

# DeepEST: A Python Library for Spatio-Temporal Epidemiology Prediction



GitHub Code

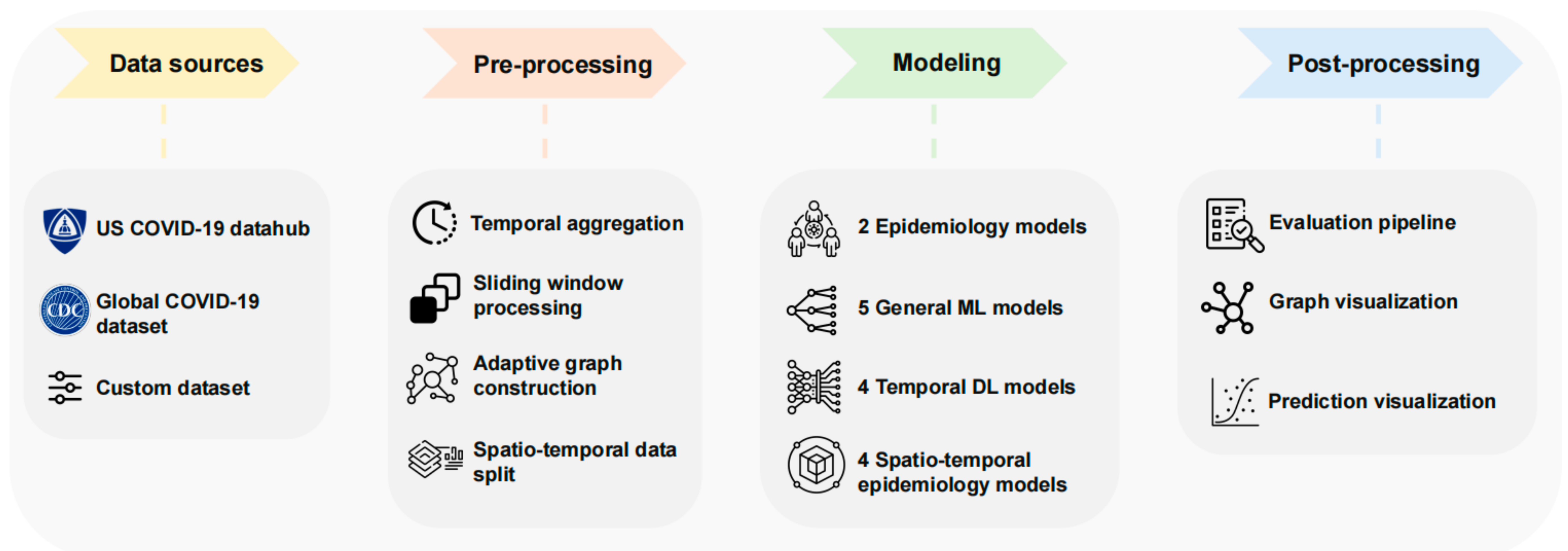
Yuhang Wang, Yinghao Zhu, Lifang Liang, Yasha Wang, Ewen M. Harrison, Liantao Ma, Junyi Gao†



HDRUK  
Health Data Research UK

- Traditional data sources in pandemic diseases field provide extensive spatio-temporal datasets, yet they underscore a need for sophisticated predictive tools that are accessible to public health researchers without deep technical expertise.
- Currently, there is no user-friendly spatio-temporal toolkit specifically designed for public health prediction.
- **Objective:** Build a **comprehensive** suite of tool which integrates advanced predictive **spatio-temporal modeling** or data preprocessing, model development, and visualization to reduce the **technical barriers** for researchers.

## Motivation



## Structure

Structure and pipeline of using DeepEST

### Deep Learning Based Models

### Machine Learning Based Models

Temporal	Spatio-temporal	General ML	Epidemiology
• RNN	• STAN	• Arima	• SIR
• GRU	• ColaGNN	• XGBoost	• SEIR
• LSTM	• HOIST	• LightGBM	
• Transformer	• TAP-RSV	• SVM	
		• Random Forest	

## Models

### Supported models in DeepEST

```
>>> from deepest import dl_trainer, dl_tester, build_model, preprocess_data, construct_adjacency_matrix
# Data was preprocessed with z-score normalization, setting input and output windows to 7 and 3
# respectively, and partitioned into training, validation, and test sets in a 3:1:1 ratio in temporal
# dimension.
>>> st_data = preprocess_data(dynamic_data, static_data, label, input_window=7, output_window=3,
# normalization="z-score", type="spatial-temporal")
# Get the data_loader with batch_size as 64
>>> trainset, valset, testset = st_data.get_dl_dataset()
>>> train_st_loader, val_st_loader, test_st_loader = get_dataloader(trainset, batch_size=64, shuffle=True),
# get_dataloader(valset, batch_size=64), get_dataloader(testset, batch_size=64)
# Calculate the Euclidean distance between static data of each node and filter nodes with a distance less
# than 0.5 to construct a graph.
>>> edge_index = construct_adjacency_matrix(static_data, distance_matrix="euclidean", threshold=0.5)
# Construct the model by preprocessed data and model name.
>>> model = build_model(preprocess=st_data, model_name="colagnn", edge_index=edge_index)
# build trainer and tester from model
>>> trainer = dl_trainer(model=model)
>>> tester = dl_tester(model=model)
# train the model for 10 epochs
>>> trainer.train(train_dataloader=train_st_loader, val_dataloader=val_st_loader, epochs=10)
# test the model, and get the MSE between true data and predict data
>>> tester.test(test_st_loader, metric="MSE")
```

## Code Example

Example code of using DeepEST to build, train and evaluate a spatio-temporal prediction model

# DeepEST: A Python Library for Spatio-Temporal Epidemiology Prediction



GitHub Code



Yuhang Wang, Yinghao Zhu, Lifang Liang, Yasha Wang, Ewen M. Harrison, Liantao Ma, Junyi Gao\*



GitHub Code



*GitHub Code*