Statistical Methodology Framework

Weather Stability vs Renewable Energy Model Performance

Mathematically Rigorous Solutions to Five Critical Methodological Challenges

Comprehensive framework for weather stability classification and renewable energy prediction model evaluation

Academic Methodology

Mathematical Derivation • Statistical Validation • Operational Relevance

Contents

1 Abstract 3

2 Introduction 3

3 Problem 1: Stability Classification Formula and Statistical Methods 3
- 3.1 Literature Review and Theoretical Foundation 4
- 3.2 Mathematical Framework 4
- 3.3 Gaussian Mixture Model Classification 4
- 3.4 Classification Rules 4

4 Problem 2: Optimal Temporal Resolution (Sliding Window) 5
- 4.1 Critical Analysis of Temporal Scales 5
  - 4.1.1 Hourly Classification Issues 5
  - 4.1.2 Daily Classification Issues 5
- 4.2 Meteorological Justification for 6-Hour Windows 6
- 4.3 Implementation Framework 6
- 4.4 Sensitivity Analysis Framework 6

5 Problem 3: Temporal Dependence in Stability Classification 7
- 5.1 Mathematical Treatment of Temporal Dependence 7
- 5.2 Method 1: Hidden Markov Model (HMM) - Recommended 7
- 5.3 Parameter Estimation 7
- 5.4 State Sequence Estimation 7
- 5.5 Alternative Methods 8
  - 5.5.1 Method 2: Autoregressive Classification 8
  - 5.5.2 Method 3: Exponentially Weighted Moving Average (EWMA) 8

6 Problem 4: Mathematical Validation of Error-Stability Relationship 8
- 6.1 Statistical Tests for Proving Error Difference 8
  - 6.1.1 Test 1: Mann-Whitney U Test (Primary, Non-parametric) 9
  - 6.1.2 Test 2: Welch's t-test (Parametric Alternative) 9
  - 6.1.3 Test 3: Linear Mixed-Effects Model 9
  - 6.1.4 Test 4: Permutation Test (Distribution-free Validation) 9
- 6.2 Effect Size Measures 10
- 6.3 Multiple Testing Correction 10

7 Problem 5: Mathematical Definition of Model Robustness 10
- 7.1 Candidate Robustness Metrics 10
  - 7.1.1 Metric 1: Relative Performance Degradation (RPD) - Recommended Primary . 11
  - 7.1.2 Metric 2: Absolute Performance Degradation (APD) 11
  - 7.1.3 Metric 3: Robustness Index (Composite Score) 11
  - 7.1.4 Metric 4: Stability of Performance (Coefficient of Variation) 11
  - 7.1.5 Metric 5: Skill Score Preservation 11
- 7.2 Recommended Combined Approach 12

7.3 Statistical Validation of Robustness Rankings 12

8 Mathematical Consistency Across All Five Problems 12

8.1 Unified Framework Integration 12

8.2 Validation Pipeline 12

8.3 Mathematical Assumptions and Limitations 13

9 Implementation Details 13

9.1 Software Requirements 13

9.2 Pseudocode Implementation 14

9.3 Reproducibility Guidelines 16

10 Expected Outcomes and Validation 16

10.1 Expected Outcomes 16

10.2 Validation Criteria 17

11 References 17

11.1 Weather Regime Detection 17

11.2 Statistical Methods 17

11.3 Renewable Energy Forecasting 18

11.4 Robustness Metrics 18

12 Conclusion 18

1 Abstract

This document provides academically rigorous, mathematically validated solutions to five critical methodological challenges in weather stability classification and renewable energy prediction model evaluation. We address: (1) stability classification formulas and statistical methods, (2) optimal temporal resolution for sliding windows, (3) temporal dependence in stability classification, (4) mathematical validation of error-stability relationships, and (5) mathematical definition of model robustness. The framework integrates all solutions into a unified validation pipeline with statistical justification at each step.

2 Introduction

Weather stability significantly impacts the performance of renewable energy prediction models. However, existing literature lacks a comprehensive, mathematically rigorous framework for quantifying this relationship. This document addresses five critical methodological challenges that must be resolved to establish a scientifically sound approach to weather stability analysis in renewable energy forecasting.

