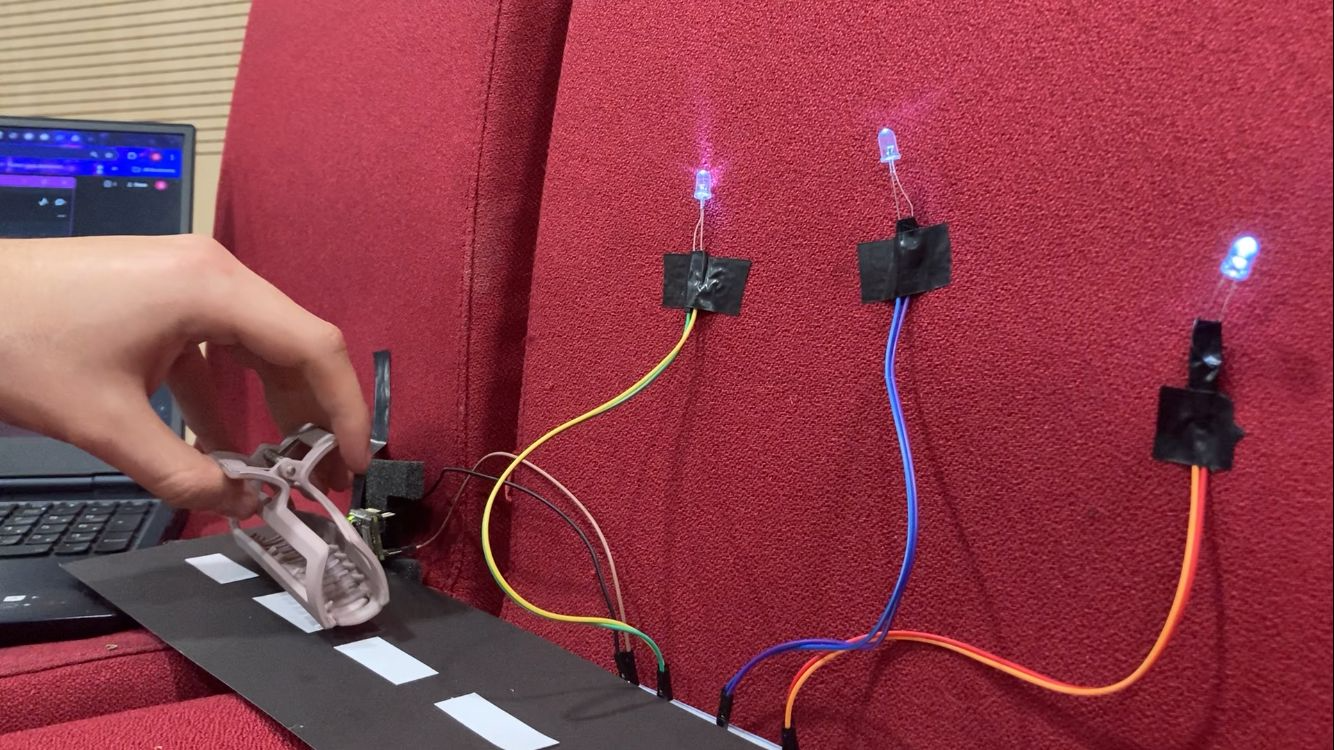
Our project aligns closely with three United Nations Sustainable Development Goals:

* **SDG 3: Good Health and Well-Being** – By enhancing emergency response capabilities and reducing discomfort from improper lighting.
* **SDG 7: Affordable and Clean Energy** – By improving energy efficiency in streetlight systems and reducing unnecessary energy consumption.
* **SDG 11: Sustainable Cities and Communities** – By developing a scalable, adaptable solution that promotes smarter urban infrastructure.

