Probing Particle Impingement in Boilers Using Computational Fluid Dynamics (CFD) with ANSYS Fluent

Steve Yang and Hyuna Kwon
Supervisor: Dr. Bryan Wong
University of California, Riverside
5/24/2021
Outline

1. Introduction
   a) Background
   b) Computational Fluid Dynamics (CFD) with ANSYS Fluent
   c) Concept of convolutional neural network (CNN) model in CFD

2. Research Objective

3. Preliminary results
   a) Optimizing mesh density & erosion model
   b) Automatization of the simulation with Ansys Fluent and biocluster
   c) CNN models for particle position and velocity prediction

4. Future Plans
   a) Constructing CNN models for erosion prediction in the boiler header
   b) Conclusion and the summary of the research progress
Predicting solid particle erosion

- Solid particle erosion: material damage caused by solid particles that impinge a surface

- Resulting in inefficient processes and financial instability.

- **Boiler header** is a pipeline that transfers steam from boiler heads to plant

However, erosion is a complex phenomenon
Predicting solid particle erosion

Previous attempts to predict erosion rate

Extensive experiments and field studies

Computational fluid dynamics (CFD)

Machine learning approach

Experiments

Computational fluid dynamics (CFD)

Neural Network

UCR
Barker and Gressley-Group College of Engineering
Factors affecting erosion rate

(1) Particle velocity

\[ ER \propto V_p^n \]

- \( V_p \) : particle velocity
- Various values of \( n \) have been proposed:
  - Finnie: \( n = 2 \)
  - Oka: \( n \) = function of hardness

(2) Impingement angle

(3) Particle properties (hardness, size, shape, etc.)

Oka and McLaury developed erosion models incorporating particle properties

\[
ER = \sum_{p=1}^{N_{\text{traj}}} \frac{m_p C(d_p) f(\alpha) v_p^n}{A_{\text{face}}}
\]

- \( m_p \): Mass flow rate of the particles
- \( f(\alpha) \): Impact angle function
- \( v_p \): Particle impact velocity
- \( n \): Velocity exponent
- \( C(d_p) \): Particle diameter function
CFD-based erosion modeling

(1) Pros & Cons

- Low-cost, fast solution
- Complex task because it involves various phenomena that must be considered

(2) Procedure

Fluid flow modeling → Discrete particle modeling → Erosion modeling
- Used ANSYS Fluent v 19.2 to generate data

- Finite-volume CFD code that solves the Navier-Stokes equation to model fluid flow

\[ \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = \frac{\mathbf{F}}{\rho} - \frac{1}{\rho} \nabla p \]

\[ \frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_j} (\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[ \mu \frac{\partial u_i}{\partial x_j} - \rho u_i \frac{\partial u_i}{\partial x_j} - \rho u_i u_j' \right] + \rho g_i + F_i \]

- ANSYS automated meshing was used to discretize the domain
Setup variables in ANSYS Fluent

- Average Mesh area: 7.8985e-003 m²

- Variables
  - Viscous: SST k-omega
  - Fluids: air / water vapor
  - Soild: steel
  - Inert Particle type: Anthracite
  - Particle diameter: 2e-05 m
  - Particle velocity magnitude: 45m/s
  - Particle flow rate: 0.00454 kg/s

- Inlet conditions (inlet)
  - Inlet pressure: 70000 Pa
  - Inlet velocity: 12.45 m/s

- Small inlet conditions (inlet_1, inlet_2, inlet_3)
  - Small inlet pressure: 200000 Pa
  - Small inlet pressure: 45m/s

- # of iterations: 200
Outline of this research

- Hybrid approach of combining CFD and machine learning models to build a predictive modeling pipeline for erosion rates.
- ANSYS Fluent is used to simulate fluid flow and particle transport on a boiler header.
- Exploratory data analysis was performed on particle trajectories and parameters to identify their significance in erosion rate predictions.
- CFD output is used as input to develop CNN model.
Variables for the CFD simulation

- **System setup variables**
  - Particle size
  - Particle velocity
  - Particle flow rate
  - Pressure of the main inlet
  - Velocity at the main inlet
  - Pressure of the branched inlet
  - Velocity at the branched inlet

- **Dependent variables**
  - Particle position vectors
  - Particle velocity vectors
  - Erosion rate
  - Erosion location

- Too many variables which are directly correlated with the result. So automatization of the variable settings and simulation running procedure is necessary.

