Data Challenge 3: Finding Hidden Patterns in High Resolution Wind Flow Model Simulations

Smoky Mountains Computational Sciences Data Challenge (SMCDC22)
Proposed by OAK RIDGE National Laboratory
Team: Yuqiao Zhao, Anna Fortenberry, Yulin Wei, Wilson Wong, Spencer Smith
Mentor: Dr. Kwai Wong
Contents

1. Introduction (Background and datasets)
2. Question 1, pearson correlation coefficient & bias
3. Question 2, PCA & t-SNE, and latent space
4. Question 3, interpolations & Unet
Introduction

Background

Wind flow dynamics at micro-scale are of paramount importance in the wind energy industry.

- Historically, wind farm designs rely on precise measurements from few meteorological masts over an entire site.
- However, at microscale, wind flow dynamics can be very sensitive to the terrain irregularities and wind conditions can drastically change from one location to another even over small distances.
- Computational Fluid Dynamics (CFD) is a promising approach for assessing atmospheric flow properties over a domain of interest.
- Large Eddy Simulation (LES) is one of the most advanced mathematical models used in CFD for resolving turbulences at a reasonable cost.

Fig 1: Visualization of data: terrain-follow
Introduction

Data sources

1. **ERA5**: a global weather model at a resolution of ~30km with hourly estimates of atmospheric variables. For each site provided, the corresponding timeseries is provided.

2. **Large Eddy Simulation (LES)**: a simulation model driven by boundary conditions derived from ERA5 data and then resolve the local wind farm site wind field at much higher resolution in space and time.

Fig 2: Demonstration of ERA5 & LES datasets
Introduction

Datasets

- **ERA5**: hourly data for the year 2020. The data has been extracted at single point (-7.737°E, 39.7°N) since its **spatial resolution** is about 30km.

- **LES**: 100m height above ground level, with two different spatial resolutions:
  - high-resolution: 80m x 80m at 1H frequency
  - low-resolution: 160m x 160m at 1H frequency
# Introduction

## Datasets

<table>
<thead>
<tr>
<th>datasets</th>
<th>time</th>
<th>xf</th>
<th>yf</th>
<th>variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERA5</td>
<td>8784</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>LES (high-resolution)</td>
<td>8784</td>
<td>192</td>
<td>192</td>
<td>6</td>
</tr>
<tr>
<td>LES (low-resolution)</td>
<td>8784</td>
<td>96</td>
<td>96</td>
<td>6</td>
</tr>
</tbody>
</table>

- **‘time’**: Timestamps at 1H frequency
- **‘xf’**: Horizontal cartesian coordinate in meter of the simulated domain (West to East)
- **‘yf’**: Vertical cartesian coordinate in meter of the simulated domain (South to North).
Table 2: Variables of ERA5 & LES

<table>
<thead>
<tr>
<th>Variables (ERA5)</th>
<th>Variable interpretations</th>
<th>Variables(LES)</th>
<th>variable interpretations</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘t2m’</td>
<td>2 meter above ground level temperature in K</td>
<td>‘temp’</td>
<td>1H average of temperature in Kelvin</td>
</tr>
<tr>
<td>‘u100’</td>
<td>100 meter above ground level U wind component in m/s</td>
<td>‘u’</td>
<td>1H average of U component of wind speed (along ‘xf’) in m/s</td>
</tr>
<tr>
<td>‘v100’</td>
<td>100 meter above ground level V wind component in m/s</td>
<td>‘v’</td>
<td>1H average of V component of wind speed (along ‘yf’) in m/s</td>
</tr>
<tr>
<td>‘i10fg’</td>
<td>10 meter above ground level instantaneous wind gust</td>
<td>‘vel’</td>
<td>1H average of horizontal wind speed in m/s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘std’</td>
<td>1H average of standard deviation of horizontal wind speed in m/s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘absolute_height’</td>
<td>Height above sea level in meter, note that this variable only depends on (xf, yf) not on time</td>
</tr>
</tbody>
</table>
Introduction

Fig 4: ‘i10fg’ of ERA5 (left), ‘vel’ of LES at a fixed time (right)
Question 1 (Compare ERA5 & LES simulation)

- Is there any **systematic bias**?
- What’s the **correlation**?