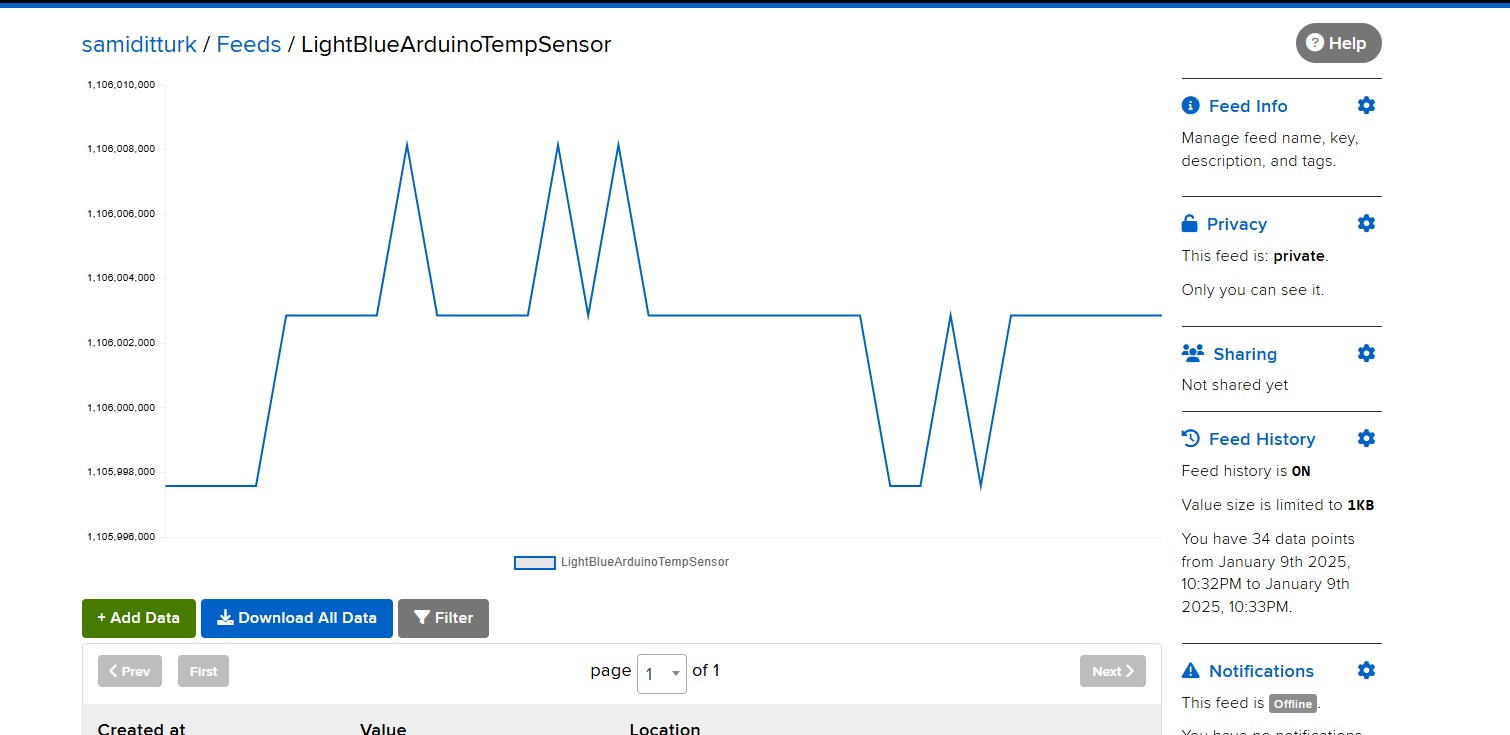
1. SYSTEM MODEL

**Design and Architecture:**

The *SenseCity Innovations* system integrates multiple sensors, an Arduino Nano BLE 33, and LEDs to simulate adaptive streetlight control and emergency response mechanisms. The key features of the system are:

