Einführung

Das Intelligente Hausüberwachungssystem dient zur Überwachung und Analyse von Sensordaten in Echtzeit. Es integriert verschiedene Sensoren wie Temperatur, Feuchtigkeit, Bewegung und Rauch, um ein umfassendes Monitoring der Umgebung zu ermöglichen. Das System bietet eine Benutzeroberfläche zur Visualisierung der Daten und nutzt maschinelles Lernen zur Erkennung von Anomalien.

Systemarchitektur

Das System besteht aus den folgenden Hauptkomponenten:

1. Sensoren: Simulierte oder echte Sensoren liefern Daten zu Temperatur, Feuchtigkeit, Bewegung und Rauch.
2. Message Queue: RabbitMQ wird verwendet, um die Sensordaten zwischen den Komponenten zu übertragen.
3. Datenbank: SQLite speichert die Sensordaten zur späteren Analyse.
4. Datenverarbeitung: Python-Skripte verarbeiten die eingehenden Daten und speichern sie in der Datenbank.
5. Benutzeroberfläche: Eine Webanwendung zeigt die Sensordaten in Echtzeit an und ermöglicht die Interaktion

Installation

### **Voraussetzungen**

* Python 3.x
* SQLite
* RabbitMQ
* Flask (für die Webanwendung)
* Plotly (für die Datenvisualisierung)
* Pika (für die RabbitMQ-Integration)

### **Schritt-für-Schritt-Anleitung**

1. **Repository klonen**:
2. bash
3. Code kopieren
4. git clone <repository-url> cd <repository-folder>
5. **Virtuelle Umgebung einrichten**:
6. bash
7. Code kopieren
8. python3 -m venv envsource env/bin/activate # On Windows use `env\Scripts\activate`
9. **Abhängigkeiten installieren**:
10. bash
11. Code kopieren
12. pip install -r requirements.txt
13. **RabbitMQ einrichten**: Installiere RabbitMQ und starte den Server. Details findest du in der offiziellen Dokumentation.
14. **Datenbank initialisieren**: Führe database\_setup.py aus, um die Datenbank und die notwendigen Tabellen zu erstellen:
15. bash
16. Code kopieren
17. python database\_setup.py
18. **Sensoren simulieren und Daten verarbeiten**: Starte sensor\_producer.py und sensor\_consumer.py:
19. bash
20. Code kopieren
21. python sensor\_producer.py
22. python sensor\_consumer.py
23. **Webanwendung starten**:
24. bash
25. Code kopieren
26. python app.py
27. **Zugriff auf die Webanwendung**: Öffne einen Webbrowser und gehe zu http://127.0.0.1:5000.

Nutzung

Die Webanwendung bietet ein Dashboard zur Visualisierung der Sensordaten. Die Diagramme werden in Echtzeit aktualisiert und zeigen Temperatur, Feuchtigkeit, Bewegung und Rauchstatus an.

* Temperatur und Feuchtigkeit: Diese Daten werden als Zeitreihen-Diagramme dargestellt.
* Bewegungs- und Rauchstatus: Diese werden als einfache Textanzeigen dargestellt.

Datenverarbeitung

Die Sensordaten werden in Echtzeit verarbeitet und in der SQLite-Datenbank gespeichert. Die Verarbeitung erfolgt in data\_processing.py und sensor\_consumer.py.

Die Funktion save\_data(data) in database\_setup.py speichert die Sensordaten in der Tabelle sensor\_data. Diese Tabelle enthält die Spalten:

* id: Eindeutige ID (Primärschlüssel)
* timestamp: Zeitpunkt der Datenerfassung
* temperature: Temperaturwert
* humidity: Feuchtigkeitswert
* motion: Bewegungsstatus (Boolean)
* smoke: Rauchstatus (Boolean)

Erweiterungsmöglichkeiten

* Weitere Sensoren: Integriere zusätzliche Sensoren, z.B. Licht, Luftqualität oder Schall.
* Benachrichtigungssystem: Implementiere Benachrichtigungen, die bei bestimmten Ereignissen ausgelöst werden (z.B. Rauchalarm).
* Maschinelles Lernen: Verwende ML-Modelle zur Vorhersage und Erkennung von Anomalien.
* Erweiterte Visualisierung: Integriere weitere Visualisierungstools oder Dashboards für eine detailliertere Datenanalyse.

Sicherheit

* Authentifizierung: Implementiere ein Authentifizierungssystem für die Webanwendung, um den Zugang zu sichern.
* Verschlüsselung: Nutze HTTPS für die sichere Datenübertragung.
* Datenbank-Sicherung: Implementiere regelmäßige Backups der Datenbank.

Fehlerbehandlung und Wartung