- The more data available, the higher performing model can be obtained.

**Artificial neural network**
Convolutional Neural Network (CNN) model

• CNN: a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery by summarizing the presence of features in an input data.
# Comparison of CNN for Video recognition and CFD

The input shape (data dimension) of the video recognition with CNN is very similar to the input of the erosion prediction with computational fluid dynamics.

<table>
<thead>
<tr>
<th>Input shape</th>
<th>Video recognition</th>
<th>CFD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch</td>
<td>Number of videos</td>
<td>Number of simulations</td>
</tr>
<tr>
<td>In_depth</td>
<td>Number of pictures at a time period</td>
<td>Time steps for particle position and velocity</td>
</tr>
<tr>
<td>In_height</td>
<td>Number of pixels</td>
<td>Number of particles per one simulation</td>
</tr>
<tr>
<td>In_width</td>
<td>Number of pixels</td>
<td>Cartesian coordinates of particle position and velocity</td>
</tr>
<tr>
<td>In_channels</td>
<td>Red, green, blue</td>
<td>1 channel</td>
</tr>
<tr>
<td>Function approximation</td>
<td>Classification</td>
<td>Regression</td>
</tr>
</tbody>
</table>
For the CNN model, the whole data should be divided into ‘test set’ and ‘training set’ for minimizing the effects of data discrepancies and better understanding of the characteristics of the model.

K-fold cross validation provides a robust estimate of the performance of a model on unseen data. It does this by splitting the training dataset into k subsets and takes turns training models on all subsets except one which is held out, and evaluating model performance on the held out validation dataset.
Particle position and velocity prediction

Hypothesis: The erosion rate and the location is predictable if the information of position and velocity after the collision can be predicted by CNN.

The factors on the general formula are deeply related to the trajectory of the particle after the collision.

This will provide an insight of the damage mechanism on the geometry.
Artificial neural network model in CFD
Research Objective

• 1. Providing a new computational analysis (with CNN) to identify and develop insight into the inefficiencies of specific physical processes in existing coal plants

• 2. Predicting Damage Rates of Headers and turbines under different cycling models with artificial neural network model.
   1) Constructing a convolutional neural network model for the particle position and velocity prediction
   2) Developing the convolutional neural network model to predict the erosion location and erosion rate

• 3. Generating numerous dataset with changing system setup variables on Ansys fluent and creating multiple input CNN for high performing erosion prediction model
Preliminary results
Mesh density optimization

• Optimized mesh number = 500,000
• Finnie model underestimates ER
• McLaury model overestimates ER
• Finnie model is more accurate, especially at high flow velocity

Calculation is running in the HPCC in UCR

Data saved in the directory folder automatically
Erosion rate, particle position and velocity data of 180 particles (x, y, z, u, v, w)

The data is saved in these csv files with different 10 time steps
Univariate CNN model for particle position X prediction

The saved data in these csv files with different 10 time steps

Total 180 particles (particle 901 to particle 1081)
Constructing simple CNN model for particle position prediction, input array is [1st ..... 9th] of x_position and the **output** is [10th] x_position
Univariate CNN model summary

**Sequential**: A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor.

**Conv1D**: Convolution layer is to extract features from an input data.

**Maxpooling1D**: reducing the spatial size of the convoluted feature with the maximum value in the pool.

**Flatten**: Flatten the input.

**Dense**: Dense layer is the regular neural network layer.
Result of particle position prediction with the univariate CNN model

Number of K-fold splits = 10
Activation function = softmax
Epochs = 1000

$$R^2 = \frac{\text{Variance explained by the model}}{\text{Total variance}}$$

R-squared is always between 0 and 100%:

- 0% represents a model that does not explain any of the variation in the response variable around its mean. The mean of the dependent variable predicts the dependent variable as well as the regression model.
- 100% represents a model that explains all of the variation in the response variable around its mean.