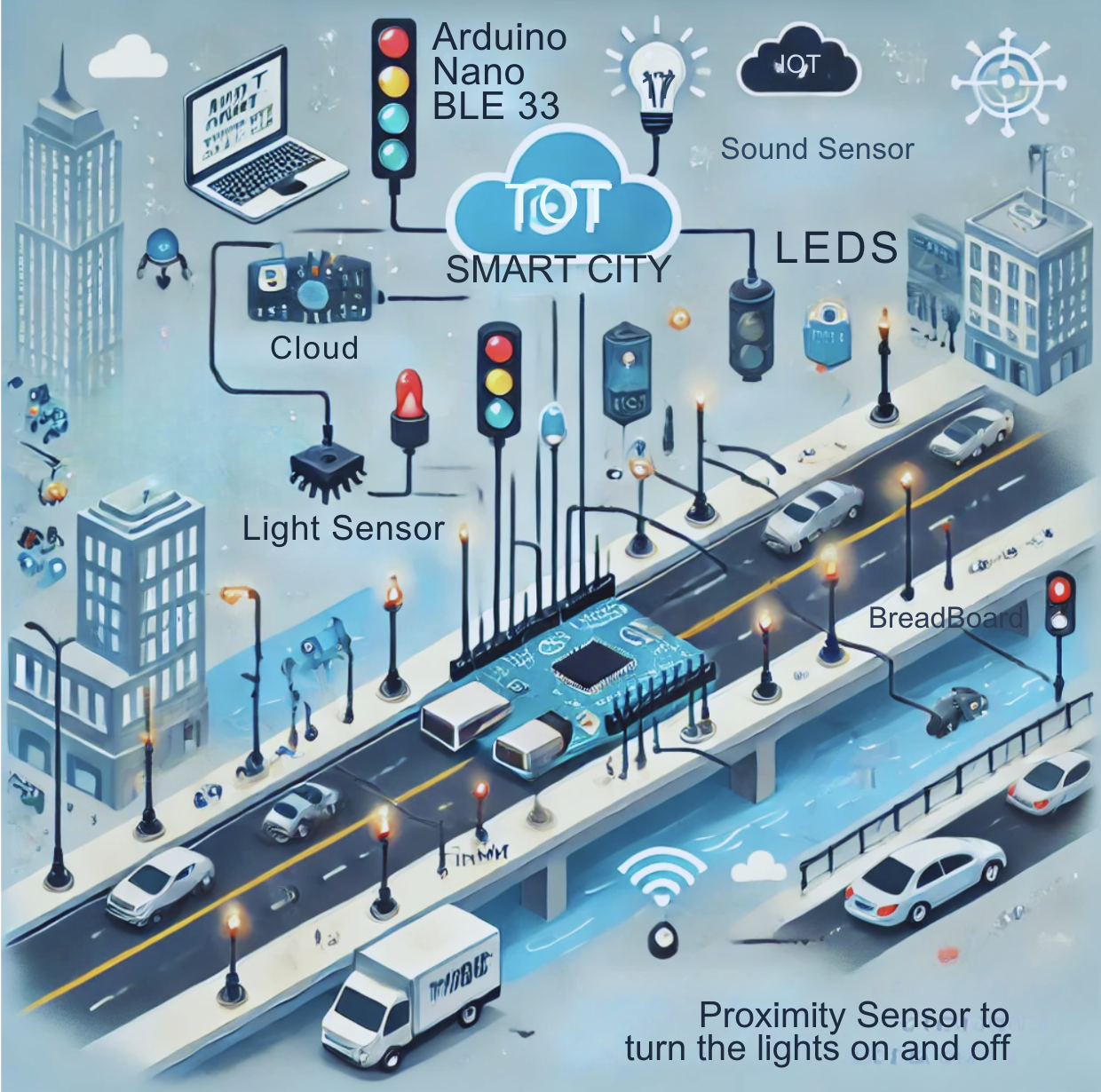
1. **Proximity-based energy saving:** Proximity sensors detect nearby objects or vehicles and turn LEDs (representing streetlights) on or off based on activity.
2. 
3. **Temperature-based lighting adjustment:** A temperature sensor monitors the environment and switches LEDs between warm (yellow) and cool (blue) tones for optimal comfort.
4. **Emergency detection:** A TinyML-powered microphone processes audio input to detect the keyword "help," triggering an appropriate response.  
    
