A Machine Learning and Time Series Approach to the Ground-Level CO_2 Forecast

Version: 1.0.0
Date: 2025-04-19
Code version: 1.2.0

Table of Contents

1. Introduction

1.1. Purpose of this Document

This document provides instructions and information on setting up the necessary environment, running the CO_2 forecasting script (`rf-sarima-co2-forecast.py`), understanding its outputs, and effectively utilizing the generated forecast results for relevant purposes.

1.2. Forecast Goal Recap

The primary goal of the forecasting system is to predict and forecast future monthly average ground-level CO_2 concentrations (in parts per million, ppm) based on historical observations and related atmospheric and meteorological data, using a hybrid machine learning (Random Forest) and time series (SARIMA) approach.

1.3. Overview of the Forecasting System

The system utilizes a Python script (`rf-sarima-co2-forecast.py`) that implements a hybrid machine learning (Random Forest) and time series (SARIMA) model. It takes historical data as input and produces several CSV forecast and prediction files, as well as PNG visualization plots, including feature importance analyses (Permutation and SHAP) and seasonal decomposition.

2. Forecasting Model Overview

2.1. Methodology

The script employs a hybrid modeling approach:

1. **Random Forest Regressor:** A machine learning model (`sklearn.ensemble.RandomForestRegressor` with `n_estimators=100`, `random_state=42`) is trained to predict ground-level CO_2 (`co2obs`) using satellite-derived CO_2 column concentrations, CAMS column concentrations (CO, CH_4), and weather variables (`tcco2`, `tcco_1e4`, `tcch4_1e4`, `u10`, `v10`, `t2m`, `mslp`). Model interpretability is enhanced using Permutation Importance and SHAP (TreeExplainer).
2. **SARIMA (Seasonal AutoRegressive Integrated Moving Average):** A statistical time series model (`statsmodels.tsa.statespace.SARIMAX`) is used to model and forecast the time series of the Random Forest's predicted CO_2 concentrations (`predicted_rf_co2`). The script automatically searches for the optimal SARIMA parameters (p, d, q ranging from 0 to 1) and seasonal parameters (P, D, Q ranging from 0 to 1, with seasonality m = 12) using a grid search, evaluated by the Akaike Information Criterion (AIC). The best SARIMA model is then used to generate future forecasts for the next 12 months.
3. **Seasonal Decomposition:** The script performs an additive seasonal decomposition (`statsmodels.tsa.seasonal.seasonal_decompose` with `period=12`) on the Random Forest predicted CO_2 time series to analyze trend, seasonal, and residual components.

2.2. Key Input Data

The script requires an input CSV file named `co2_downscaling_data.csv`. This file must contain historical monthly data with at least the following columns:

- `year`: The year of the observation.
- `month`: The month of the observation (1-12).
- `co2obs`: The observed ground-level CO_2 concentration (target variable for Random Forest).
- `tcco2`: Total column CO_2 (feature).
- `tcco_1e4`: Total column CO (scaled, feature).
- `tcch4_1e4`: Total column CH_4 (scaled, feature).
- `u10`: 10m U-component of wind (feature).
- `v10`: 10m V-component of wind (feature).
- `t2m`: 2m temperature (feature).
- `mslp`: Mean sea level pressure (feature).

