Reward-Based Smart Waste Collection: A Cornerstone of Smart City Development

Sanjukta Pal

I. Problems Identified:

Rising urbanization and population growth have exacerbated global waste management challenges, requiring innovative solutions to address inefficiencies and environmental impacts. The challenges are,

- **Improper Waste Segregation**: Inadequate segregation leads to resource wastage, excessive landfill use, and increased air pollution.
- **Inefficient Collection Systems**: Traditional waste collection schedules are often inflexible, resulting in unnecessary trips when bins are not full, inefficient routing, and underutilized resources. This increases fuel costs, labour expenses, and equipment wear.
- **Environmental Impact**: Inefficient waste collection practices contribute to increased CO_2 emissions and a larger carbon footprint due to non-optimized routes and excessive transportation.
- **Low Public Awareness**: Many citizens lack awareness or motivation to participate in proper waste disposal, limiting the effectiveness of waste management programs.
- **Lack of Real-Time Data**: Insufficient real-time data on bin fill levels and waste collection patterns prevents optimization, leading to inefficiencies and delays in the collection process.

II. Solutions:

A reward-based smart waste collection system, combining IoT (Long Range Wireless Area Network), AI, and machine learning, tackles urban waste management challenges effectively.

IoT-enabled smart bins equipped with sensors track fill levels and alert collection services when bins are nearing full capacity, preventing overflow and maintaining urban cleanliness.

LoRaWAN connectivity enables real-time data transmission on bin statuses, allowing for optimized collection routes that reduce fuel usage and CO_2 emissions. Smart bins equipped proper sensors can segregate wastes at the source, which may result reducing landfill, air pollution, and promoting recycling.

AI-powered algorithms dynamically adjust routes based on fill levels, saving resources and minimizing unnecessary trips. Machine learning analyses waste collection patterns, continuously improving system efficiency and adapting to evolving needs.

To engage citizens, the system integrates a gamification model, awarding points for correct waste disposal, which can be direct monitory profit or point based rewards depending on usage of bins through mobile app, thereby motivating responsible behaviour.

Additionally, real-time waste data aids urban planners by providing insights into waste generation patterns, optimizing bin placement, collection frequency, and helping in environmental impact measurement. This data-driven approach supports smarter urban planning, reduces landfill contributions, and promotes sustainable resource use.

III. Implementation details

This implementation details are the step wise procedure for project development.

1. Hardware Setup and Integration

A. **Select and Assemble Components for IoT Devices**:
   a. **Microcontroller**: Use an **Arduino MKR WAN 1300** or **ESP32**.
   b. **Ultrasonic Sensor**: HC-SR04 for waste level measurement.
   c. **GPS Module**: Use the **Neo-6M GPS** module for location tracking.
   d. **LoRaWAN Transceiver**: Install an RFM95W LoRa module.
   e. **Power Supply**: Connect a solar panel and a Li-ion battery for renewable energy support.
   f. **Note**: Connect the ultrasonic sensor to the microcontroller's trigger and echo pins, GPS to UART, and LoRa module to SPI.

B. **Install Arduino Libraries**:
   a. Download the necessary libraries from the Arduino Library Manager:
      i. **MKRWAN.h** (for LoRaWAN communication)

ii. **TinyGPS++** (for GPS data parsing)
iii. **NewPing** (for ultrasonic sensor readings)

C. **Upload Code to IoT Devices**:

a. Using Arduino IDE, upload the provided **Arduino code** to each IoT device. Ensure **LoRaWAN parameters (appEui and appKey)** are configured.

b. **Use code:** Arduino-based IoT Device.cpp

D. **Test Sensor and Data Transmission**:

a. Check if the ultrasonic sensor accurately measures the fill level.

b. Verify GPS functionality by comparing location data.

c. Ensure data transmission via LoRaWAN by connecting to a nearby LoRa gateway.

2. Communication Network Setup (LoRaWAN)

A. **Set Up LoRaWAN Gateway**:

a. Place LoRaWAN gateways strategically throughout the city for optimal coverage.

b. Configure each gateway to connect to a LoRaWAN network server, like **The Things Network (TTN)**.