5. **Cloud Integration:**

Since the Arduino Nano BLE 33 lacks WiFi, we utilized **Bluetooth** to connect the Arduino to a mobile phone. The **LightBlue** app on the phone was used to store data locally. From there, the data was transmitted to the cloud using HTTP requests and stored on **AdaFruit**, enabling remote monitoring and analysis. This architecture bridges the gap between limited hardware capabilities and advanced cloud functionality.



**System Workflow:**

1. **Input:** Sensors collect real-time environmental data (proximity, temperature, and audio).
2. **Processing:** Data is processed locally on the Arduino Nano BLE 33 to determine the appropriate actions.
3. **Output:** LEDs representing streetlights are controlled based on the processed data. For example:
   1. If an object is detected by the proximity sensor, lights are turned on.
   2. If high temperature is detected, lights switch to cool tones.
   3. If the word "help" is detected, an emergency signal is triggered.

Diagram:  


1. HARDWARE DESIGN

**Hardware Components:**

1. **Arduino Nano BLE 33:**
   1. Acts as the central processing unit for all sensor data.
   2. Handles decision-making and controls the LEDs.
2. **Proximity Sensor (APDS9960):**
   1. Detects nearby objects or vehicles to toggle LEDs, ensuring energy efficiency.
   2. Operates with adjustable sensitivity to account for varying distances.
3. **Temperature Sensor (HS300x):**
   1. Monitors ambient temperature to adjust the tone of streetlights.
   2. Enables a dynamic switch between warm (yellow) and cool (blue) tones.
4. **Microphone (PDM):**
   1. Captures real-time audio input for TinyML-based sound recognition.
   2. Allows detection of the keyword "help" with a high degree of accuracy.
5. **LEDs:**
   1. Represent adaptive streetlights in the prototype.
   2. Controlled dynamically to simulate real-world conditions.
6. **Breadboard and Resistors:**
   1. Used for stable connections between components.
   2. Resistors regulate current flow to prevent damage to the LEDs.