<table>
<thead>
<tr>
<th></th>
<th>Average r2_score</th>
<th>Stdev of r2_score</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0.99</td>
<td>0.0031</td>
</tr>
<tr>
<td>Y</td>
<td>0.99</td>
<td>0.0007</td>
</tr>
<tr>
<td>Z</td>
<td>0.94</td>
<td>0.0008</td>
</tr>
</tbody>
</table>

Ref: statisticsbyjim
Constructing multivariate CNN model for particle position and velocity prediction, the input array is [1st ..... 9th] of (x, y, z, u, v, w) and the output is [10th] (x, y, z, u, v, w).

<table>
<thead>
<tr>
<th>particle</th>
<th>pos x</th>
<th>pos y</th>
<th>pos z</th>
<th>vel x</th>
<th>vel y</th>
<th>vel z</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>900</td>
<td>-0.061229</td>
<td>-0.100000</td>
<td>...</td>
<td>45.00000</td>
<td>0.000000</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>-0.061207</td>
<td>0.033156</td>
<td>...</td>
<td>45.21200</td>
<td>-0.191907</td>
<td>...</td>
<td>2</td>
</tr>
<tr>
<td>1080</td>
<td>0.054202</td>
<td>0.027245</td>
<td>...</td>
<td>31.29820</td>
<td>0.158833</td>
<td>...</td>
<td>3</td>
</tr>
<tr>
<td>180</td>
<td>0.034603</td>
<td>0.020200</td>
<td>...</td>
<td>-13.38570</td>
<td>0.449223</td>
<td>...</td>
<td>4</td>
</tr>
<tr>
<td>1260</td>
<td>0.004692</td>
<td>0.010090</td>
<td>...</td>
<td>-1.70672</td>
<td>-0.376443</td>
<td>...</td>
<td>5</td>
</tr>
<tr>
<td>360</td>
<td>0.031438</td>
<td>0.010926</td>
<td>...</td>
<td>1.71514</td>
<td>-2.14338</td>
<td>...</td>
<td>6</td>
</tr>
<tr>
<td>540</td>
<td>0.069629</td>
<td>0.013032</td>
<td>...</td>
<td>1.13121</td>
<td>-3.476570</td>
<td>...</td>
<td>7</td>
</tr>
<tr>
<td>1620</td>
<td>0.100000</td>
<td>0.016345</td>
<td>...</td>
<td>3.88938</td>
<td>-3.750080</td>
<td>...</td>
<td>8</td>
</tr>
<tr>
<td>720</td>
<td>0.100000</td>
<td>0.016345</td>
<td>...</td>
<td>3.88938</td>
<td>-3.750080</td>
<td>...</td>
<td>9</td>
</tr>
<tr>
<td>1440</td>
<td>0.100000</td>
<td>0.016345</td>
<td>...</td>
<td>3.88938</td>
<td>-3.750080</td>
<td>...</td>
<td>10</td>
</tr>
</tbody>
</table>
Result of the multivariate CNN model

Number of K-fold splits = 10
Activation function = softmax
Epochs = 1000

<table>
<thead>
<tr>
<th></th>
<th>Average r2_score</th>
<th>Stdev of r2_score</th>
</tr>
</thead>
<tbody>
<tr>
<td>x/u</td>
<td>0.90</td>
<td>0.05</td>
</tr>
<tr>
<td>y/v</td>
<td>0.91</td>
<td>0.01</td>
</tr>
<tr>
<td>z/w</td>
<td>0.71</td>
<td>0.01</td>
</tr>
<tr>
<td>xyz/uvw</td>
<td>0.84</td>
<td>0.03</td>
</tr>
</tbody>
</table>
Conclusion & Future Plans
Erosion location and erosion rate prediction based on the particle tracking data

The number of meshes (cells) on the surface geometry: 8012

Each cell has its own cartesian coordinates and the erosion rate after the calculation from Ansys fluent.

