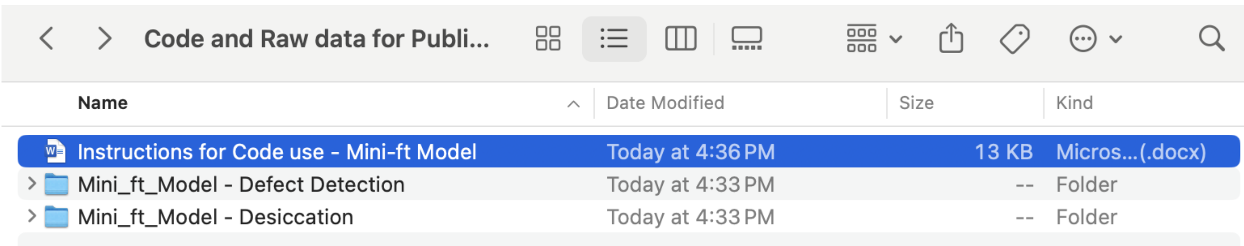
Instructions for Mini-ft model in "*AI-powered smart manufacturing of carbon-negative building materials*"

Welcome, thank you for downloading the data and source code behind the Mini-ft model. This post covers all the steps to re-create the plots and results for the Mini-ft model in the paper "*AI-powered smart manufacturing of carbon-negative building materials*". For any questions, please direct them to barneym@stanford.edu. Note the results shown in the paper were obtained when the code was run on Windows (there may be slight differences, when the code is run on Mac, such as for t-SNE).

Overview of files

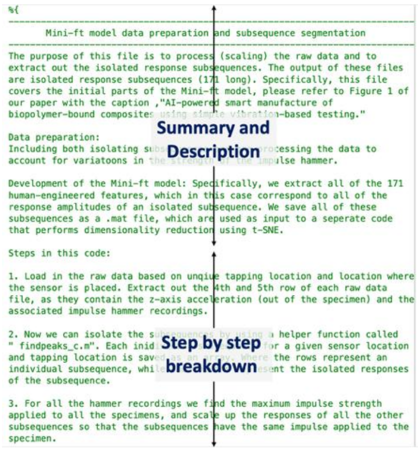


Upon downloading the files, there should be two other folders other than the README file (this one) in the main directory. "Mini_ft_Model - Defect Detection" contains the raw data (in a subfolder, located in this folder) and code for the Mini-ft model in relation to defect detection. "Mini_ft_Model – Desiccation" also contains the raw data and code, leading to the results shown in the paper.

Mini_ft_Model - Defect Detection

In the defect detection folder, open the file titled "Data_processing_and_subsequence_segmentation.m" or (for mac users) "Data_processing_and_subsequence_segmentation_mac_version.m".

When you open the file you will see the following breakdown in the summary:



This file essentially loads in the raw time series data obtained from the experiments, and performs the following steps (please refer to the script for more details about each step):

1. Load in the data based on the separate entities
2. Get the hammer hits and responses (Subsequence generation)
3. Find the maximum hammer strength of all the recordings
4. Apply Scaling to each specimen according to hammer impulse
5. Normalize the amplitudes of the response to be from 0 to 1
6. Saving the isolated subsequences for analysis on associated python file

Upon running the code, you will see the following pop-up. If you would like to get the isolated subsequences without any scaling, press 0, and if you would like to get the isolated subsequences after scaling has been performed, please enter in 1.

```
Want scaled features? [1=yes 0=no]: 1
Running scaling case...
Done running!
```

If you see this message at the end of the run, that means everything has been run successfully!

At the end of the code we are saving isolated subsequences as individual files starting with "features", these files will be the ones loaded in the next file titled "Dimensionality_Reduction.py".