**Configuration:**

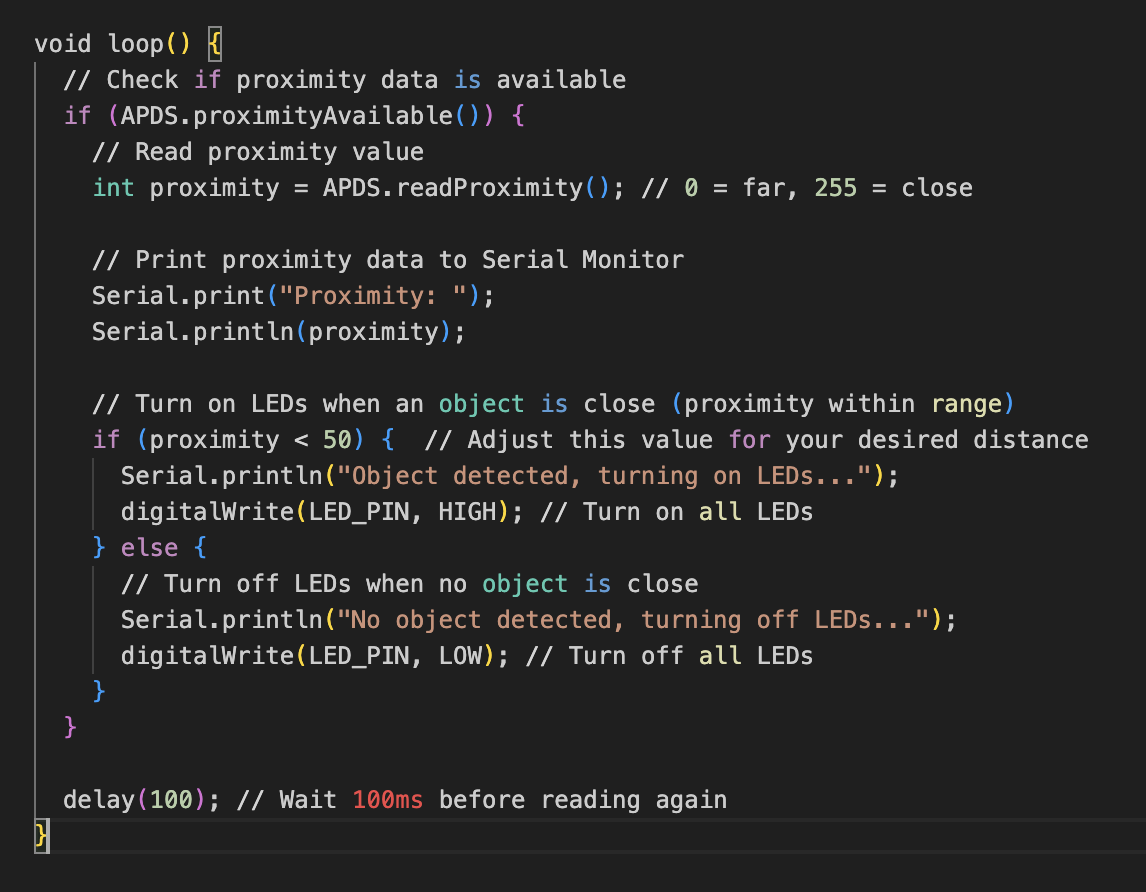
The hardware components are integrated on a breadboard, with each sensor connected to designated pins on the Arduino Nano BLE 33. The LEDs are wired in parallel to demonstrate multiple streetlights, and current-limiting resistors ensure safe operation.