<table>
<thead>
<tr>
<th>[x, y, z, u, v, w]</th>
<th>time steps</th>
<th>Particle #</th>
<th># of simulations</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>50</td>
<td>180</td>
<td>1000</td>
</tr>
</tbody>
</table>

Input shape

Erosion prediction

CNN
Part of the erosion data sample

<table>
<thead>
<tr>
<th>Data dimension: 2D</th>
<th>X: (x, y, z, Erosion rate)</th>
<th>Y: Number of cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data dimension: 2D</td>
<td>X: (x, y, z, Erosion rate)</td>
<td>Y: Number of cells</td>
</tr>
<tr>
<td>7027, -7.229456252E-03, -8.475893457E-02, 6.794456389E-03, 0.000000000E+00</td>
<td>147</td>
<td></td>
</tr>
<tr>
<td>7028, 8.859659385E-02, -1.185526649E-02, -3.014118935E-02, 0.000000000E+00</td>
<td>7,865</td>
<td></td>
</tr>
<tr>
<td>7029, -9.933472754E-04, -6.668572991E-02, -1.634321178E-02, 0.000000000E+00</td>
<td>0.000000000E+00</td>
<td></td>
</tr>
<tr>
<td>7030, -1.150099804E-02, -2.909124298E-02, -2.218282419E-02, 0.000000000E+00</td>
<td>0.000000000E+00</td>
<td></td>
</tr>
<tr>
<td>7031, -6.130247936E-02, -5.302600230E-02, 7.324894234E-03, 0.000000000E+00</td>
<td>0.000000000E+00</td>
<td></td>
</tr>
<tr>
<td>7032, -7.957999075E-02, -6.670793135E-02, 1.147684461E-03, 0.000000000E+00</td>
<td>0.000000000E+00</td>
<td></td>
</tr>
<tr>
<td>1, 1.000000000E+00, -2.089699851E-02, -2.354808735E-02, 0.000000000E+00</td>
<td>0.000000000E+00</td>
<td></td>
</tr>
<tr>
<td>2, 1.000000000E+00, -2.300878490E-02, -2.243536897E-02, 0.000000000E+00</td>
<td>0.000000000E+00</td>
<td></td>
</tr>
<tr>
<td>3, 1.000000000E+00, -2.236575738E-02, 3.106292939E-02, 0.000000000E+00</td>
<td>0.000000000E+00</td>
<td></td>
</tr>
<tr>
<td>4, 1.000000000E+00, -2.380873442E-02, 2.861719975E-02, 0.000000000E+00</td>
<td>0.000000000E+00</td>
<td></td>
</tr>
<tr>
<td>5, 1.000000000E+00, -2.296154387E-02, -3.122163737E-02, 0.000000000E+00</td>
<td>0.000000000E+00</td>
<td></td>
</tr>
<tr>
<td>6, 1.000000000E+00, -1.925433566E-02, -2.853363300E-02, 0.000000000E+00</td>
<td>0.000000000E+00</td>
<td></td>
</tr>
<tr>
<td>7, 1.000000000E+00, -2.513686716E-02, -2.217791387E-02, 0.000000000E+00</td>
<td>0.000000000E+00</td>
<td></td>
</tr>
<tr>
<td>8, 1.000000000E+00, -2.722620850E-02, -2.421408999E-02, 0.000000000E+00</td>
<td>0.000000000E+00</td>
<td></td>
</tr>
<tr>
<td>9, 1.000000000E+00, -3.463008616E-04, -3.006517646E-02, 5.340282450E-05</td>
<td>0.000000000E+00</td>
<td></td>
</tr>
</tbody>
</table>

Maximum value of the erosion rate: $4.74 \times 10^{-4}$ kg/m$^2$ s

Erosion rate is too small for the input of CNN → Normalization

Normalized erosion rate

\[ x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)} \]
ReLU
\[ \max(0, x) \]

Rectified Linear unit, Computationally efficient
Non-linear (It has derivative function and allows Backpropagation)

But the network cannot perform backpropagation when the inputs approach zero. (Dying ReLU problem)

Softmax
\[ \sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \]

Softmax turns arbitrary real values into probabilities. The outputs of the Softmax transform are always in the range \([0, 1]\) and add up to 1. Hence, they form a probability distribution. (not good for erosion prediction)

tanh
\[ \tanh(x) \]