Problem Statement

Research Challenge: How can we mathematically and statistically validate that weather stability affects renewable energy prediction model performance, and provide operational guidance on model selection under different weather conditions?

The five problems addressed are:

1. **Stability Classification Formula:** What mathematical formula and statistical methods should be used to determine if a period is stable or unstable?
2. **Temporal Resolution:** What sliding window should be used for stability classification (hourly, daily, or other)?
3. **Temporal Dependence:** How do we account for the dependence of each period on previous periods in stability classification?
4. **Error-Stability Validation:** What mathematical metrics can prove that error points corresponding to unstable periods have significantly more error than stable periods?
5. **Model Robustness:** How do we mathematically determine the most robust model?

3 Problem 1: Stability Classification Formula and Statistical Methods

Problem Statement

Problem: How would we say that this period is stable or not based on the stability index? What formula exactly, and what statistical methods will be used over the data to determine its stability?

3.1 Literature Review and Theoretical Foundation

Weather regime detection has been extensively studied in meteorology and climatology. The fundamental challenge is distinguishing between persistent atmospheric states (stable regimes) and transitional periods (unstable regimes). Traditional approaches include:

- **Threshold-based methods:** Simple but arbitrary cutoffs
- **K-means clustering:** Sensitive to initialization and assumes spherical clusters
- **Principal Component Analysis:** Linear dimensionality reduction, may miss non-linear patterns
- **Gaussian Mixture Models:** Probabilistic approach with uncertainty quantification

Solution

Recommended Approach: Multi-level hierarchical classification combining instantaneous stability metrics, rolling window variability metrics, and regime detection with state persistence.

3.2 Mathematical Framework

Mathematical Framework

Weather Stability Index (WSI) Computation

Let $\mathbf{X}_t = [x_{1,t}, x_{2,t}, ..., x_{p,t}]$ be the vector of p weather features at time t. We compute the WSI using robust normalization:

$$\text{WSI}_t = \frac{1}{p}\sum_{i=1}^{p} \frac{x_{i,t} - \text{median}(x_i)}{\text{IQR}(x_i)} \tag{1}$$

where IQR is the interquartile range, providing robustness to outliers.

3.3 Gaussian Mixture Model Classification

We model the distribution of WSI values using a Gaussian Mixture Model:

$$P(\text{WSI}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\text{WSI}|\mu_k, \sigma_k^2) \tag{2}$$

where:

- K is the number of regimes (determined by BIC)
- π_k are mixing proportions
- μ_k, σ_k^2 are regime-specific means and variances

3.4 Classification Rules

Soft Classification:

$$P(\text{regime}_k|\text{WSI}_t) = \frac{\pi_k \mathcal{N}(\text{WSI}_t|\mu_k, \sigma_k^2)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\text{WSI}_t|\mu_j, \sigma_j^2)} \tag{3}$$

Hard Classification:

$$\text{regime}_t = \arg\max_k P(\text{regime}_k | \text{WSI}_t) \tag{4}$$

Binary Classification (Stable/Unstable):

$$\text{unstable}_t = \begin{cases} 1 & \text{if } P(\text{unstable}|\text{WSI}_t) > \tau \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\tau = 0.5$ is the classification threshold.

Validation Criteria

Statistical Validation Criteria:

- **Silhouette Score:** $s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$ where a_i is the average distance to points in the same cluster, b_i is the average distance to points in the nearest other cluster. Acceptable threshold: $s > 0.4$.
- **Davies-Bouldin Index:** $\text{DB} = \frac{1}{K}\sum_{i=1}^{K} \max_{j \neq i} \frac{\sigma_i + \sigma_j}{d(c_i, c_j)}$. Lower values indicate better clustering.
- **Calinski-Harabasz Score:** $\text{CH} = \frac{\text{SSB}/(K-1)}{\text{SSW}/(N-K)}$ where SSB is between-cluster sum of squares, SSW is within-cluster sum of squares. Higher values indicate better clustering.
- **Bayesian Information Criterion (BIC):** $\text{BIC} = -2\ln(L) + k\ln(n)$ where L is the likelihood, k is the number of parameters, n is the sample size.

