1. Shamima Nasrin Tumpa et al., "Utilizing RNN for Real-time Cryptocurrency Price Prediction and Trading Strategy Optimization", 2024.

```python
# Install required libraries
!pip install yfinance numpy pandas matplotlib scikit-learn tensorflow

import yfinance as yf
import pandas as pd
import random
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import requests                                    #ADDED
from textblob import TextBlob                      #ADDED

data = pd.read_csv('Book.csv')          #ADDED

# Set seeds for reproducibility
random.seed(42)
np.random.seed(42)
tf.random.set_seed(42)

# Define the tickers for Bitcoin, Ethereum, and Litecoin
cryptos = ['BTC-USD', 'ETH-USD', 'LTC-USD']

# Define the start and end date for the data
start_date = '2019-01-01'
end_date = '2024-01-01'

# Create an empty dictionary to store the data
crypto_data = {}

# Download and store data in the dictionary
for ticker in cryptos:
    crypto_data[ticker] = yf.download(ticker, start=start_date,
end=end_date)

    # Display the first few rows of each dataframe
    print(f"Showing data for {ticker}:")
    display(crypto_data[ticker].head())

# Function to fetch sentiment data (example using Twitter API)
def fetch_sentiment_data(keyword, start_date, end_date):
#ADDED
    dates = pd.date_range(start=start_date, end=end_date)
    #sentiment_scores = np.random.uniform(-1, 1, len(dates))
#COMMENTED OUT
    sentiment_scores = [TextBlob(sentence).sentiment.polarity for
sentence in data['Tweet']]
    sentiment_data = pd.DataFrame({'Sentiment':
sentiment_scores},index=dates)                          #REMOVE
'Date': dates  #ADDED index=dates
```

```python
    #sentiment_data.set_index('Date', inplace=True)
#COMMENTED OUT
    return sentiment_data

# Fetch sentiment data for Bitcoin
btc_sentiment = fetch_sentiment_data('Bitcoin', start_date,
end_date='2023-12-31')                       #ADDED

from sklearn.preprocessing import MinMaxScaler

# Assuming we're working with Bitcoin data for this example
btc_data = crypto_data['BTC-USD']

# 1. Handle missing data
# For simplicity, we'll fill missing values with the last available
value
btc_data.fillna(method='ffill', inplace=True)



# Flatten the multi-level columns in the Bitcoin data
#ADDED
btc_data.columns = [f"{col[0]}_{col[1]}" for col in btc_data.columns]

# Rename columns by removing '_BTC-USD' suffix
#ADDED
btc_data.columns = [col.replace('_BTC-USD', '') for col in
btc_data.columns]

# Ensure both DataFrames have datetime index
#ADDED
btc_data.index = pd.to_datetime(btc_data.index)
btc_sentiment.index = pd.to_datetime(btc_sentiment.index)





#btc_data = btc_data.merge(btc_sentiment, left_index=True,
right_index=True)                    #ADDED       #COMMENTED OUT

# Merge Bitcoin data with sentiment data
btc_data = btc_data.join(btc_sentiment, how='left')


print(btc_data)
```

```python
# 2. Normalize the features
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data_btc = scaler.fit_transform(btc_data[['Open', 'High',
'Low', 'Close', 'Sentiment']])  #CHANGE: REMOVED 'Adj Close'
#ADDED 'Sentiment'

# 3. Convert the data into a time series format
def create_dataset(dataset, look_back=1):
    X, Y = [], []
    for i in range(len(dataset) - look_back - 1):
        #a = dataset[i:(i + look_back), :]
#WRONG
        a = dataset[i:(i + look_back), [0, 1, 2, 4]]  # Include
'Open', 'High', 'Low', and 'Sentiment'     #EXCLUDING TARGET
        X.append(a)
        Y.append(dataset[i + look_back, -2])  # Assuming close price
is the target variable   #ADDED CHANGE -1 TO -2 to refer to 'Close'
    return np.array(X), np.array(Y)

# Define look_back period
look_back = 60  # for example, using past 60 days to predict the next
day

# Split into features (X) and target (Y)
X_btc, Y_btc = create_dataset(scaled_data_btc, look_back)

# Reshape features into the format expected by LSTM (samples, time
steps, features)
X_btc = np.reshape(X_btc, (X_btc.shape[0], look_back, X_btc.shape[2]))

# Split into train and test sets
split_percent = 0.80
split = int(split_percent * len(X_btc))

X_train_btc = X_btc[:split]
Y_train_btc = Y_btc[:split]
X_test_btc = X_btc[split:]
Y_test_btc = Y_btc[split:]

print("Data preprocessed and ready for model training.")

