Karachi Air Quality Prediction System - Technical Report

Syed Muhammad Aun Haider (Habib University x 10Pearls Pakistan)

Dataset

- **Source**: Open-Meteo API for historical and real-time AQI and PM2.5 data
- **Timeframe**: 3-year dataset covering 2023-2025
- **Features**: Hourly timestamp, calculated AQI, PM2.5 concentration, temporal features (hour, day, month, year), AQI change metrics
- **Storage**: Hugging Face feature store
- **Update Mechanism**: New rows appended hourly with current AQI measurements
- **Target Variables**: Three columns for 24-hour, 48-hour, and 72-hour future AQI values filled using forward-looking logic (when available)

Models

- **Algorithm Evaluation**: Tested XGBoost, Random Forest, Ridge Regression, and Neural Network
- **Selection**: Random Forest selected for deployment based on performance metrics
- **Performance**: Random Forest achieved lowest Mean Absolute Error (MAE) and highest R^2 score
- **Architecture**: Three separate models for 1-day, 2-day, and 3-day predictions
- **Training Features**: Temporal patterns, current AQI, PM2.5 levels, and 24-hour AQI changes
- **Storage**: Hugging Face model registry
- **Retraining**: Daily updates at 02:00 UTC to incorporate recent data patterns

Hourly Script

- **File**: hourly_predict.py

- **Execution**: Runs every hour via GitHub Actions
- **Functions**:
  - Fetches current AQI from Open-Meteo API
  - Loads pre-trained Random Forest models from Hugging Face
  - Generates 3-day AQI predictions
  - Updates dataset with new hourly entry
  - Implements forward-filling for target columns by matching with future AQI values if available
- **Target Update Logic**: Each row's target columns populated with AQI values from rows 24, 48, and 72 indices ahead
- **Output**: Appends new data to dataset and uploads predictions to Hugging Face

Daily Script

- **File**: daily_train.py
- **Execution**: Runs daily at 02:00 UTC via GitHub Actions
- **Functions**:
  - Loads complete dataset from Hugging Face
  - Retrains Random Forest models with updated data
  - Evaluates model performance using MAE and R^2 metrics
  - Uploads improved models to Hugging Face registry
  - Maintains model version history
- **Objective**: Ensures models adapt to changing air quality patterns and seasonal variations

Streamlit Dashboard

- **File**: app.py
- **Platform**: Streamlit
- **Features**:
  - Real-time AQI display with color-coded health categories
  - 3-day forecast visualization using bar charts

  - Health advisory system with specific recommendations
  - System status monitoring with update schedules
  - Alert when aqi is hazardous

- **Synchronization**: Implements 90-second delay after each hour to ensure predictions match current AQI
- **User Interface**: Responsive design with sidebar controls and main display panels
- **Data Sources**: Live AQI from Open-Meteo, predictions from Hugging Face

Automation

- **Platform**: GitHub Actions with cronjob triggers
- **Cronjob Schedules**:
  - Hourly predictions: 0 * * * * (Runs at minute 0 of every hour)
  - Daily training: 0 2 * * * (Runs at 02:00 UTC daily)
- **Trigger Mechanism**: GitHub Actions workflows activated by cronjob schedules defined in YAML configuration
- **Workflow Files**:
  - .github/workflows/hourly_predict.yml: Contains hourly cron schedule
  - .github/workflows/daily_train.yml: Contains daily training schedule
- **Error Handling**: Automatic retry on failure with exponential backoff
- **Execution Environment**: GitHub-hosted runners with pre-configured Python environments
- **Dependency Management**: Automated package installation from requirements.txt
- **Logging**: Complete execution logs available in GitHub Actions interface
- **Notification**: Email alerts for workflow failures (configured in repository settings)
- **Reliability**: Cronjob persistence ensures scheduled execution even during GitHub downtime
- **Monitoring**: Real-time status tracking through GitHub Actions dashboard

- **Time Zone**: All cronjobs configured in UTC time zone for consistency