LLM's for Environmental Compliance

Data Monitoring and Anomaly Detection is a crucial part of environmental compliance. This process involves continuous monitoring of key environmental metrics like emissions, waste levels, water quality, or other pollutants, and using LLMs to analyse the data for compliance with regulations and detect anomalies that may signal a violation or a potential risk.

Steps for LLM-Assisted Environmental Data Monitoring and Anomaly Detection

Step 1: Data Collection from Sensors or Monitoring Systems

Environmental data can be collected from various sensors installed at facilities to monitor metrics like:

- **Air quality**: CO2, NOx, SOx, particulate matter.

Step 2: Data Processing

The raw data from environmental sensors needs to be cleaned and prepared for analysis. This involves handling missing data, normalizing values, and setting threshold limits based on regulatory standards.

- **Input**: Raw environmental data (e.g., air emissions, water quality, waste levels).
- **LLM Task**: Process data, calculate averages, and compare results to established regulatory limits.

Step 3: Real-Time Compliance Monitoring

An LLM can be used to analyze this processed data in real time and determine whether the monitored values are within permitted regulatory thresholds. If the values exceed predefined limits, the LLM can flag these instances for review.

Step 4: Anomaly Detection

Anomaly detection models (e.g., Isolation Forest, Autoencoders) can be combined with LLMs to detect abnormal data patterns that could indicate potential compliance issues or system malfunctions (e.g., a sensor malfunction, sudden spike in emissions). LLMs can also generate natural language explanations for the detected anomalies, helping users understand the significance of the issue.

- **Input**: Sensor data over time (e.g., daily, hourly emissions or water pollutant levels).
- **LLM Task**: Detect and describe anomalies that deviate from normal operational levels and suggest possible causes or corrective actions.

Anomaly Detection using Isolation Forest Model

```
import pandas as pd
import numpy as np
from fpdf import FPDF
import os
from dotenv import load_dotenv
from sklearn.ensemble import IsolationForest
import openai

# Set your OpenAI API key
load_dotenv()
openai_api_key = os.environ['OPENAI_API_KEY']

# Sample environmental data (e.g., emissions over time)
data = pd.DataFrame({
    'date': pd.date_range(start='2024-01-01', periods=30, freq='D'),
    'CO2_emissions': np.random.normal(loc=1500, scale=100, size=30),  # Emissions
in metric tons
    'NOx_emissions': np.random.normal(loc=300, scale=30, size=30),  # NOx
emissions in metric tons
})

# Step 1: Anomaly Detection using Isolation Forest
iso_forest = IsolationForest(contamination=0.05)
data['anomaly'] = iso_forest.fit_predict(data[['CO2_emissions',
'NOx_emissions']])

# Flagging rows that are anomalies
anomalies = data[data['anomaly'] == -1]
```

```
anomalies
```

	date	CO2_emissions	NOx_emissions	anomaly
0	2024-01-01	1280.196819	286.188987	-1
7	2024-01-08	1729.492271	331.891826	-1

LLM Analysis:

- For each anomaly, a natural language description is generated that includes the date and the emissions levels.
- The LLM (e.g., GPT-4) then interprets the anomaly, providing possible reasons (e.g., equipment failure, environmental conditions) and suggesting corrective actions (e.g., equipment maintenance, revisiting operational processes).

Automated Reporting and Notifications

Once anomalies are detected, LLMs can automatically generate reports or notifications to compliance officers, plant operators, or environmental engineers, summarizing the issues and recommending steps to correct them. This ensures timely action before regulatory violations occur.

```
for _, row in anomalies.iterrows():
    anomaly_description = f"""
    On {row['date'].strftime('%Y-%m-%d')}, an anomaly was detected with CO2 emissions
of {row['CO2_emissions']} metric tons and NOx emissions of {row['NOx_emissions']}
metric tons.
    These values are outside the expected range based on previous observations.
    """
    response = openai.Completion.create(
        engine="gpt-3.5-turbo-instruct",
        prompt=f"Explain the possible causes of this anomaly in environmental emissions
data and suggest potential corrective actions:\n\n{anomaly_description}",
        max_tokens=150
    )

    # Output the explanation provided by the LLM
    print(response.choices[0].text.strip())
```

Automated Reporting and Notifications

```
# Generate a compliance report for detected anomalies
report = "Environmental Compliance Report\n"
for _, row in anomalies.iterrows():
    report += f"Date: {row['date'].strftime('%Y-%m-%d')}\n"
    report += f"CO2 Emissions: {row['CO2_emissions']} metric tons\n"
    report += f"NOx Emissions: {row['NOx_emissions']} metric tons\n"
    report += f"Anomaly detected. Suggested action:
{response.choices[0].text.strip()}\n\n"
```

Create a Compliance Report in PDF Format

```
def create_pdf(summary, table_data=None):
    pdf = FPDF()
    pdf.add_page()

    # Add title
    pdf.set_font("Arial", size=12)
    pdf.cell(200, 10, txt="Summary", ln=True, align='C')

    # Add summary text
```

```
    pdf.set_font("Arial", size=10)
    pdf.multi_cell(0, 10, summary)

    # Add table if data is provided
    if table_data:
        pdf.ln(10)
        pdf.set_font("Arial", size=12)
        pdf.cell(200, 10, txt="Table", ln=True, align='C')

        pdf.set_font("Arial", size=10)
        col_width = pdf.w / 4.5
        row_height = pdf.font_size

        for row in table_data:
            for item in row:
                pdf.cell(col_width, row_height*1.5, txt=item, border=1)
            pdf.ln(row_height*1.5)

    pdf.output("../output/summary.pdf")



summary = create_pdf(report)
```

By analyzing historical data and feedback from anomalies, LLMs can improve over time, providing more accurate anomaly predictions and suggesting better solutions. LLMs can also integrate feedback from field engineers and adjust their models to adapt to evolving environmental regulations.

Benefits of LLM-Assisted Environmental Monitoring

- **Proactive Issue Detection**: Anomalies are flagged in real-time, allowing for quicker response and corrective action.
- **Efficiency**: Automates the generation of compliance reports, reducing human effort and error.
- **Cost Reduction**: Identifying issues early can reduce fines and penalties for non-compliance, and minimize the cost of repairs or operational downtime.
- **Transparency**: LLMs provide clear, explainable insights that help decision-makers understand environmental risks and ensure that corrective actions align with regulations.

By integrating LLMs into environmental data monitoring systems, organizations can better manage compliance risks, improve operational efficiency, and contribute to more sustainable environmental practices.