4 Problem 2: Optimal Temporal Resolution (Sliding Window)

Problem Statement

Problem: What are the sliding windows that will be used to say the point is stable? Is it hourly so this hour is stable, or daily so this day is stable, or not? This is a very critical problem.

4.1 Critical Analysis of Temporal Scales

4.1.1 Hourly Classification Issues

- **Noise:** Single-hour measurements are noisy and may not represent true atmospheric state
- **Lack of persistence:** Ignores meteorological memory and regime persistence
- **Operational irrelevance:** Grid operators typically plan on longer timescales

4.1.2 Daily Classification Issues

- **Loss of transitions:** Misses intra-day weather changes (e.g., storm front passage)
- **Coarse resolution:** May average out important sub-daily patterns
- **Limited sensitivity:** May miss rapid changes affecting renewable generation

Solution

Recommended Solution: 6-hour blocks with temporal smoothing, justified by meteorological synoptic scale changes and operational alignment.

4.2 Meteorological Justification for 6-Hour Windows

Weather systems operate on characteristic timescales:

- **Mesoscale:** 2-6 hours (convective systems, local winds)
- **Synoptic scale:** 6-12 hours (frontal systems, pressure changes)
- **Planetary scale:** Days to weeks (large-scale patterns)

The 6-hour window captures synoptic-scale changes while maintaining sensitivity to mesoscale phenomena.

4.3 Implementation Framework

Mathematical Framework

Multi-Scale Temporal Hierarchy

Level 1: Instantaneous (hourly) WSI computation Level 2: Rolling 6-hour window statistics

For each 6-hour window $W_t = [t-2, t-1, t, t+1, t+2, t+3]$:

$$\mathrm{WSI}_{\mathrm{window},t} = \frac{1}{6}\sum_{i \in W_t} \mathrm{WSI}_i \tag{6}$$

$$\mathrm{WSI}_{\mathrm{std},t} = \sqrt{\frac{1}{5}\sum_{i \in W_t} (\mathrm{WSI}_i - \mathrm{WSI}_{\mathrm{window},t})^2} \tag{7}$$

$$\mathrm{WSI}_{\mathrm{trend},t} = \frac{\mathrm{WSI}_{t+3} - \mathrm{WSI}_{t-2}}{5} \tag{8}$$

Level 3: Temporal smoothing

$$\mathrm{WSI}_{\mathrm{smoothed},t} = \mathrm{median}(\mathrm{WSI}_{\mathrm{window},t-1}, \mathrm{WSI}_{\mathrm{window},t}, \mathrm{WSI}_{\mathrm{window},t+1})$$

Level 4: Hourly label assignment for model evaluation

4.4 Sensitivity Analysis Framework

Test window sizes: $w \in \{3, 6, 12, 24\}$ hours

For each window size:

1. Compute stability classifications
2. Calculate agreement with reference (6-hour) using Cohen's kappa
3. Measure discriminative power in model performance

Discriminative Power Metric:

$$\mathrm{DP}_w = \frac{|\mathrm{MAE}_{\mathrm{unstable}} - \mathrm{MAE}_{\mathrm{stable}}|}{\mathrm{MAE}_{\mathrm{pooled}}} \tag{9}$$

where $\mathrm{MAE}_{\mathrm{pooled}}$ is the overall mean absolute error.

5 Problem 3: Temporal Dependence in Stability Classification

Problem Statement

Problem: Regarding the formula that determines the period we choose to be stable or not, isn't each nth period dependent on all the previous ones n-1, n-2, and so on? How will we account for that dependence?

5.1 Mathematical Treatment of Temporal Dependence

Weather stability exhibits strong temporal persistence due to atmospheric inertia. The stability at time t depends on previous states: $\text{stability}_t = f(\text{stability}_{t-1}, \text{stability}_{t-2}, ..., \text{weather}_t)$.

Solution

Primary Solution: Hidden Markov Model (HMM) with Viterbi algorithm for state sequence estimation, explicitly modeling temporal dependence and regime persistence.