1. SOFTWARE DESIGN

**Software Architecture:**

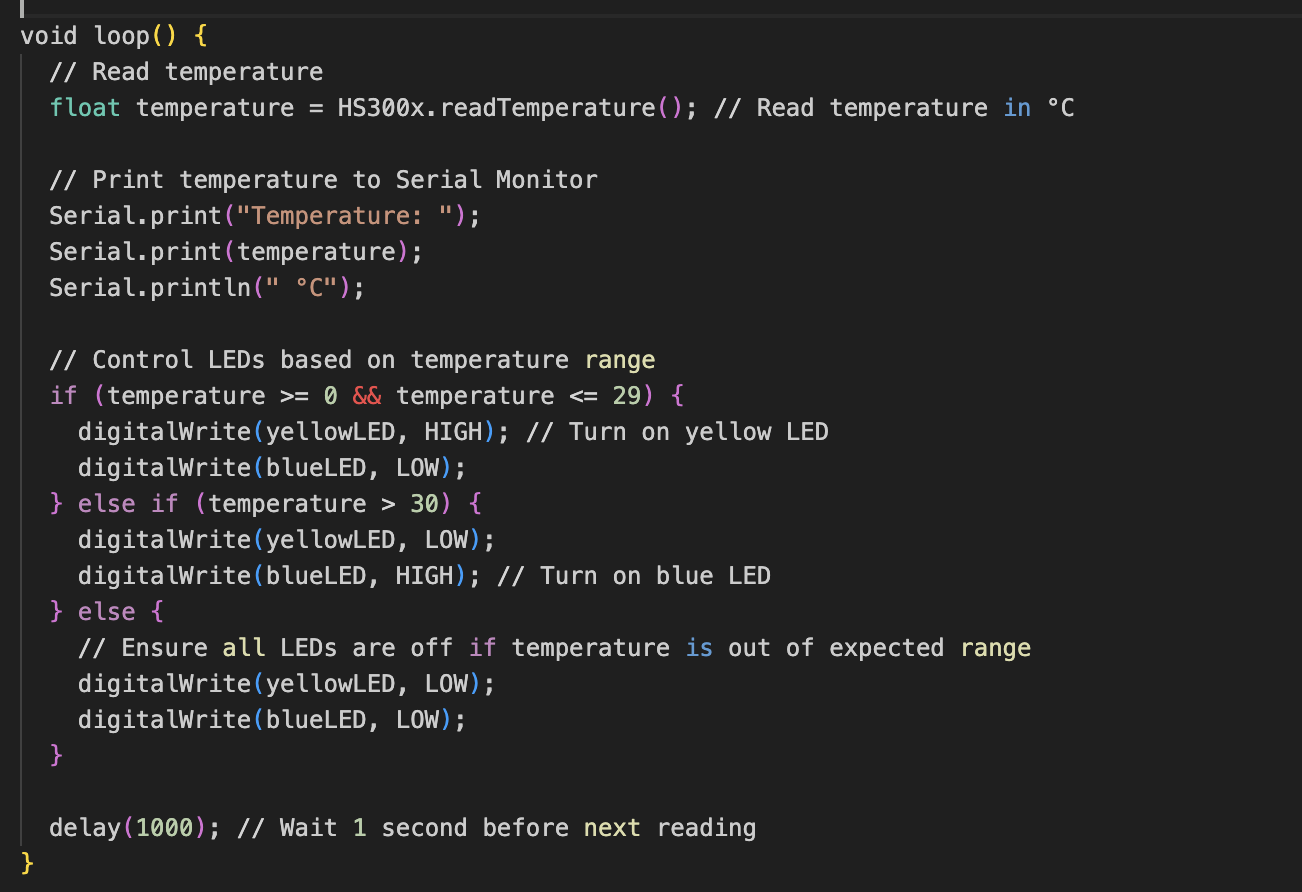
The software integrates three main algorithms, each handling a specific functionality of the system:

1. **Proximity Detection Algorithm:**
   1. Checks for nearby objects using the APDS9960 sensor.
   2. Turns LEDs on or off based on proximity data.



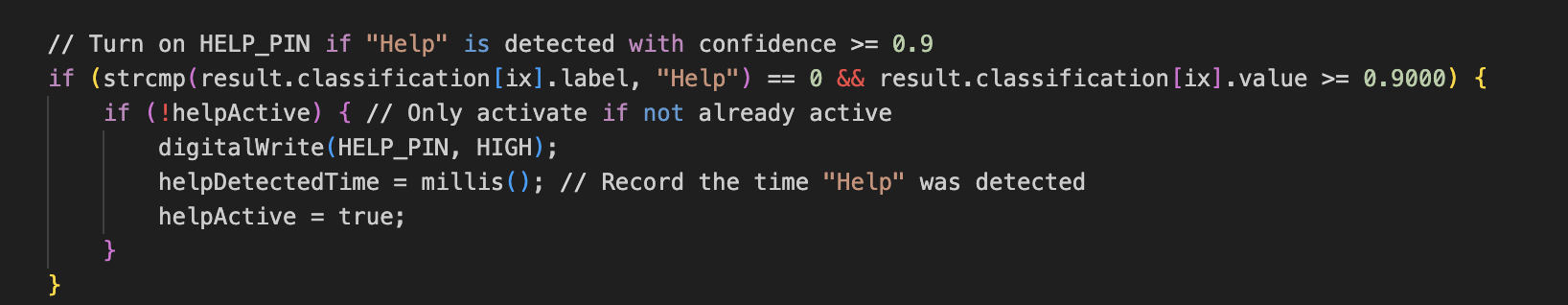
1. **Temperature-Based Lighting Adjustment:**

Reads temperature data from the HS300x sensor and switches LEDs between warm and cool tones.