# Import TensorFlow for model training
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, GRU, Bidirectional

# Define the LSTM model
lstm_model = Sequential()
```

```python
lstm_model.add(LSTM(units=100, return_sequences=True,
input_shape=(X_train_btc.shape[1], X_train_btc.shape[2])))
lstm_model.add(LSTM(units=100))
lstm_model.add(Dense(1))

lstm_model.compile(optimizer='adam', loss='mean_squared_error')

# Define the GRU model
gru_model = Sequential()
gru_model.add(GRU(units=100, return_sequences=True,
input_shape=(X_train_btc.shape[1], X_train_btc.shape[2])))
gru_model.add(GRU(units=100))
gru_model.add(Dense(1))

gru_model.compile(optimizer='adam', loss='mean_squared_error')

# Define the BiLSTM model
bilstm_model = Sequential()
bilstm_model.add(Bidirectional(LSTM(units=100, return_sequences=True),
input_shape=(X_train_btc.shape[1], X_train_btc.shape[2])))
bilstm_model.add(Bidirectional(LSTM(units=100)))
bilstm_model.add(Dense(1))

bilstm_model.compile(optimizer='adam', loss='mean_squared_error')

# Train the models
lstm_history = lstm_model.fit(X_train_btc, Y_train_btc, epochs=100,
batch_size=32, validation_data=(X_test_btc, Y_test_btc), verbose=1)
gru_history = gru_model.fit(X_train_btc, Y_train_btc, epochs=100,
batch_size=32, validation_data=(X_test_btc, Y_test_btc), verbose=1)
bilstm_history = bilstm_model.fit(X_train_btc, Y_train_btc,
epochs=100, batch_size=32, validation_data=(X_test_btc, Y_test_btc),
verbose=1)

# Evaluate the models
lstm_scores = lstm_model.evaluate(X_test_btc, Y_test_btc, verbose=0)
gru_scores = gru_model.evaluate(X_test_btc, Y_test_btc, verbose=0)
bilstm_scores = bilstm_model.evaluate(X_test_btc, Y_test_btc,
verbose=0)

print('LSTM Test loss:', lstm_scores)
print('GRU Test loss:', gru_scores)
print('BiLSTM Test loss:', bilstm_scores)

# Plot training and validation loss
plt.figure(figsize=(18, 5))
plt.subplot(1, 3, 1)
plt.plot(lstm_history.history['loss'], label='LSTM training Loss')
plt.plot(lstm_history.history['val_loss'], label='LSTM Validation
Loss')
```

```python
plt.title('LSTM Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()


plt.subplot(1, 3, 2)
plt.plot(gru_history.history['loss'], label='GRU training Loss')
plt.plot(gru_history.history['val_loss'], label='GRU Validation Loss')
plt.title('GRU Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()



plt.subplot(1, 3, 3)
plt.plot(bilstm_history.history['loss'], label='Bi-LSTM training
Loss')
plt.plot(bilstm_history.history['val_loss'], label='Bi-LSTM Validation
Loss')
plt.title('Bi-LSTM Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

# Save the plot to the current directory
plt.savefig('btc-loss.png')

plt.show()

from sklearn.metrics import mean_squared_error, mean_absolute_error

# Make predictions
lstm_predictions = lstm_model.predict(X_test_btc)
gru_predictions = gru_model.predict(X_test_btc)
bilstm_predictions = bilstm_model.predict(X_test_btc)

# Calculate MSE and MAE for LSTM
lstm_mse = mean_squared_error(Y_test_btc, lstm_predictions)
lstm_mae = mean_absolute_error(Y_test_btc, lstm_predictions)

# Calculate MSE and MAE for GRU
gru_mse = mean_squared_error(Y_test_btc, gru_predictions)
gru_mae = mean_absolute_error(Y_test_btc, gru_predictions)

# Calculate MSE and MAE for BiLSTM
bilstm_mse = mean_squared_error(Y_test_btc, bilstm_predictions)
bilstm_mae = mean_absolute_error(Y_test_btc, bilstm_predictions)
```

```python
# Print the scores
print(f'LSTM MSE: {lstm_mse}, MAE: {lstm_mae}')
print(f'GRU MSE: {gru_mse}, MAE: {gru_mae}')
print(f'BiLSTM MSE: {bilstm_mse}, MAE: {bilstm_mae}')

# Generate date range for x-axis
date_range = pd.date_range(start=start_date, end=end_date,
periods=len(Y_test_btc))

# Plot actual vs. predicted prices for the LSTM model
plt.figure(figsize=(10, 6))
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(Y_test_btc),
scaled_data_btc.shape[1]-1)), Y_test_btc.reshape(-1, 1)), axis=1))[:,
-1], label='Actual Prices', color='b')
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(lstm_prediction
s), scaled_data_btc.shape[1]-1)), lstm_predictions), axis=1))[:, -1],
label='Predicted Prices', color='red')
plt.title('For BTC Comparison of Actual and LSTM Predicted Prices')
plt.xlabel('Time')
plt.ylabel('Price (USD)')
plt.gca().xaxis.set_major_locator(mdates.YearLocator())  # Set major
ticks to each year
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y'))  #
Format date to only show years