5.2 Method 1: Hidden Markov Model (HMM) - Recommended

Mathematical Framework

HMM Framework

States: $S = \{\text{Stable}, \text{Unstable}\}$ (or $\{\text{Stable}, \text{Transitional}, \text{Unstable}\}$)

Observations: Computed WSI features $\mathbf{O}_t$

Transition Probabilities: $A_{ij} = P(S_t = j | S_{t-1} = i)$

Emission Probabilities: $B_j(\mathbf{O}_t) = P(\mathbf{O}_t | S_t = j)$

Initial State Probabilities: $\pi_i = P(S_1 = i)$

5.3 Parameter Estimation

Using Baum-Welch algorithm (Expectation-Maximization):

E-step: Compute forward-backward probabilities

$$\alpha_t(i) = P(\mathbf{O}_1, ..., \mathbf{O}_t, S_t = i | \lambda) \tag{10}$$

$$\beta_t(i) = P(\mathbf{O}_{t+1}, ..., \mathbf{O}_T | S_t = i, \lambda) \tag{11}$$

M-step: Update parameters

$$\xi_t(i,j) = \frac{\alpha_t(i) A_{ij} B_j(\mathbf{O}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) A_{ij} B_j(\mathbf{O}_{t+1}) \beta_{t+1}(j)} \tag{12}$$

5.4 State Sequence Estimation

Use Viterbi algorithm to find most likely state sequence:

$$V_t(i) = \max_j V_{t-1}(j) A_{ji} B_i(\mathbf{O}_t) \tag{13}$$

5.5 Alternative Methods

5.5.1 Method 2: Autoregressive Classification

Include lagged WSI values as features:

$$\mathbf{X}_t = [\text{WSI}_t, \text{WSI}_{t-1}, ..., \text{WSI}_{t-24}, \text{features}_t]$$

Use logistic regression:

$$P(\text{unstable}_t|\mathbf{X}_t) = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^{p} \beta_i X_{i,t})}}$$

5.5.2 Method 3: Exponentially Weighted Moving Average (EWMA)

$$\text{WSI}_{\text{smoothed},t} = \alpha \cdot \text{WSI}_t + (1-\alpha) \cdot \text{WSI}_{\text{smoothed},t-1} \tag{14}$$

where $\alpha \approx 0.2$ provides 5-hour memory.

Validation Criteria

Statistical Validation of Temporal Dependence:

- **Autocorrelation Analysis:** Compute autocorrelation function (ACF) of classified states

$$\text{ACF}(k) = \frac{\sum_{t=1}^{T-k}(S_t - \bar{S})(S_{t+k} - \bar{S})}{\sum_{t=1}^{T}(S_t - \bar{S})^2}$$

- **Regime Duration Statistics:** Report average regime duration

$$\text{Avg Duration} = \frac{1}{N_{\text{regimes}}} \sum_{i=1}^{N_{\text{regimes}}} \text{duration}_i$$

- **Persistence Validation:** Compare observed persistence with climatological expectations

6 Problem 4: Mathematical Validation of Error-Stability Relationship

Problem Statement

Problem: When we reach the point where we have the two parallel lines - one with the errors for each model and the line of periods determined with stability - what metric can we use to have the point proven mathematically that the error points corresponding to the unstable periods have much more error than the points corresponding to the stable points?

Solution

Solution: Four complementary statistical tests with effect size calculations and multiple testing corrections to mathematically validate the error-stability relationship.

6.1 Statistical Tests for Proving Error Difference

6.1.1 Test 1: Mann-Whitney U Test (Primary, Non-parametric)

Null Hypothesis: $H_0 : \text{MAE}_{\text{stable}} = \text{MAE}_{\text{unstable}}$

Alternative Hypothesis: $H_1 : \text{MAE}_{\text{unstable}} > \text{MAE}_{\text{stable}}$ (one-tailed)

Test Statistic:

$$U = n_1 n_2 + \frac{n_1(n_1 + 1)}{2} - R_1 \tag{15}$$

where R_1 is the sum of ranks in the stable group.

Why This Test: Robust to non-normal error distributions, no assumptions about variance equality, appropriate for skewed error distributions common in forecasting.