Non-linear activation function
Zero centered
Could have vanishing gradient problem
CNN model summary for erosion prediction

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output Shape</th>
<th>Param #</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv3d (Conv3D)</td>
<td>(None, 50, 180, 6, 10)</td>
<td>280</td>
</tr>
<tr>
<td>max_pooling3d (MaxPooling3D)</td>
<td>(None, 25, 90, 3, 10)</td>
<td>0</td>
</tr>
<tr>
<td>dropout (Dropout)</td>
<td>(None, 25, 90, 3, 10)</td>
<td>0</td>
</tr>
<tr>
<td>conv3d_1 (Conv3D)</td>
<td>(None, 25, 90, 3, 20)</td>
<td>1620</td>
</tr>
<tr>
<td>max_pooling3d_1 (MaxPooling3D)</td>
<td>(None, 13, 45, 2, 20)</td>
<td>0</td>
</tr>
<tr>
<td>dropout_1 (Dropout)</td>
<td>(None, 13, 45, 2, 20)</td>
<td>0</td>
</tr>
<tr>
<td>conv3d_2 (Conv3D)</td>
<td>(None, 13, 45, 2, 40)</td>
<td>6440</td>
</tr>
<tr>
<td>max_pooling3d_2 (MaxPooling3D)</td>
<td>(None, 6, 22, 1, 40)</td>
<td>0</td>
</tr>
<tr>
<td>flatten (Flatten)</td>
<td>(None, 5280)</td>
<td>0</td>
</tr>
<tr>
<td>dense (Dense)</td>
<td>(None, 8012)</td>
<td>42311372</td>
</tr>
</tbody>
</table>

- 3 Consecutive convolution3D & Maxpooling3D with padding with increasing number of filters
- Activation function used: tanh
- Flatten into dense layer with 8012 in order to get the dimension of the erosion input data

Epochs: 100
Number of K-fold splits = 5
CNN model evaluation method = r2_score
Summary

Hybrid approach of combining CFD and machine learning models to build a predictive modeling pipeline for erosion rates.

**CFD Modeling**
- Modeling fluid flow
- Modeling particle transport
- Input data:
  - CAD model for elbow
  - Initial flow rates
  - Boundary conditions
- Output:
  - Particle trajectory
  - Erosion rates

**Particle trajectory prediction**
- CNN model
  - Input:
    - Particle trajectories
  - Output:
    - CNN model for particle trajectory prediction

**Erosion rate prediction**
- CNN model
  - Input:
    - Initial conditions
    - Particle properties
    - Particle trajectories
    - Erosion rate
  - Output:
    - CNN model for erosion rate predictions
CFD Modeling
* Modeling fluid flow
* Modeling particle transport
  - Optimized mesh density
  - Optimized erosion model → Finnie
- Preliminary results
  - Connection angle vs. Sum erosion rate
    → Optimal connection angle = 40°
  - Pipe diameter vs. Sum erosion rate
    → Quadratic proportional

Particle trajectory prediction
* CNN model
* Input: Initial particle velocities/positions
* Output: Particle velocities/positions
* R2 score = 0.84

Done

Erosion rate prediction
* CNN model
* Input: Particle trajectories
* Output: Erosion rate

In progress
Conclusion

(1) CFD calculations to simulate fluid flow and particle transport in a boiler header.

(2) CFD results are considered as the input parameter

(3) A new model based on statistical analysis of CFD output parameters is developed.

- CNN models are explored to act as a black box

- CFD output is taken as input parameter and erosion is the output.

- A multilayer feed-forward network with a back-propagation algorithm is implemented

- CNN has potential to be developed further as a tool to predict erosion rate.
Future plan

- Any statistical model, more data points lead to better models
- This can be achieved by more CFD calculations
- Correlation analysis will be performed on CFD output parameters to identify ones affecting erosion rate prediction the most.

Particle size
Mass flow rate
Flow velocity
Surface shear stress
...

Time series particle tracking data

Multi-input CNN
Thank you!

Supported by UCR/HBCU DE-FE0031746
Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.