**Real-Time Japan Pollution Prediction**

This repository contains code for evaluating and selecting the best deep learning models for predicting future pollution levels based on historical data from Japan. The goal is to determine which deep learning model performs best in terms of Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) for predicting pollution levels.

Dataset : !wget https://www.dropbox.com/s/wa8d1sujzlx56hh/ETL\_DATA\_new.csv

The dataset used in this project contains historical pollution data from various locations in Japan. Each column in the dataset represents a different pollution metric or feature. The dataset is used to train and evaluate several deep learning models to predict future pollution levels based on past observations.

**Models**

The following models are evaluated in this project:

1. LSTM (Long Short-Term Memory) : A type of Recurrent Neural Network (RNN) designed to capture long-term dependencies in sequential data.

2. GRU (Gated Recurrent Unit) : A variant of LSTM that uses a different gating mechanism for better performance and reduced computational complexity.

3. SimpleRNN : A basic form of RNN that captures temporal dependencies in sequences.

4. Dense Neural Network (DNN) : A feedforward neural network that does not use sequential information but instead relies on dense layers to learn from the data.

5. Bidirectional LSTM : An extension of LSTM where the network is trained in both forward and backward directions to capture dependencies from both past and future data.

**Code Structure**

- `models.py`: Contains functions to create and compile various deep learning models.

- `evaluate\_models.py`: Contains the logic to prepare the data, train each model, evaluate performance, and save metrics.

**Installation**

To run the code, you'll need to install the required packages. You can do this using pip:

```bash

pip install numpy pandas scikit-learn tensorflow

**Usage**

1. **Prepare the Data**: Ensure your dataset is in a suitable format and is loaded into a DataFrame df.
2. **Run Model Evaluation**:
   * Execute evaluate\_models.py to train and evaluate the models.
   * The script will output performance metrics for each model in CSV format.

bash

Copy code

python evaluate\_models.py

1. **Metrics**:
   * The evaluation script will generate CSV files containing performance metrics for each model, including:
     + mse\_metrics.csv: Contains Mean Squared Error metrics.
     + rmse\_metrics.csv: Contains Root Mean Squared Error metrics.
     + mae\_metrics.csv: Contains Mean Absolute Error metrics.
2. **Review Results**:
   * Review the CSV files to compare the performance of different models and select the best-performing one based on your criteria.

**Results**

The results include performance metrics for each model, which are stored in the CSV files. These metrics will help you determine which model provides the best predictions for future pollution levels.