6.1.2 Test 2: Welch's t-test (Parametric Alternative)

Test Statistic:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{16}$$

Degrees of Freedom:

$$\text{df} = \frac{(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2})^2}{\frac{s_1^4}{n_1^2(n_1-1)} + \frac{s_2^4}{n_2^2(n_2-1)}} \tag{17}$$

6.1.3 Test 3: Linear Mixed-Effects Model

Mathematical Framework

Model Specification:

$$\text{MAE}_{ijt} = \beta_0 + \beta_1 \cdot \text{WSI}_t + \beta_2 \cdot \text{Model}_i + \beta_3 \cdot (\text{WSI}_t \times \text{Model}_i) + u_j + \varepsilon_{ijt}$$

where:

- i: model index
- j: day (random effect)
- t: hour within day
- $u_j \sim N(0, \sigma^2_{\text{day}})$: day-level random effect
- ε_{ijt}: residual error with AR(1) structure

Why This Model: Accounts for temporal clustering, handles repeated measures design, controls for confounding variables, provides interaction effects.

6.1.4 Test 4: Permutation Test (Distribution-free Validation)

Procedure:

1. Randomly shuffle stability labels 10,000 times
2. Compute $\Delta_{\text{MAE}} = \text{MAE}_{\text{unstable}} - \text{MAE}_{\text{stable}}$ for each permutation
3. Calculate p-value: $p = \frac{\text{count}(\Delta_{\text{MAE,permuted}} \geq \Delta_{\text{MAE,observed}})}{10000}$

6.2 Effect Size Measures

Mathematical Framework

Cohen's d:

$$d = \frac{\mu_{\text{unstable}} - \mu_{\text{stable}}}{\sigma_{\text{pooled}}}$$

where $\sigma_{\text{pooled}} = \sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}$

Interpretation:

- Small: 0.2
- Medium: 0.5
- Large: 0.8

Percentage Increase:

$$\text{PI} = \frac{\text{MAE}_{\text{unstable}} - \text{MAE}_{\text{stable}}}{\text{MAE}_{\text{stable}}} \times 100\%$$

Practical Significance: 15% increase considered operationally meaningful.

6.3 Multiple Testing Correction

Bonferroni Correction:

$$\alpha_{\text{corrected}} = \frac{0.05}{n_{\text{tests}}} \tag{18}$$

False Discovery Rate (Benjamini-Hochberg):

1. Order p-values: $p_{(1)} \leq p_{(2)} \leq ... \leq p_{(m)}$
2. Find largest i such that $p_{(i)} \leq \frac{i}{m}\alpha$
3. Reject hypotheses with $p_{(j)} \leq p_{(i)}$ for $j = 1, ..., i$

7 Problem 5: Mathematical Definition of Model Robustness

Problem Statement

Problem: How do we mathematically determine the most robust model?

Solution

Solution: Five robustness metrics with Relative Performance Degradation (RPD) as the primary metric, combined with hierarchical ranking strategy and statistical validation.

7.1 Candidate Robustness Metrics

7.1.1 Metric 1: Relative Performance Degradation (RPD) - Recommended Primary

Mathematical Framework

$$\text{RPD}_{\text{model}} = \frac{\text{MAE}_{\text{unstable}} - \text{MAE}_{\text{stable}}}{\text{MAE}_{\text{stable}}} \times 100\%$$

Why Chosen as Primary:

- Normalizes for baseline accuracy
- Allows fair comparison across models with different baseline errors
- Lower RPD = more robust
- **Justification:** A model with 10% baseline MAE increasing to 12% (20% RPD) is more concerning than 50% MAE increasing to 55% (10% RPD)

7.1.2 Metric 2: Absolute Performance Degradation (APD)

$$\text{APD}_{\text{model}} = \text{MAE}_{\text{unstable}} - \text{MAE}_{\text{stable}} \tag{19}$$

Why Supplementary: Captures absolute operational impact, useful when baseline accuracy matters (e.g., grid operations).

7.1.3 Metric 3: Robustness Index (Composite Score)

$$\text{RI}_{\text{model}} = (1 - \text{normalized\_RPD}) \times (1 - \text{normalized\_MAE}_{\text{overall}}) \tag{20}$$

where normalization is min-max scaling to [0,1].

