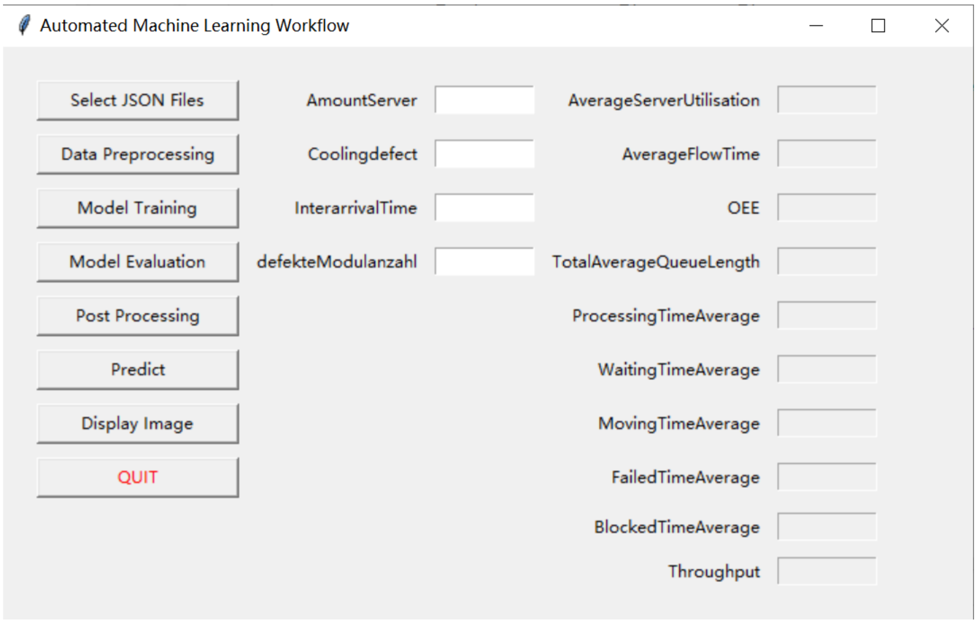
Welcome to the **ANN Approximation Framework User Guide**. This document is designed to help you understand and utilize the ANN Approximation Framework efficiently.

Overview of the User Interface

Upon running the main file, a straightforward interactive interface will be displayed, as shown in the image below. The interface consists of three columns:

1. **Left Column:** Lists all available functions within the framework.
2. **Middle Column:** Displays input parameters (writable).
3. **Right Column:** Displays output parameters (read-only).



Explanation of Functions

We will now go through each function listed in the left column of the interface:

Function 1: Select JSON Files

Click on **'Select JSON Files'** button to open a window where you can choose the data files you want to use.

Typically, these data files are stored in the results folder of the scenario analysis, usually located at '.\postprocessor\output\scenario_analysis_result.json'.

After selecting a data file, a new window will appear, prompting you to choose the purpose of the file. There are three options:

- **Train:** Use the file for training.

- **Test:** Use the file for testing.
- **Split:** Use a single data file for both training and testing.

After clicking **'Confirm'**, the system will ask if you wish to load additional data files. If no more files are needed, this function is complete.

Function 2: Data Preprocessing

Click on **'Data Preprocessing'** button to open a window where you can select the data scaling method: either **MinMaxScaler** for normalization or **StandardScaler** for standardization. While MinMaxScaler is the recommended option, users can choose based on their preferences.

Note that Function 2 can only be used after completing Function 1, as preprocessing requires selected data files.

Function 3: Model Training

Click on **'Model Training'** button to have the framework automatically fit an appropriate network structure based on the training data and train the network to obtain its weight parameters. The results of each experiment and the best model for each tuner will be saved in the respective tuner's folder (e.g., project_****).

Function 3 relies on the completion of Functions 1 and 2. This means you need to have selected and preprocessed the data for training before proceeding with model training.

Function 4: Model Evaluation

Click on **'Model Evaluation'** button to evaluate the model's performance using various metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R2), based on the test data.

To use Function 4, the following prerequisites must be met:

1. A test set must be defined using Function 1.
2. The test data must be preprocessed using Function 2.
3. A trained model must be available.

Function 5: Post Processing

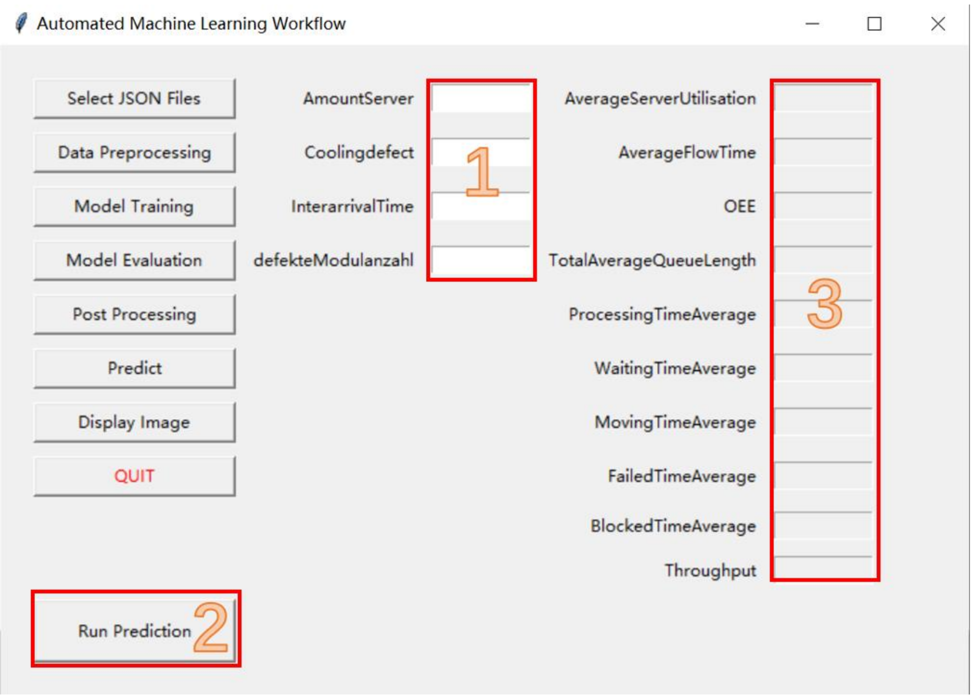
Click on **'Post Processing'** button to visualize the data generated during the training and validation processes. This function helps create supplementary images to better understand these processes.

Function 5 can only be used after completing Function 3. This means you need to have completed the training process and have access to the training process files.

Function 6: Predict

Click on **'Predict'** button to open a window prompting you to select the best model obtained from a specific tuner for the prediction task. The **'Run Prediction'** button will appear in the bottom right corner of the interface, indicating that the model has been loaded and is ready for use.

To perform a prediction, enter the appropriate values into the input parameters and click **'Run Prediction'**. The results will be displayed in the output parameters. The operational sequence is illustrated in the diagram below.



Function 6 requires a trained model from Function 3 to perform predictions based on custom inputs.

Function 7: Display Image

Click on **'Display Image'** button to select and display images of your choice within the interface.

Exiting the Framework

When you no longer wish to use this framework, simply click 'QUIT' to exit.