**Implementation:**

The first step is to set up the environment with the previously given instructions, then merging their codebases. The new folder structure becomes:

/model\_intergration/

/CommFormer/

/DHGAS/

The following steps follow:

· Loading and preprocessing the DHGAS dataset.

· Loading pre-trained models for DHGAS and CommFormer.

· Merging the models in TensorFlow to form a combined model.

· Training the combined model using the DHGAS dataset.

· Logging the training progress and evaluating the model.

**File Structure:**

normalize.py : To normalize data.

intergrated\_model.py : Will define the combined models.

rl\_environment.py: For reinforcement learnging.

train.py: For training the new model using RL.

validate.py: To validate the trained model.

config.yaml : Will store parameters for training.

A README.md and requirements.txt file will be included for a brief explanation of the project. The requirements file will list dependencies needed to be installed with pip.

**Loading and prepossessing the DHGAS dataset:**

The DHGAS has a data set containing 4 columns of numeric data, divided into training and test data . The data will be read and normalized so each element of data gives equal contribution, since the some values in the training data significantly vary in magnitude compared to the test set.

The normalized data will be stored separately

Sample train data :

23067537 1184664230 1 20190616

23067537 248930187 1 20190616

23067537 697449153 1 20190616

23067537 697449153 1 20190611

23067537 697449153 3 20190611

Sample test data:

58762 609331381 0 2

58762 228485031 0 2

58762 893436848 0 2

58762 279285297 0 2

Code to normalize:

import pandas as pd

import numpy as np

# Function to normalize data

def normalize\_data(df):

    # Min-Max Normalization

    return (df - df.min()) / (df.max() - df.min())

# Load training data from text file

train\_data\_path = 'ecomm\_edge\_train.txt'

train\_data = pd.read\_csv(train\_data\_path, sep='\t', header=None)

# Load test data from text file

test\_data\_path = 'ecomm\_edge\_val\_lr\_train\_test.txt'

test\_data = pd.read\_csv(test\_data\_path, sep='\t', header=None)

# Normalize training data

normalized\_train\_data = normalize\_data(train\_data)

# Normalize test data

normalized\_test\_data = normalize\_data(test\_data)

# Save normalized training data to new text file

normalized\_train\_data\_path = 'normalized\_train\_data.txt'

normalized\_train\_data.to\_csv(normalized\_train\_data\_path, sep='\t', header=False, index=False)

# Save normalized test data to new text file

normalized\_test\_data\_path = 'normalized\_test\_data.txt'

normalized\_test\_data.to\_csv(normalized\_test\_data\_path, sep='\t', header=False, index=False)

print("Normalization and saving completed.")

*Sample normalized test data:*

*0.0013350158433104809 0.7427272329205317 0.0 1.0*

*0.0038302687776674564 0.9506906007702923 0.0 1.0*

*0.0038302687776674564 0.8469793123290771 0.0 1.0*

*0.0038302687776674564 0.17931310457083133 0.0 1.0*

*0.0038302687776674564 0.6876208356175422 0.0 1.0*

*0.0038302687776674564 0.799239020773327 0.0 1.0*

*0.0038302687776674564 0.6750804500815432 0.0 1.0*