2.3. Model Training Process

1. Data is loaded from `co2_downscaling_data.csv`.
2. Features (X: `tcco2`, `tcco_1e4`, `tcch4_1e4`, `u10`, `v10`, `t2m`, `mslp`) and target (y: `co2obs`) are defined.
3. A Random Forest model (`RandomForestRegressor`) is trained on the entire dataset (X, y).
4. Random Forest model performance is evaluated using R, R^2, RMSE, MAE, MBE, and 5-fold cross-validation (scoring='r2').
5. Feature importance is assessed using Permutation Importance (`n_repeats=30`) and SHAP (`shap.TreeExplainer`).
6. The Random Forest predictions (`predicted_rf_co2`) are generated for the input data.
7. The trained Random Forest model is saved to `random_forest_co2_model.pkl` using the `joblib library`.
8. The original data frame with the added `predicted_rf_co2` column is saved to `co2_predictions_random_forest.csv`.
9. A datetime index is created for the `predicted_rf_co2` time series.
10. A grid search (`optimize_sarima` function) is performed using the `predicted_rf_co2` time series to find the optimal (p,d,q)(P,D,Q)m parameters for the SARIMA model based on AIC.
11. The best SARIMA model configuration found is then fit on the entire `predicted_rf_co2` time series.
12. This final SARIMA model is used to generate forecasts (`get_forecast`) for the next 12 months, including 95% confidence intervals.
13. The forecast results are saved to `sarima_forecast_next_12_months.csv`.
14. Seasonal decomposition is performed on the `predicted_rf_co2` time series, and results are saved to `seasonal_decomposition_rf_co2.csv`.
15. Various plots visualizing the data, predictions, feature importance, forecast, and decomposition are generated and saved as PNG files.

2.4. Final Output

The script generates several output files:

- **Model:**
  - `random_forest_co2_model.pkl`: Saved trained Random Forest model.
- **Predictions & Data:**
  - `co2_predictions_random_forest.csv`: Original data with added `predicted_rf_co2` column and datetime index.
  - `sarima_forecast_next_12_months.csv`: 12-month SARIMA forecast (based on

  - `predicted_rf_co2`) including mean, lower CI, and upper CI.
  - `seasonal_decomposition_rf_co2.csv`: Results of seasonal decomposition on `predicted_rf_co2` (trend, seasonal, residual, trend+seasonal).
- **Plots (PNG):**
  - `rf_scatter_plot.png`: Observed vs. Predicted CO_2 (Random Forest) scatter plot.
  - `feature_importance_percent.png`: Permutation feature importance bar chart (%).
  - `shap_bar_plot.png`: SHAP feature importance bar chart (%).
  - `shap_summary_plot.png`: SHAP summary (beeswarm) plot.
  - `shap_dependence_dashboard.png`: SHAP dependence plots for all features, colored by `tcco2`.
  - `co2_timeseries_comparison.png`: Time series plot comparing `co2obs`, `tcco2`, and `predicted_rf_co2`.
  - `sarima_forecast_plot.png`: Full historical `predicted_rf_co2` time series plus 12-month SARIMA forecast with confidence intervals.
  - `co2_last6_forecast12.png`: Plot showing the last 6 months of `predicted_rf_co2` and the 12-month SARIMA forecast.
  - `seasonal_decomposition_rf_co2.png`: Plot showing the components of seasonal decomposition (`predicted_rf_co2`).
  - `trend_plus_seasonal_timeseries.png`: Plot comparing `predicted_rf_co2` with its trend + seasonal component.

The SARIMA forecast provides monthly point forecasts for the *Random Forest-predicted* ground-level CO_2, along with 95% confidence intervals.

3. Prerequisites & Setup

3.1. Required Software

- Python (version 3.x recommended)

3.2. Required Libraries

- pandas
- numpy
- shap
- os
- seaborn
- scikit-learn (`sklearn`)
- joblib
- matplotlib
- statsmodels
- warnings (used to suppress warnings)

3.3. Setting up the Environment (Recommended)

Using a Python virtual environment is highly recommended to manage dependencies.

1. Navigate to your project directory in the terminal.
2. Create the virtual environment: `python3 -m venv .venv`
3. Activate the environment: `source .venv/bin/activate` (Linux/macOS) or `.\.venv\Scripts\activate` (Windows)
4. Install required libraries (while the environment is active): `pip install pandas numpy shap seaborn scikit-learn joblib matplotlib statsmodels`

3.4. Input Data File (`co2_downscaling_data.csv`)

- **Location:** Place the input CSV file (`co2_downscaling_data.csv`) in the same directory as the Python script (`rf-sarima-co2-forecast.py`).
- **Format:** Ensure the CSV file contains the necessary columns with appropriate monthly data as described in section 2.2.