Justification: Combines accuracy and robustness, range [0, 1], higher is better, balances "good on average" vs "consistent across conditions".

7.1.4 Metric 4: Stability of Performance (Coefficient of Variation)

$$\text{CV}_{\text{model}} = \frac{\text{SD}(\text{MAE}_{\text{across\_regimes}})}{\text{Mean}(\text{MAE}_{\text{across\_regimes}})} \tag{21}$$

Why Not Primary: Penalizes consistently bad models equally to consistently good ones. Lower CV = more consistent performance. Useful as tie-breaker.

7.1.5 Metric 5: Skill Score Preservation

$$\text{Skill}_{\text{stable}} = 1 - \frac{\text{MAE}_{\text{model,stable}}}{\text{MAE}_{\text{persistence,stable}}} \tag{22}$$

$$\text{Skill}_{\text{unstable}} = 1 - \frac{\text{MAE}_{\text{model,unstable}}}{\text{MAE}_{\text{persistence,unstable}}} \tag{23}$$

$$\text{Skill}_{\text{preservation}} = \frac{\text{Skill}_{\text{unstable}}}{\text{Skill}_{\text{stable}}} \tag{24}$$

Why Supplementary: Measures if model maintains advantage over baseline, context-dependent on persistence model performance, values > 1 indicate maintained skill.

7.2 Recommended Combined Approach

Validation Criteria

Primary Ranking Criteria:

1. **Primary:** Rank by RPD (lowest = most robust)
2. **Secondary filter:** Minimum overall accuracy threshold ($\text{MAE}_{\text{overall}}$ < threshold)
3. **Tie-breaker:** Stability of Performance (CV)

Threshold Selection:

- **High accuracy:** $\text{MAE}_{\text{overall}} < 10\%$ of installed capacity
- **Medium accuracy:** $\text{MAE}_{\text{overall}} < 20\%$ of installed capacity
- **Baseline:** $\text{MAE}_{\text{overall}} < \text{MAE}_{\text{persistence}}$

7.3 Statistical Validation of Robustness Rankings

Bootstrap Confidence Intervals: For each model's RPD:

1. Resample with replacement 1000 times
2. Compute RPD for each bootstrap sample
3. Report 95% confidence interval: $[\text{RPD}_{2.5\%}, \text{RPD}_{97.5\%}]$

Significance Testing: Test if RPD differences between models are significant using pairwise t-tests with Bonferroni correction.

8 Mathematical Consistency Across All Five Problems

8.1 Unified Framework Integration

The five problems form an integrated validation pipeline:

1. **Temporal resolution** (Problem 2) determines input to stability classification
2. **Temporal dependence modeling** (Problem 3) produces final stability labels
3. **Stability classification formula** (Problem 1) assigns labels based on WSI + temporal context
4. **Error-stability validation** (Problem 4) tests if labels meaningfully separate model performance
5. **Robustness metrics** (Problem 5) quantify and rank models using validated error differences

8.2 Validation Pipeline

Algorithm 1 Unified Validation Pipeline

```
1: Input: Raw Weather Data (hourly)
2: Apply 6-hour windows → Aggregated Features
3: Apply GMM classification + HMM smoothing → Stability Labels
4: Merge with Model Predictions → Model Performance by Stability Regime
5: Apply Statistical tests → Validated Error Differences
6: Apply Robustness metrics → Model Rankings
```

8.3 Mathematical Assumptions and Limitations

Mathematical Framework

Key Assumptions:

1. **Spatial Independence:** Weather stations are spatially independent after aggregation
2. **Stationarity:** Weather statistics are stationary within 2024
3. **Causality:** Stability causes performance difference (not confounded)
4. **Sample Size:** Sufficient observations in each regime (aim for $n_{\text{stable}}, n_{\text{unstable}} > 1000$ hours each)

Limitations:

1. Single Year: Results may not generalize to other years
2. Germany-specific: May not apply to other geographic regions
3. Weather-dependent: Performance may vary with climate patterns
4. Model-specific: Results apply to tested models only