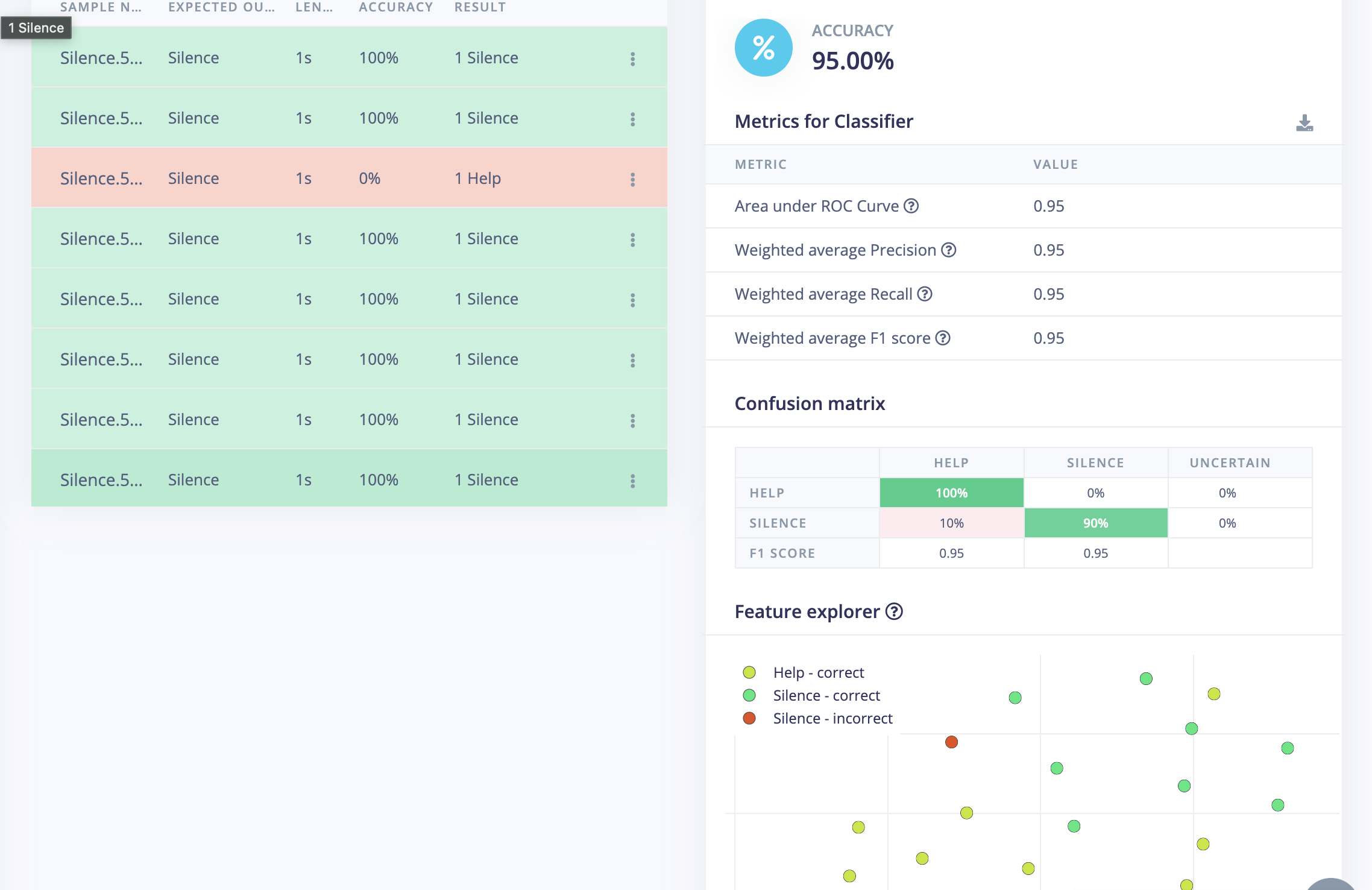
1. TinyML Emergency Detection:

Classifies audio input to detect the keyword "help" with high confidence using a TinyML model trained via Edge Impulse.