plt.legend()
plt.grid(True)

# Save the plot to the current directory
plt.savefig('btc-lstm.png')
plt.show()

# Plot actual vs. predicted prices for the GRU model
plt.figure(figsize=(10, 6))
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(Y_test_btc),
scaled_data_btc.shape[1]-1)), Y_test_btc.reshape(-1, 1)), axis=1))[:,
-1], label='Actual Prices', color='b')
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(gru_predictions
), scaled_data_btc.shape[1]-1)), gru_predictions), axis=1))[:, -1],
label='Predicted Prices', color='red')
plt.title('For BTC Comparison of Actual and GRU Predicted Prices')
plt.xlabel('Time')
plt.ylabel('Price (USD)')
plt.gca().xaxis.set_major_locator(mdates.YearLocator())  # Set major
ticks to each year
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y'))  #
Format date to only show years
```

```python
plt.legend()
plt.grid(True)

# Save the plot to the current directory
plt.savefig('btc-gru.png')
plt.show()

# Plot actual vs. predicted prices for the Bi-LSTM model
plt.figure(figsize=(10, 6))
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(Y_test_btc),
scaled_data_btc.shape[1]-1)), Y_test_btc.reshape(-1, 1)), axis=1))[:,
-1], label='Actual Prices', color='b')
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(bilstm_predicti
ons), scaled_data_btc.shape[1]-1)), bilstm_predictions), axis=1))[:, -
1], label='Predicted Prices', color='red')
plt.title('For BTC Comparison of Actual and Bi-LSTM Predicted Prices')
plt.xlabel('Time')
plt.ylabel('Price (USD)')
plt.gca().xaxis.set_major_locator(mdates.YearLocator())  # Set major
ticks to each year
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y'))  #
Format date to only show years
plt.legend()
plt.grid(True)

# Save the plot to the current directory
plt.savefig('btc-bilstm.png')
plt.show()

def calculate_rmse(actuals, predictions):
    """
    Calculate Root Mean Squared Error
    """
    mse = np.mean((actuals - predictions) ** 2)
    rmse = np.sqrt(mse)
    return rmse

def calculate_mape(actuals, predictions):
    """
    Calculate Mean Absolute Percentage Error
    """
    mape = np.mean(np.abs((actuals - predictions) / actuals)) * 100
    return mape


# Assuming Y_test, lstm_predictions, gru_predictions, and
bilstm_predictions are already defined
```

```python
# Calculate RMSE for each model
lstm_rmse = calculate_rmse(Y_test_btc, lstm_predictions.flatten())
gru_rmse = calculate_rmse(Y_test_btc, gru_predictions.flatten())
bilstm_rmse = calculate_rmse(Y_test_btc, bilstm_predictions.flatten())

# Calculate MAPE for each model
lstm_mape = calculate_mape(Y_test_btc, lstm_predictions.flatten())
gru_mape = calculate_mape(Y_test_btc, gru_predictions.flatten())
bilstm_mape = calculate_mape(Y_test_btc, bilstm_predictions.flatten())

# Print RMSE and MAPE for each model
print(f'LSTM RMSE: {lstm_rmse:.3f}, MAPE: {lstm_mape:.2f}%')
print(f'GRU RMSE: {gru_rmse:.3f}, MAPE: {gru_mape:.2f}%')
print(f'BiLSTM RMSE: {bilstm_rmse:.3f}, MAPE: {bilstm_mape:.2f}%')




# Assuming we're working with Ethereum data for this example
eth_data = crypto_data['ETH-USD']

# 1. Handle missing data
# For simplicity, we'll fill missing values with the last available
value
eth_data.fillna(method='ffill', inplace=True)

# 2. Normalize the features
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data_eth = scaler.fit_transform(eth_data[['Open', 'High',
'Low', 'Close']])  #CHANGE, SAME As above

# 3. Convert the data into a time series format
def create_dataset(dataset, look_back=1):
    X, Y = [], []
    for i in range(len(dataset) - look_back - 1):
        a = dataset[i:(i + look_back), :]
        X.append(a)
        Y.append(dataset[i + look_back, -1])  # Assuming close price
is the target variable
    return np.array(X), np.array(Y)

# Define look_back period
look_back = 60  # for example, using past 60 days to predict the next
day

# Split into features (X) and target (Y)
X_eth, Y_eth = create_dataset(scaled_data_eth, look_back)
```

```python
# Reshape features into the format expected by LSTM (samples, time
steps, features)
X_eth = np.reshape(X_eth, (X_eth.shape[0], look_back, X_eth.shape[2]))

# Split into train and test sets
split_percent = 0.80
split = int(split_percent * len(X_eth))

X_train_eth = X_eth[:split]
Y_train_eth = Y_eth[:split]
X_test_eth = X_eth[split:]
Y_test_eth = Y_eth[split:]

print("Data preprocessed and ready for model training.")

