**DigitalTwinX: Revolutionizing**

**Industry 5.0**

**Project Report: Digital Twin with BCF and IoT Integration for Facilities Management**

**Team members :**

**Amira merzougui**

**Khalil znazen**

**Manar afli**

**Submission date : 22/9/2024**

**Institution : INSAT**

**Table of Contents**

1. **Introduction  
   1.1 Project Overview and Scope  
   1.2 Objectives**
2. **System Architecture  
   2.1 Overview  
   2.2 System Diagram**
3. **Hardware Components  
   3.1 Microcontroller: ESP32-WROOM-32D  
   3.2 Sensors Used**
4. **Communication System  
   4.1 ESP-NOW Protocol  
   4.2 Firebase Integration  
   4.3 Real-Time Visualization: Google Analytics**
5. **AI Model for Anomaly Detection  
   5.1 Objective  
   5.2 AI Integration Pipeline  
   5.2.1 Data Collection and Preprocessing  
   5.2.2 Model Selection  
   5.2.3 Model Training  
   5.2.4 Model Deployment  
   5.3 Model Inference  
   5.4 Anomaly Detection and Alerts  
   5.5 Continuous Improvement**
6. **BCF Integration  
   6.1 BCF Format for Alerting**
7. **Results and Visualization  
   7.1 Firebase Real-Time Data Monitoring  
   7.2 AI Alerts and BCF Updates**
8. **Future Work  
   8.1 Scalability  
   8.2 Enhanced AI Models  
   8.3 Advanced Visualization**
9. **Suggestions**
10. **Conclusion**

**1. Introduction**

**Project Overview and scope:**

The challenge of optimizing facilities management in the context of complex infrastructure requires real-time, data-driven insights to enable predictive analysis and proactive management. This project aims to leverage the Building Collaboration Format (BCF) within digital twin technology to provide a comprehensive overview of infrastructure performance and conditions. The integration of real-time sensor data into BCF-georeferenced digital models is a key objective, offering enhanced decision-making processes and actionable insights for industry stakeholders.

**Objectives:**

* Enable real-time monitoring of infrastructure through sensor data.
* Predict anomalies such as overconsumption or potential failures.
* Automate the decision-making process through AI-powered analysis.
* Visualize infrastructure status and issues within the digital twin.

**2. System Architecture**

**Overview:**

* **Sensors**: Each room includes sensors for temperature and humidity, vibration, gas detection, water leaks, ambient light, occupancy, and intrusion detection.
* **Room ESP32s**: Each room is equipped with an ESP32 microcontroller to collect data from the sensors specific to that room.
* **ESP32 Gateway**: A central ESP32 gateway aggregates the data from the room ESP32s via the ESP-NOW communication protocol.
* **Firebase Database**: The gateway transmits the collected sensor data to a Firebase real-time database for storage and monitoring.
* **AI Anomaly Detection**: An AI model analyzes the data to detect anomalies or alerts (e.g., energy consumption irregularities).
* **BCF Integration**: Alerts and detected anomalies are sent to the Building Information Modeling (BIM) system via BCF for seamless updates to the digital twin.

System Diagram:

**3. Hardware Components**

Microcontroller:

* **ESP32-WROOM-32D:** Chosen for its robust Wi-Fi and Bluetooth capabilities, the ESP32-WROOM-32D is also compatible with the ESP-NOW protocol, allowing seamless communication with other ESP32 devices over long distances while maintaining low power consumption. Its affordability, small form factor, and ease of integration make it a cost-effective solution for large-scale IoT applications. Additionally, the ESP32’s dual-core processor offers efficient performance, ideal for handling multiple sensors and real-time data processing, further enhancing the system's reliability and scalability.