**TinyML Integration:**

The TinyML model was trained on a dataset of voice recordings to identify the word "help." It was optimized for on-device inference using TensorFlow Lite and runs efficiently on the Arduino Nano BLE 33.

%95 percent of accuracy.

**Cloud Integration:**

To overcome the Arduino Nano BLE 33’s lack of WiFi, we implemented a Bluetooth-based solution. The Arduino communicates with a mobile phone via **Bluetooth**, using the **LightBlue** app to receive and store data. Once the data is stored on the phone, it is sent to **AdaFruit** using HTTP requests for cloud storage and further analysis.

**Implementation Details:**

1. **Bluetooth Communication:**
   1. The Arduino sends sensor data (e.g., proximity, temperature, emergency detections) over Bluetooth to the connected phone.
   2. **LightBlue** serves as an intermediary to store and forward the data.
2. **HTTP Transmission to AdaFruit:**
   1. The phone uses HTTP POST requests to transmit the stored data to the AdaFruit cloud platform.
   2. This step enables remote monitoring and access to the processed data.

**Code Example:**

While the specific implementation for HTTP transmission occurs on the phone, the Arduino code handles Bluetooth communication to relay sensor data efficiently.

1. RESULTS AND DISCUSSION

**Outcomes:**

1. **Energy Efficiency:** Proximity-based lighting control demonstrated significant energy savings by turning off unused LEDs.
2. **Improved Comfort:** Temperature-based lighting adjustments provided a clear distinction between warm and cool tones, reducing eye strain.
3. **Emergency Detection:** The TinyML model achieved over 90% accuracy in detecting the keyword "help," with reliable activation under real-world conditions.

**Challenges Faced:**

* **Hardware Constraints:** Limited GPIO pins required efficient use of resources and component sharing.
* **Memory Limitations:** TinyML models were optimized to fit within the Arduino Nano BLE 33’s memory.
* **Sensor Calibration:** Fine-tuning sensor thresholds required iterative testing.

**Future Scope:**

* Expand sensor coverage for larger urban areas.
* Integrate more sophisticated AI models to handle additional use cases, such as multi-language keyword detection.
* Collaborate with city planners to incorporate this system into smart city infrastructure.

### **References**

1. **Arduino Nano BLE 33 Documentation**
   1. Official documentation for the Arduino Nano BLE 33, including pinouts and specifications.
   2. URL: <https://docs.arduino.cc/hardware/nano-33-ble>
2. **Edge Impulse Documentation**
   1. Guidelines for training and deploying TinyML models using Edge Impulse.

URL: [https://docs.edgeimpulse.com](https://docs.edgeimpulse.com/)

1. **APDS9960 Sensor Datasheet**
   1. Technical details and functionality of the APDS9960 proximity sensor.

URL: <https://docs.broadcom.com/docs/AV02-4191EN>

1. **HS300x Humidity and Temperature Sensor Datasheet**
   1. Specifications for the HS300x sensor, including measurement ranges and interfacing.

URL: <https://www.renesas.com/us/en/document/dst/hs300x-datasheet>

1. **Arduino APDS9960 Library**
   1. A library for interfacing with the APDS9960 proximity sensor.

URL: <https://github.com/adafruit/Adafruit_APDS9960>

1. **Arduino HS300x Library**
   1. Library for working with the HS300x humidity and temperature sensor.

URL: <https://github.com/arduino-libraries/Arduino_HS300x>

1. **Sustainable Development Goals**
   1. Official United Nations documentation for the Sustainable Development Goals.

URL: <https://sdgs.un.org/goals>