9 Implementation Details

9.1 Software Requirements

Python Libraries:

```python
# Core scientific computing
import numpy as np
import pandas as pd
import scipy.stats as stats
from scipy.signal import medfilt

# Machine learning
from sklearn.mixture import GaussianMixture
from sklearn.metrics import silhouette_score, calinski_harabasz_score
from sklearn.preprocessing import RobustScaler

# Time series analysis
from statsmodels.tsa.stattools import acf
from statsmodels.stats.diagnostic import acorr_ljungbox
import hmmlearn.hmm as hmm

# Statistical tests
import pingouin as pg
from statsmodels.stats.multitest import multipletests
```

R Alternative:

```r
# Mixed models
library(lme4)
library(nlme)

# Clustering
library(mclust)
```

```

# HMM
library(depmixS4)

# Statistical tests
library(coin)
library(perm)
```

9.2 Pseudocode Implementation

```
def compute_wsi(weather_features, window_size=6):
    """
    Compute Weather Stability Index with temporal smoothing
    """
    # Step 1: Robust normalization
    scaler = RobustScaler()
    features_normalized = scaler.fit_transform(weather_features)

    # Step 2: Compute instantaneous WSI
    wsi_instantaneous = np.mean(features_normalized, axis=1)

    # Step 3: Rolling window statistics
    wsi_windowed = []
    for i in range(len(wsi_instantaneous)):
        start_idx = max(0, i - window_size//2)
        end_idx = min(len(wsi_instantaneous), i + window_size//2 + 1)
        window_data = wsi_instantaneous[start_idx:end_idx]

        wsi_windowed.append({
            'mean': np.mean(window_data),
            'std': np.std(window_data),
            'trend': np.polyfit(range(len(window_data)), window_data, 1)[0]
        })

    # Step 4: Temporal smoothing
    wsi_smoothed = medfilt([w['mean'] for w in wsi_windowed], kernel_size=3)

    return wsi_smoothed

def classify_stability_gmm(wsi_values, max_components=3):
    """
    Classify stability using Gaussian Mixture Model
    """
    # Step 1: Model selection using BIC
    bic_scores = []
    models = []

    for n_components in range(1, max_components + 1):
        gmm = GaussianMixture(n_components=n_components, random_state=42)
        gmm.fit(wsi_values.reshape(-1, 1))
```

```
            bic_scores.append(gmm.bic(wsi_values.reshape(-1, 1)))
            models.append(gmm)

        # Select model with lowest BIC
        best_model = models[np.argmin(bic_scores)]

        # Step 2: Classification
        labels = best_model.predict(wsi_values.reshape(-1, 1))
        probabilities = best_model.predict_proba(wsi_values.reshape(-1, 1))

        # Step 3: Validation
        silhouette = silhouette_score(wsi_values.reshape(-1, 1), labels)
        calinski_harabasz = calinski_harabasz_score(wsi_values.reshape(-1, 1),
            labels)

        return labels, probabilities, silhouette, calinski_harabasz

def test_error_difference(mae_stable, mae_unstable):
    """
    Test if error differs between stable and unstable periods
    """
    results = {}

    # Test 1: Mann-Whitney U test
    u_stat, p_mw = stats.mannwhitneyu(mae_unstable, mae_stable,
                                      alternative='greater')
    results['mann_whitney'] = {
        'statistic': u_stat,
        'p_value': p_mw,
        'effect_size': pg.mannwhitney(mae_unstable, mae_stable)['r']
    }

    # Test 2: Welch's t-test
    t_stat, p_ttest = stats.ttest_ind(mae_unstable, mae_stable,
                                     equal_var=False)
    results['welch_ttest'] = {
        'statistic': t_stat,
        'p_value': p_ttest,
        'cohens_d': pg.ttest(mae_unstable, mae_stable)['cohen-d']
    }

    # Test 3: Effect sizes
    pooled_std = np.sqrt(((len(mae_stable)-1)*np.var(mae_stable) +
                          (len(mae_unstable)-1)*np.var(mae_unstable)) /
                         (len(mae_stable) + len(mae_unstable) - 2))

    cohens_d = (np.mean(mae_unstable) - np.mean(mae_stable)) / pooled_std
    percentage_increase = ((np.mean(mae_unstable) - np.mean(mae_stable)) /
                          np.mean(mae_stable)) * 100