**Methods:**

\[
\text{bias} = \frac{\sum (x - y)}{\text{number of } x}
\]

Pearson correlation coefficient:

\[
r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}}
\]
LES (2D+time) \rightarrow \text{reduce dimension} \rightarrow \text{LES (1D)} \leftrightarrow \text{ERA5 (time)}

\text{LES (8784*96*96)} \rightarrow \text{flatten} \rightarrow \text{LES (8784*9216)} \rightarrow \text{PCA} \rightarrow \text{LES (8784)}
Conclusion:
● LES is *unbiased* since the bias of all variables are small
● LES has **strong linear correlation** with ERA5, since the magnitudes are all close to one

Results:

Table 3: Bias and correlation coefficient of ERA5 & LES

<table>
<thead>
<tr>
<th>ERA5 v.s. High_res</th>
<th>temp (t2m &amp; temp)</th>
<th>Wind speed (i10fg &amp; vel)</th>
<th>U component of wind speed (u100 &amp; u)</th>
<th>V component of wind speed (v100 &amp; v)</th>
</tr>
</thead>
<tbody>
<tr>
<td>bias</td>
<td>-2.13 × 10^{-5}</td>
<td>-1.66 × 10^{-7}</td>
<td>3.48 × 10^{-8}</td>
<td>5.29 × 10^{-8}</td>
</tr>
<tr>
<td>correlation</td>
<td>0.8833</td>
<td>0.8065</td>
<td>-0.9677</td>
<td>0.9561</td>
</tr>
</tbody>
</table>
Question 2 (compress to a lower dimension)
- Compare standard method and deep learning method.
- What’s the interpretability & visual insights of the latent space?

Methods:

Principal Component Analysis (PCA) \( \text{v.s.} \) t-distributed stochastic neighbor embedding (t-SNE)
PCA (Principal Component Analysis)

Linear dimensionality reduction using **Singular Value Decomposition (SVD)** of the data to **project** it to a lower dimensional space.

t-SNE (t-distributed stochastic neighbor embedding)

It is a **non-linear** dimensionality reduction technique that is particularly suited for the visualization of high-dimensional datasets.

**Scaled score** = score / sum of all scores

![Diagram of t-SNE process]

1. Calculate "unscaled similarity" score of the points in original dimension.
2. Scale them to get "similarity matrix 1".
3. Project points onto a line randomly.
4. Calculate "unscaled similarity" score of the points in the line.
5. Scale them to get "similarity matrix 2".
6. Move points to make the matrix 2 like matrix 1.

Resource: [https://www.youtube.com/watch?v=NEaUSP4YerM](https://www.youtube.com/watch?v=NEaUSP4YerM)
Compare PCA and t-SNE (1D)

Reduce one-month high-resolution data with “vel” variable to 1D (744*192*192 → 744) by PCA and t-SNE independently (blue line), and compare with enlarged ERA5 (“i10fg”, red lines). Multiple “i10fg” by 100 times and 10 times respectively, to better compare the shape of compressed data and true data.

Conclusion: With similar shapes around peaks, PCA preserves most information of the original data, but t-SNE keeps some information.
Compare PCA and t-SNE (2D)

Reduce one-month high-resolution data to 2D (36864*6 → 36864*2)

The image after PCA compressing forms **one cluster**.

The t-SNE compressed image is divided into **different clusters**.

Conclusion: t-SNE **extract the features** of the original data, but PCA does not.
Choose wind power density as the indicator because it combines variables “temp” and “vel” together, and it is important to the wind energy industry. Wind turbines convert the kinetic energy in the wind into mechanical power.

Wind power density \( (\text{w/m}^2) \) \( \text{wp/s} = \frac{1}{2} \ast \rho \ast v^3 \)

\( \text{wp} \) - wind power, \( s \) - unit area, \( \rho \) - air density, \( v \) - wind speed

Table 4: Wind energy scale at elevation of 100m

<table>
<thead>
<tr>
<th>Wind Power Class</th>
<th>Wind Power Density (W/m$^2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1 (Poor)</td>
<td>0-114.87</td>
</tr>
<tr>
<td>C2 (Marginal)</td>
<td>114.87-172.31</td>
</tr>
<tr>
<td>C3 (Fair)</td>
<td>172.31-229.75</td>
</tr>
<tr>
<td>C4 (Good)</td>
<td>229.75-287.19</td>
</tr>
<tr>
<td>C5 (Excellent)</td>
<td>287.19-344.62</td>
</tr>
<tr>
<td>C6 (Outstanding)</td>
<td>344.62-459.50</td>
</tr>
<tr>
<td>C7 (Superb)</td>
<td>459.50-1148.75</td>
</tr>
<tr>
<td>C8 (Out of Superb)</td>
<td>over 1148.75</td>
</tr>
</tbody>
</table>

Conclusion:

- The multiple clusters in (a) and (c) are due to “absolute_height”.
- Each cluster in all figures separates different classes clearly.

<table>
<thead>
<tr>
<th>Table 5: variables included in each figure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>a            b            c            d</td>
</tr>
<tr>
<td>“vel”        ✓            ✓            ✓            ✓</td>
</tr>
<tr>
<td>“temp”       ✓            ✓            ✓            ✓</td>
</tr>
<tr>
<td>“std”        ✓            ✓            ✓            ✓</td>
</tr>
<tr>
<td>“u”          ✓            ✓            ✔            ✔</td>
</tr>
<tr>
<td>“v”          ✓            ✓            ✔            ✔</td>
</tr>
<tr>
<td>“absolute_height” ✓            ✓            ✔            ✔</td>
</tr>
</tbody>
</table>
Observation: The **boundaries become smoother** without “u”, “v” variables
Observations:
- The boundaries become smoother without “u”, “v” variables.
- Different classes are separated clearly without “absolute_height” variable.
- The proportion of data in each class can be observed intuitively.
latent space

- 5 variables (drop “absolute_height”)

Conclusion: Different classes are no longer separated by ribbons with over 4 hours data as input, but begin to form spherical.
Conclusion: The strength of wind power density on different days can be told by latent space of 24 hours data.
Question 3 (Upscaling from a low-resolution to high-resolution grid)

- Can we make a **prediction** of a high-resolution dataset based on a low-resolution input dataset?
- What’s the **accuracy** of upscaling?