4. Generating the Forecast

4.1. The Python Script

The forecasting script is named `rf-sarima-co2-forecast.py`.

4.2. Running the Script

1. Open your terminal or command prompt.
2. Navigate to the directory containing the script and the `co2_downscaling_data.csv` file.
3. Activate the virtual environment (if used): `source .venv/bin/activate` or `.\.venv\Scripts\activate`
4. Execute the script: `python rf-sarima-co2-forecast.py`
5. The script will print evaluation metrics, progress updates, and final forecast data to the console and save output files to the current directory.

4.3. Expected Runtime

The script execution time can vary depending on the size of the input dataset and the computational cost of the Random Forest training, SHAP analysis, and SARIMA parameter grid search. The grid search and SHAP value computation are typically the most time-consuming parts.

5. Understanding the Output Files

Upon successful execution, the script generates multiple output files in the same directory:

5.1. Forecast Data File (`sarima_forecast_next_12_months.csv`)

This CSV file contains the numerical SARIMA forecast results based on the *Random Forest predicted CO_2*:

- **date:** The month and year of the forecast (e.g., 2025-01-01).
- **forecasted_co2:** The SARIMA model's best estimate (point forecast) for the *predicted* ground-level CO_2 concentration (in ppm) for that specific month.
- **lower_ci:** The lower bound of the 95% confidence interval for the forecast (in ppm).
- **upper_ci:** The upper bound of the 95% confidence interval for the forecast (in ppm).
- **Interpreting Confidence Intervals:** The range between `lower_ci` and `upper_ci` represents the statistically likely range where the actual value of the *Random Forest predicted CO_2* could fall, according to the SARIMA model's assumptions. A wider interval indicates greater uncertainty.

(Refer to Section 2.4 for other data files like co2_predictions_random_forest.csv and seasonal_decomposition_rf_co2.csv).

5.2. Forecast Plot Files

Multiple PNG image files provide visual representations:

- `sarima_forecast_plot.png`: Shows the full historical time series of `predicted_rf_co2` (grey line), followed by the 12-month SARIMA forecast (blue line) and its 95% confidence interval (blue shaded area).
- `co2_last6_forecast12.png`: Focuses on the last 6 months of historical `predicted_rf_co2` data and the subsequent 12-month SARIMA forecast with confidence intervals.

(Refer to Section 2.4 for other plot files visualizing Random Forest performance, feature importance, time series comparisons, and seasonal decomposition).

6. Utilizing the Forecast Results

The generated outputs (predictions, forecasts, confidence intervals, plots, feature importance) can be used in various ways:

6.1. Reporting

Include the numerical forecasts (`sarima_forecast_next_12_months.csv`), confidence intervals, and relevant plots (e.g., `sarima_forecast_plot.png`, `feature_importance_percent.png`, `shap_summary_plot.png`, `co2_timeseries_comparison.png`) in reports, summaries, or presentations regarding future CO_2 level expectations and model insights.

6.2. Planning & Decision Support

Use the expected trend (from decomposition) and range (forecast + CIs) of CO_2 levels to inform planning activities, resource allocation, or policy considerations. Use feature importance plots (`feature_importance_percent.png`, `shap_summary_plot.png`, `shap_dependence_dashboard.png`) to understand the key drivers influencing the Random Forest predictions.

6.3. Monitoring & Evaluation

As new actual CO_2 observations (`co2obs`) become available, compare them against the previously generated Random Forest predictions (`predicted_rf_co2` in `co2_predictions_random_forest.csv`). Evaluate how well the SARIMA forecast (`sarima_forecast_next_12_months.csv`) anticipated the evolution of the predicted series. This helps evaluate the performance of both model components over time.