```

```
    results['effect_sizes'] = {
        'cohens_d': cohens_d,
        'percentage_increase': percentage_increase
    }

    return results
```

9.3 Reproducibility Guidelines

Random Seeds:

```
# Set global random seed
np.random.seed(42)
random.seed(42)

# Set seeds for specific libraries
sklearn.utils.check_random_state(42)
```

Parameter Documentation:

```
# config/parameters.yaml
stability_classification:
  window_size: 6  # hours
  smoothing_factor: 0.2
  classification_threshold: 0.5
  max_components: 3

statistical_tests:
  alpha_level: 0.05
  multiple_testing: "bonferroni"
  bootstrap_samples: 1000
  permutation_samples: 10000

robustness_metrics:
  primary_metric: "RPD"
  accuracy_threshold: 0.15  # 15% of installed capacity
  confidence_level: 0.95
```

10 Expected Outcomes and Validation

10.1 Expected Outcomes

1. **Mathematically rigorous methodology** that can withstand peer review
2. **Transparent decision-making** with statistical justification at each step
3. **Reproducible framework** applicable beyond 2024 Germany data
4. **Operational recommendations** for model selection under different weather conditions

10.2 Validation Criteria

Statistical Validation:

- All p-values < 0.05 (with multiple testing correction)
- Effect sizes 0.3 (medium effect)
- Confidence intervals exclude null hypothesis
- Bootstrap validation confirms results

Methodological Validation:

- Cross-validation shows consistent results
- Sensitivity analysis confirms robustness
- Literature comparison validates approach
- Expert review confirms meteorological soundness

Practical Validation:

- Results align with operational experience
- Recommendations are implementable
- Performance improvements are meaningful ($>$10%)
- Framework generalizes to other contexts

11 References

11.1 Weather Regime Detection

1. Huth, R., et al. (2008). "Classifications of atmospheric circulation patterns: recent advances and applications." *Annals of the New York Academy of Sciences*, 1146(1), 105-152.
2. Michelangeli, P. A., et al. (1995). "Weather regimes: Recurrence and quasi stationarity." *Journal of the Atmospheric Sciences*, 52(8), 1237-1256.
3. Vautard, R. (1990). "Multiple weather regimes over the North Atlantic: Analysis of precursors and successors." *Monthly Weather Review*, 118(10), 2056-2081.

11.2 Statistical Methods

1. McLachlan, G., & Peel, D. (2000). *Finite mixture models.* John Wiley & Sons.
2. Rabiner, L. R. (1989). "A tutorial on hidden Markov models and selected applications in speech recognition." *Proceedings of the IEEE*, 77(2), 257-286.
3. Mann, H. B., & Whitney, D. R. (1947). "On a test of whether one of two random variables is stochastically larger than the other." *The Annals of Mathematical Statistics*, 18(1), 50-60.

11.3 Renewable Energy Forecasting

1. Giebel, G., et al. (2011). "The state of the art in short-term prediction of wind power: A literature overview." *ANEMOS.plus*, 1-100.

2. Antonanzas, J., et al. (2016). "Review of photovoltaic power forecasting." *Solar Energy*, 136, 78-111.

3. Zhang, Y., et al. (2019). "Short-term wind speed prediction based on spatial correlation and artificial neural networks." *Journal of Wind Engineering and Industrial Aerodynamics*, 186, 17-25.

11.4 Robustness Metrics

1. Hastie, T., et al. (2009). *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media.

2. Cohen, J. (1988). *Statistical power analysis for the behavioral sciences.* Routledge.

3. Benjamini, Y., & Hochberg, Y. (1995). "Controlling the false discovery rate: a practical and powerful approach to multiple testing." *Journal of the Royal Statistical Society*, 57(1), 289-300.

12 Conclusion

This document provides a comprehensive framework for addressing the five critical methodological challenges in weather stability analysis for renewable energy prediction. All methods are mathematically rigorous, statistically validated, and operationally relevant. The unified validation pipeline ensures consistency across all components, while the implementation details provide practical guidance for application.

The framework establishes a solid foundation for scientifically sound research into weather stability impacts on renewable energy forecasting, with clear operational implications for grid operators and energy planners.