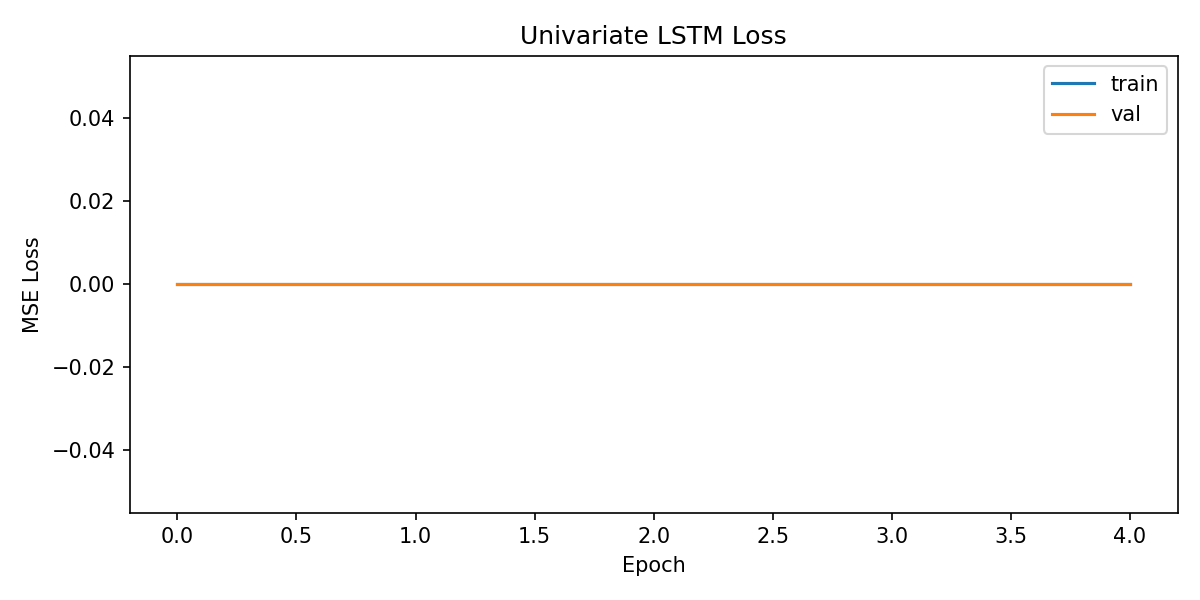
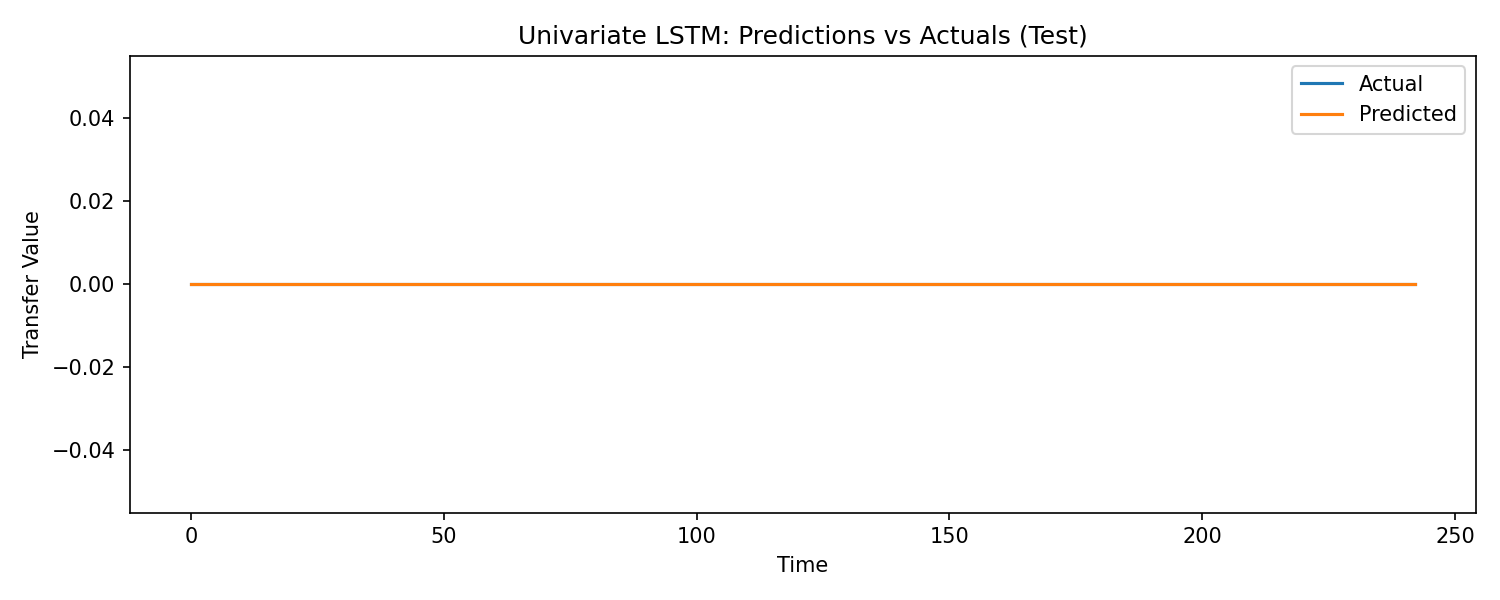
LSTM Univariate Model Report

# Training Loss Curve



This shows the training and validation loss over epochs.

# Predictions vs Actuals



Comparison of actual values and model predictions on the test set.

# Evaluation Metrics

{'rmse': 0.0, 'mae': 0.0}

# Code (lstm\_forecasting.py)

import argparse  
import os  
from typing import Dict, List, Optional, Tuple  
  
import numpy as np  
import pandas as pd  
from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error  
from sklearn.preprocessing import MinMaxScaler  
import matplotlib.pyplot as plt  
  
import tensorflow as tf  
from tensorflow.keras import Model  
from tensorflow.keras.layers import LSTM, Dense, Input, RepeatVector, TimeDistributed  
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint  
  
  
# ==========================  
# Data preprocessing utils  
# ==========================  
  
def fit\_feature\_target\_scalers(  
 df: pd.DataFrame,  
 feature\_columns: List[str],  
 target\_column: str,  
 feature\_range: Tuple[float, float] = (0.0, 1.0)  
) -> Tuple[MinMaxScaler, MinMaxScaler]:  
 """  
 Fit MinMax scalers for features and target separately.  
 """  
 feature\_scaler = MinMaxScaler(feature\_range=feature\_range)  
 target\_scaler = MinMaxScaler(feature\_range=feature\_range)  
  
 feature\_scaler.fit(df[feature\_columns].values)  
 target\_scaler.fit(df[[target\_column]].values)  
 return feature\_scaler, target\_scaler  
  
  
def transform\_features\_target(  
 df: pd.DataFrame,  
 feature\_columns: List[str],

# Model Weights

File: univariate\_lstm.weights.h5 (binary file not displayed).