Sensors Used:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sensor | Purpose | |  | | --- | | Integration Points |  |  | | --- | |  | | Advantages |
| Temperature and Humidity Sensor (DHT22) | Measures ambient temperature and humidity levels. | Integrated into rooms where temperature and humidity need monitoring (e.g., offices, living spaces). | Provides accurate environmental data for climate control and comfort optimization; affordable and low power consumption. |
| Vibration Sensor (ADXL345) | Detects vibrations to monitor structural stability or abnormal activities. | Placed in structural areas or sensitive zones to detect irregular movements. | Helps identify early signs of structural damage or abnormal activity; low cost and lightweight, making it easy to install. |
| Gas Detector (MQ-2) | Detects gas leaks (e.g., smoke, propane, methane). | Positioned in kitchens, basements, or industrial areas where gas leaks are a concern. | Provides early warning for hazardous gas leaks, enhancing safety; low cost and sensitive to a range of gases. |
| Water Leak Sensor (Grove) | Identifies water leaks in areas prone to flooding or moisture damage. | Installed near plumbing systems, bathrooms, or basements. | Helps prevent water damage and flooding; highly responsive and easy to install, with a low power requirement. |
| Ambient Light Sensor (BH1750) | Measures ambient light intensity for lighting optimization. | Integrated into rooms to optimize natural and artificial lighting. | Allows for energy-efficient lighting adjustments based on room brightness; precise light sensing and low power consumption. |
| Occupancy Sensor (HC-SR501) | Detects human presence in areas like restrooms or offices. | Used in restrooms (e.g., WC) or rooms with variable occupancy levels. | Improves energy efficiency by controlling lighting or HVAC systems based on room usage; low cost and easy to configure. |
| Intrusion Sensor (SimpliSafe Entry Sensor) | Monitors windows or doors for unauthorized access, providing security alerts. | Installed on doors or windows to detect unauthorized entry. | Enhances building security with immediate alerts for intrusions; compact, reliable, and easily integrated with security systems. |

**4. Communication System**

ESP-NOW Protocol:

* Why ESP-NOW?: ESP-NOW allows direct communication between ESP32 devices without requiring Wi-Fi, making it ideal for low-latency and low-power environments.
* Implementation: Each room ESP32 sends its sensor data to the central ESP32 gateway using the ESP-NOW protocol.
* Challenges: Managing communication with multiple devices (11+ ESPs) requires careful handling of addressing and bandwidth.

Firebase Integration:

* **Firebase SDK**: The central ESP32 uses the Firebase SDK to send data to a real-time database.

Why Firebase SDK? Firebase SDK is chosen for its ease of integration with IoT devices, real-time data synchronization, and secure cloud storage. It allows for seamless communication between the ESP32 and the cloud, offering robust scalability and efficient data management. Its real-time capabilities ensure instant updates and monitoring, while its cost-effectiveness makes it an ideal choice for IoT projects requiring continuous data flow and remote access.

* Data Format: Sensor readings from each room are sent as JSON objects to corresponding nodes in the Firebase database.

Real-Time Visualization:

* **Google Analytics:** Google Analytics is integrated for data visualization and performance monitoring. It provides insightful analytics on sensor data, enabling real-time tracking of environmental conditions, energy consumption, and security alerts. Google Analytics is chosen for its powerful reporting tools, customizable dashboards, and ability to track trends over time. Its cost-effectiveness and wide integration capabilities make it an excellent choice for understanding and optimizing building performance in IoT projects.

**5. AI Model for Anomaly Detection**

Objective:

* The AI model is designed to process the incoming sensor data to identify potential issues such as:
  + Overconsumption of energy.
  + Detection of hazardous conditions (e.g., gas leaks, water leaks).
  + Unusual vibration patterns (indicating potential structural issues).

AI Integration Pipeline:

**Data Collection and Preprocessing**:

* + **Collection**: Sensor data from the ESP32 devices is transmitted to the Firebase real-time database.
  + **Preprocessing**: Data is cleaned, normalized, and transformed for analysis. This can either be done on the ESP32 devices or within Firebase using Cloud Functions, ensuring that the input data is ready for the AI model.

**Model Selection**:

* + **Option 1: Firebase ML Kit**: For lighter AI tasks such as basic anomaly detection, Firebase ML Kit is an ideal choice. It offers pre-built models and enables on-device machine learning, making it suitable for real-time analysis.
  + **Option 2: Custom Model**: For more advanced use cases, a custom machine learning model can be developed using TensorFlow or other frameworks. Once trained, the model can be deployed using Firebase's custom model hosting service.