**Methods:** Interpolation (nearest-neighbor, bilinear, and bicubic interpolations) & Unet

Interpolation is a statistical method using related **known** values to **estimate unknown** values.

Fig 6: Simple demonstration of upscaling
### Table 6 Comparison of interpolation methods

<table>
<thead>
<tr>
<th>Interpolation method</th>
<th>Nearest neighbor interpolation</th>
<th>Bilinear interpolation</th>
<th>Bicubic interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pixel value</strong></td>
<td>Use value of nearest pixels</td>
<td>Use weighted average of two pixels</td>
<td>Use weighted average of four pixels</td>
</tr>
<tr>
<td><strong>Subjective Feelings</strong></td>
<td>Mosaic phenomenon</td>
<td>Blurring, not sharp</td>
<td>Sharper and fuzzy</td>
</tr>
<tr>
<td><strong>Image visibility</strong></td>
<td>Not clear</td>
<td>Jaggy, not clear</td>
<td>Better than bilinear</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>Worst</td>
<td>Poor</td>
<td>Better</td>
</tr>
<tr>
<td><strong>Computation time</strong></td>
<td>Less</td>
<td>Less than bicubic</td>
<td>more</td>
</tr>
<tr>
<td><strong>Speed</strong></td>
<td>Simple and fast</td>
<td>Slightly slower</td>
<td>fast</td>
</tr>
</tbody>
</table>

**Fig 7**: Patterns formed by different interpolation methods

source: https://www.semanticscholar.org/paper/Survey-on-Image-Interpolation-Kaur-Kaur/bad7a7dde3c13d6bfd7bbddfc3455022854b4934
### Table 7: Performance of different interpolation methods

<table>
<thead>
<tr>
<th></th>
<th>Nearest-neighbor interpolation</th>
<th>Bilinear interpolation</th>
<th>Bicubic interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>23.48</td>
<td>24.12</td>
<td>24.13</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.64</td>
<td>0.67</td>
<td>0.68</td>
</tr>
</tbody>
</table>

- **Peak signal-to-noise ratio (RSNR)**

\[
PSNR = 10 \cdot \log_{10}\left(\frac{\text{MAX}_I^2}{\text{MSE}}\right)
\]

- **Structural similarity (SSIM)**

\[
\text{SSIM}(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}
\]

---

Fig 8: upscaling of ‘vel’ variable using bicubic interpolation
Unet Method

- Normally use for segmentation
- Extract features of the image
Unet Method

Network Architecture

Concatenation for feature extraction
ResNet Unet
- Add one more upsampling layer

From 96 x 96 to 192 x 192
### Result compare to bicubic interpolation

<table>
<thead>
<tr>
<th></th>
<th>Unet</th>
<th>Bicubic interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>16.88</td>
<td>24.13</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.49</td>
<td>0.68</td>
</tr>
</tbody>
</table>
Unet Method

- Upscaled 'vel'
- High-resolution 'vel'
Possible reasons

- Not enough training (140 epoch for 8hrs)
- ResNet Unet is not good for image upsampling
- Parameter settings:
  - Learning rate
  - Optimization algorithm
**USRNet**


**Input:**

- LR image
- Estimated Kernel:
  - kernel width = 0.01
- Estimated Noise Level:
  - sigma = 4.5
- Scale factor = 2
DASR


Input:

- LR image only
Super-resolution Method

Results

DASR
PSNR 22.71dB
SSIM 0.7087

USRNet
PSNR 23.06dB
SSIM 0.7430

Interpolation
PSNR 23.82dB
SSIM 0.7410

Ground Truth
Possible reasons

- Model normally use for general photos
- Looking clear ≠ good performance
- Not enough training
Reference

- https://www.youtube.com/watch?v=NEaUSP4YerM
- https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a
- https://www.nature.com/articles/s41467-019-13056-x
- http://educypedia.karadimov.info/library/Lesson1_windenergycalc.pdf
- https://www.grc.nasa.gov/www/k-12/airplane/atmosmet.html
- https://www.researchgate.net/publication/343382220_An_Evaluation_of_the_Wind_Energy_Resources_along_the_Spanish_Coastal_Nearshore
- https://www.semanticscholar.org/paper/Survey-on-Image-Interpolation-Kaur-Kaur/bad7a7d7c83b69b7bbd4b34550228548b4934
Thank you

Q&A