6.4. Input for Further Analysis

The Random Forest predictions (`co2_predictions_random_forest.csv`) or the SARIMA forecast data (`sarima_forecast_next_12_months.csv`) can serve as input for subsequent analyses, simulations, or impact models that require future CO_2 projections.

7. Limitations & Considerations

It is crucial to use the results responsibly, keeping the following limitations in mind:

7.1. Model Assumptions

- **Random Forest:** Captures non-linear relationships between features and `co2obs`, but its predictive power is limited by the information contained in the provided features. It does not inherently extrapolate trends beyond the range seen in training data.
- **SARIMA:** Models the temporal patterns (trend, seasonality) present in the *Random Forest predicted CO_2* series. It relies on statistical assumptions (such as stationarity after differencing) and assumes that these historical patterns will persist into the future. The forecast quality depends on how well the Random Forest predictions capture the true underlying CO_2 behavior.

7.2. Dependence on Input Data Quality

The accuracy of both the Random Forest predictions and the subsequent SARIMA forecast is highly dependent on the quality, completeness, accuracy, and representativeness of the input data (`co2_downscaling_data.csv`). Errors or biases in the input will propagate through the models.

7.3. Forecast Horizon Uncertainty

Forecast uncertainty naturally increases for periods further into the future. This is reflected in the widening of the SARIMA confidence intervals (`lower_ci`, `upper_ci`) on the plots and in the CSV data. Forecasts for distant months are inherently less reliable.

7.4. Static Model

The current script trains the Random Forest model and generates a SARIMA forecast based on the data available at the time of execution. It does not automatically update or learn from new data points unless the script is re-run with updated input data. The saved Random Forest model (`.pkl`) remains static unless it is explicitly retrained.

7.5. Exogenous Factors

The model primarily relies on historical patterns within the provided features (`X`) and the target (`y`). It cannot predict the impact of sudden, unforeseen events (e.g., major volcanic eruptions, significant policy shifts, pandemics affecting emissions) that are not reflected in the historical data patterns or input features.

8. Maintenance & Updates

To keep the forecast relevant and accurate:

8.1. Incorporating New Data

Regularly update the input file (`co2_downscaling_data.csv`) with the latest monthly observations for all required columns.

8.2. Re-running the Forecast

Execute the `rf-sarima-co2-forecast.py` script periodically after updating the input data. This will retrain the Random Forest model on the latest data, generate updated predictions, re-optimize and refit the SARIMA model, and produce fresh forecasts and analyses. The optimal frequency depends on the application and the availability of data (e.g., monthly).

8.3. Potential Need for Model Retraining/Parameter Re-tuning

Continuously monitor the performance of the Random Forest predictions against new `co2obs` data and the SARIMA forecast performance (Section 6.3). If accuracy degrades significantly over time, it might indicate a shift in underlying processes. Re-running the script (Section 8.2) handles retraining. Significant degradation might warrant revisiting feature engineering or model choices, although the current script automatically re-optimizes SARIMA parameters.

8.4. Library Updates

Occasionally, update the Python libraries within the virtual environment (`pip install --upgrade pandas numpy shap seaborn scikit-learn joblib matplotlib statsmodels`) to benefit from bug fixes and improvements. However, always test the script thoroughly after updates, as changes in libraries can sometimes introduce compatibility issues or slightly alter results.

9. Conclusion

This hybrid Random Forest and SARIMA forecasting system provides a valuable tool for predicting and estimating future ground-level CO_2 concentrations based on historical data and related environmental variables. By understanding the model's methodology, including feature importance analysis and seasonal decomposition, as well as its inputs, outputs, and limitations outlined in this document, users can effectively generate, interpret, and utilize the forecast results for informed reporting, planning, and decision-making. Remember to use the forecasts responsibly, acknowledging the inherent uncertainties and the specific methodology employed in forecasting the ML model's output.

Complete Code

Access to the code via the following link:
https://link.bmkg.go.id/rf-sarima-co2-forecast