**Model Training** (for custom models):

* + **Offline Training**: The AI model is trained using historical sensor data in a separate environment, such as Google Cloud AI or TensorFlow.
  + **Model Export**: After training, the model is exported in a format suitable for Firebase integration, such as TensorFlow Lite.

**Model Deployment**:

* + **Firebase ML Kit**:
  + For lightweight tasks like basic anomaly detection, Firebase ML Kit is a highly efficient solution. It offers pre-built machine learning models that are easy to integrate and optimized for real-time, on-device inference. If Firebase ML Kit is chosen, the model can be uploaded and managed directly within Firebase, enabling efficient distribution to all ESP32 devices
  + **Custom AI Model**:
  + For more complex tasks requiring advanced analysis, developing a custom AI model using TensorFlow or similar tools would be more suitable. The trained model can be hosted and deployed via Firebase, leveraging its powerful cloud infrastructure for inference and analysis. For custom models, Firebase's hosting service will manage model storage and scaling.
  + **Hybrid Approach**:
  + A hybrid approach could be implemented where simpler, on-device tasks are handled by Firebase ML Kit, while more complex analysis is offloaded to the cloud, using Firebase’s infrastructure for scalable, efficient AI processing and data management.

**Model Inference**:

* + **On-device Inference**: In cases where Firebase ML Kit is used, the model runs directly on the ESP32 devices, enabling low-latency, real-time decision-making.
  + **Cloud-based Inference**: For more complex models, sensor data is sent to Firebase where Cloud Functions are triggered to run inference in the cloud.
  + **Anomaly Detection and Alerts**:

The AI model continuously analyzes the incoming sensor data, identifying anomalies such as irregular energy consumption or environmental issues.

* + Firebase Cloud Functions are used to trigger real-time alerts and notifications whenever an anomaly is detected, ensuring immediate response and action.

**Continuous Improvement:**

* + The AI model can be retrained periodically using new sensor data stored in Firebase, allowing the system to improve its detection accuracy over time. This can be automated using Firebase’s scheduled Cloud Functions.

Technologies:

* Python for the AI model
* TensorFlow or Scikit-learn for machine learning.
* Firebase SDK to pull data and train the model.

**6. BCF Integration :**

* + The solution will include a mechanism to convert detected anomalies into BCF format, ensuring that alerts are easily communicated to the IFC model. This integration enhances the building's digital twin by providing real-time updates and ensuring that any identified issues are documented and managed effectively.

Technologies:

* BCF Format for alerting: The AI detects anomalies and automatically writes alerts into BCF files, which are integrated into the building’s IFC model.
* Rest api : to make a connection between the ai model and replica-a

**7. Results and Visualization**

Firebase Real-Time Data Monitoring:

* Data from each room (temperature, humidity, occupancy, etc.) is visualized in real-time using Firebase’s dashboard and Google Analytics.

AI Alerts and BCF Updates:

* Upon detecting anomalies, alerts are pushed to the BCF format, which integrates with the Building Information Modeling (BIM) system.
* This creates a dynamic and real-time view of the building’s status, effectively transforming it into a digital twin.

**8. Future Work**

* Scalability: Expand the system to support more rooms and buildings.
* Enhanced AI Models: Use more advanced deep learning models for better anomaly detection.
* Advanced Visualization: Implement 3D visualization of the building, where BCF alerts can be viewed directly on the digital twin’s georeferenced model.

**9.Future Suggestions**

To further enhance the capabilities of this project, an interesting extension would be integrating a **Raspberry Pi** for more complex processing and computational tasks. By leveraging the power of the Raspberry Pi, the project could incorporate **computer vision** to detect and analyze real-time visual data using cameras. For instance, implementing **object detection** or **pattern recognition** algorithms could provide more precise insights and improve decision-making capabilities. This would be particularly useful in scenarios where visual input is crucial, such as monitoring environments or recognizing specific objects. Furthermore, the Raspberry Pi’s compatibility with various sensors and its ability to run advanced machine learning models would allow for more autonomous and intelligent system behavior.

**10. Conclusion**

This project demonstrates how real-time IoT data can be integrated with digital twin technology and BCF for smarter facilities management. The system’s modular architecture and use of robust technologies such as ESP-NOW, Firebase, and AI make it scalable and adaptable for various infrastructure monitoring applications.