B. **Configure Network Server**:

a. Use TTN to manage and monitor data from IoT devices.

b. Create **applications** within TTN to register and connect each IoT device.

C. **Test Device Communication with Gateway**:

a. Use TTN Console to ensure data packets are received from IoT devices.

b. Troubleshoot connectivity issues and monitor real-time data.

3. Cloud-Based Backend Server Development

A. **Install Node.js and Dependencies**:

a. Install required packages

B. **Set Up Backend Server**:

a. Choose a **Node.js with Express framework**.

b. **Use code:** Code for Backend Server.js

c. Install MongoDB on the cloud or use a hosted MongoDB service like **MongoDB Atlas**.

C. **Define Database Models**:

   a. Set up MongoDB schemas for waste bins and reward activities.

   b. **Use code:** Implementation Details of Database Schema.sql

D. **Create API Endpoints**:

   a. Create API routes for receiving waste data, fetching waste bins, and handling gamified activities.

E. **Deploy Backend on Cloud Server**:

   a. Use **AWS (**other back end cloud server like **Heroku, or DigitalOcean** may also be applied**)** to deploy the backend API server.

4. AI Module for Route Optimization and Waste Sorting

A. **Set Up Python Environment for AI Development**:

   a. Install required libraries

B. **Develop Route Optimization Code**:

   a. Use the **OR-Tools** library for route optimization.

   b. Use code: Code for Route Optimization.py

C. **Integrate AI Module with Backend Server**:

   a. Use the backend to receive optimized routes from the AI module and send them to the mobile app.

5. Mobile App Development for Users and Waste Collectors

A. **Set Up Development Environment**:

   a. Install **React Native** CLI, Node.js, and Android Studio or Xcode for mobile app development.

B. **Build Mobile App Functionality**:

   a. Implement features in React Native for:

      i. **Displaying optimized routes**

      ii. **Reporting full bins**

      iii. **Reward system** integration

   b. Use code: Code for Mobile App.ts

      Mobile App Integration for Reward system.ts

C. **Connect Mobile App with Backend**:
    a. Use **fetch API** to communicate with backend endpoints for bin data and reporting functionality.

D. **Test Mobile App on Simulators and Devices**:
    a. Run the app on Android/iOS simulators and perform field testing with IoT devices.

6. Implementation of Gamified Reward System

A. **Database Schema for Rewards**:
    a. Define tables in PostgreSQL for tracking user activities, points, levels, achievements, and rewards.
    b. Use code: [Implementation Details of Database Schema.sql]

B. **Backend API Endpoints for Rewards**:
    a. Develop backend endpoints in Node.js to manage user activities, points, achievements, and rewards redemption.

C. **Mobile App Integration**:
    a. In React Native, create views for users to view their points, achievements, and redeem rewards.
    b. **Use code:** [Backend API Endpoints for Reward system.js]

D. **Test and Deploy Reward System**:
    a. Test reward system functionality with users to ensure smooth point accumulation and redemption.

7. Final Testing and Deployment

A. **Test End-to-End Workflow**:
    a. Conduct full testing: IoT data collection, transmission, backend processing, AI optimization, and mobile app display.

B. **Deployment**:
    a. Deploy all components, monitor using analytics, and gather user feedback for improvements.

This systematic approach should smoothen implementation and maintenance of the IoT-based Waste Collection System.

IV. Conclusion:

This smart waste collection system provides a sustainable solution to urban waste management by using IoT, AI, and machine learning to streamline waste collection, reducing inefficiencies and minimizing environmental impact through smarter routing and resource use. This smart waste collection technology also reduces inefficiencies, and lower CO_2 emissions. Gamified rewards motivate citizens to practice proper waste disposal, fostering community involvement. Meanwhile, data insights help city planners make informed decisions for optimized bin placement and collection frequency. Ultimately, this solution encourages eco-friendly behaviour, improves urban cleanliness, and supports long-term environmental health, contributing to the sustainable development of cleaner, smarter cities.