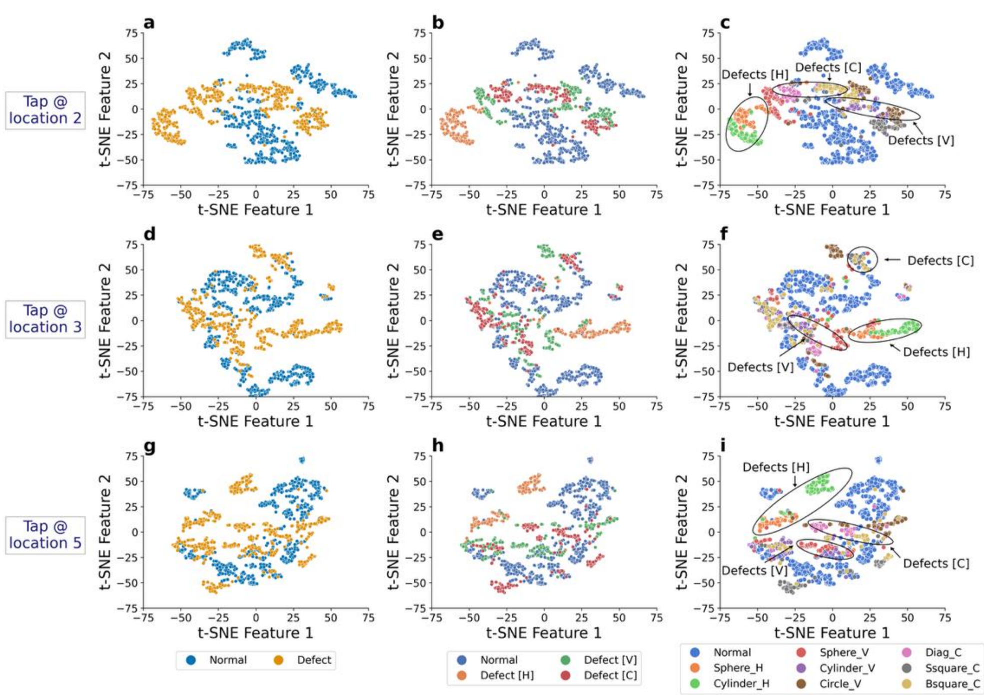
Upon opening the “Dimensionality_Reduction.py” file, we can run the file to perform t-SNE on the isolated subsequences. Upon running the files, the code will save the first and second t-SNE dimensions, which can be loaded later by the file titled “TSNE_Visualization,py” to plot Figure 3 in the paper. Note that fluctuations in the results are possible due to the stochastic nature of the t-SNE operation.

To obtain the t-SNE visualization according to 2-class, 4-class, and 9-class defect detection, you can change the variable titled “class_mode” to the desired class. By changing this variable, we are able to obtain all the tsne_features, which are used in the visualization file.

```python
# --- Configurable Section ---
# Choose class mode: '2', '4', or '9'
class_mode = '4'

# Define label setup based on class_mode
if class_mode == '2':
    non_normal_labels = [1] * 8
    label_suffix = '2cls'
elif class_mode == '4':
    non_normal_labels = [1, 1, 2, 2, 2, 3, 3, 3]
    label_suffix = '4cls'
elif class_mode == '9':
    non_normal_labels = list(range(1, 9))
    label_suffix = '9cls'
else:
    raise ValueError("Invalid class_mode. Use '2', '4', or '9'.")
```

Now after running for all three class modes, the final file titled “TSNE_Visualization,py” can be used to plot Figure 3 in the paper, which leads to the following.



Mini-ft model – desiccation

Similar to defect detection, run the code file titled "Data_processing_and_Subsequence_segmentation.m", and isolate the subsequences by running the code up until step 4 (performing the regression). Note, that scaling is automatically performed in this script (please see the output files below).

Data_processing_and_S...quence_segmentation.m	Today at 8:40 AM	12 KB	MATLAB Code
Dimensionality_Reduction.py	May 3, 2025 at 4:31 PM	2 KB	Python File
features_response_front_2hr.mat	May 3, 2025 at 4:42 PM	86 KB	MATLAB Data
features_response_front_3hr.mat	May 3, 2025 at 4:42 PM	77 KB	MATLAB Data
features_response_front_4hr.mat	May 3, 2025 at 4:42 PM	109 KB	MATLAB Data
features_response_front_5hr.mat	May 3, 2025 at 4:42 PM	92 KB	MATLAB Data
features_response_front_6hr.mat	May 3, 2025 at 4:42 PM	92 KB	MATLAB Data
features_response_front_7hr.mat	May 3, 2025 at 4:42 PM	92 KB	MATLAB Data
features_response_oppo_2hr.mat	May 3, 2025 at 4:19 PM	83 KB	MATLAB Data
features_response_oppo_3hr.mat	May 3, 2025 at 4:19 PM	79 KB	MATLAB Data
features_response_oppo_4hr.mat	May 3, 2025 at 4:19 PM	103 KB	MATLAB Data
features_response_oppo_5hr.mat	May 3, 2025 at 4:19 PM	84 KB	MATLAB Data
features_response_oppo_6hr.mat	May 3, 2025 at 4:19 PM	81 KB	MATLAB Data
features_response_oppo_7hr.mat	May 3, 2025 at 4:19 PM	92 KB	MATLAB Data
findpeaks_c.m	Feb 20, 2025 at 10:10 PM	790 bytes	MATLAB Code
Raw_Data	Yesterday at 4:33 PM	--	Folder
tsne_encoded_response_front.mat	Today at 8:33 AM	10 KB	MATLAB Data

Now proceed to the "Dimensionality_Reduction.py" file, and run the code to perform t-SNE on the isolated subsequences again. From here we can obtain "tsne_encoded_response_front.mat", and the visual for the Mini-inset window that was used in Figure 6.

Now we can return to the original file, and proceed with the code from step 4. Where, by running we can obtain the final results shown in the paper. Note that the results may fluctuate and result in slightly different results as shown in the paper.

```
Summary of 100 GPR Runs (Exponential Kernel):
Average MAE : 1.4268
Average MAPE: 2.42%
Average MSE : 12.9515
Average R²  : 0.9579
```