# Import TensorFlow for model training
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, GRU, Bidirectional

# Define the LSTM model
lstm_model = Sequential()
lstm_model.add(LSTM(units=100, return_sequences=True,
input_shape=(X_train_eth.shape[1], X_train_eth.shape[2])))
lstm_model.add(LSTM(units=100))
lstm_model.add(Dense(1))

lstm_model.compile(optimizer='adam', loss='mean_squared_error')

# Define the GRU model
gru_model = Sequential()
gru_model.add(GRU(units=100, return_sequences=True,
input_shape=(X_train_eth.shape[1], X_train_eth.shape[2])))
gru_model.add(GRU(units=100))
gru_model.add(Dense(1))

gru_model.compile(optimizer='adam', loss='mean_squared_error')

# Define the BiLSTM model
bilstm_model = Sequential()
bilstm_model.add(Bidirectional(LSTM(units=100, return_sequences=True),
input_shape=(X_train_eth.shape[1], X_train_eth.shape[2])))
bilstm_model.add(Bidirectional(LSTM(units=100)))
bilstm_model.add(Dense(1))

bilstm_model.compile(optimizer='adam', loss='mean_squared_error')

# Train the models
lstm_history = lstm_model.fit(X_train_eth, Y_train_eth, epochs=100,
batch_size=32, validation_data=(X_test_eth, Y_test_eth), verbose=1)
```

```python
gru_history = gru_model.fit(X_train_eth, Y_train_eth, epochs=100,
batch_size=32, validation_data=(X_test_eth, Y_test_eth), verbose=1)
bilstm_history = bilstm_model.fit(X_train_eth, Y_train_eth,
epochs=100, batch_size=32, validation_data=(X_test_eth, Y_test_eth),
verbose=1)

# Evaluate the models
lstm_scores = lstm_model.evaluate(X_test_eth, Y_test_eth, verbose=0)
gru_scores = gru_model.evaluate(X_test_eth, Y_test_eth, verbose=0)
bilstm_scores = bilstm_model.evaluate(X_test_eth, Y_test_eth,
verbose=0)

print('LSTM Test loss:', lstm_scores)
print('GRU Test loss:', gru_scores)
print('BiLSTM Test loss:', bilstm_scores)

# Plot training and validation loss
plt.figure(figsize=(18, 5))
plt.subplot(1, 3, 1)
plt.plot(lstm_history.history['loss'], label='LSTM training Loss')
plt.plot(lstm_history.history['val_loss'], label='LSTM Validation
Loss')
plt.title('LSTM Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()


plt.subplot(1, 3, 2)
plt.plot(gru_history.history['loss'], label='GRU training Loss')
plt.plot(gru_history.history['val_loss'], label='GRU Validation Loss')
plt.title('GRU Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()


plt.subplot(1, 3, 3)
plt.plot(bilstm_history.history['loss'], label='Bi-LSTM training
Loss')
plt.plot(bilstm_history.history['val_loss'], label='Bi-LSTM Validation
Loss')
plt.title('Bi-LSTM Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

# Save the plot to the current directory
```

```python
plt.savefig('eth-loss.png')

plt.show()

from sklearn.metrics import mean_squared_error, mean_absolute_error

# Make predictions
lstm_predictions = lstm_model.predict(X_test_eth)
gru_predictions = gru_model.predict(X_test_eth)
bilstm_predictions = bilstm_model.predict(X_test_eth)

# Calculate MSE and MAE for LSTM
lstm_mse = mean_squared_error(Y_test_eth, lstm_predictions)
lstm_mae = mean_absolute_error(Y_test_eth, lstm_predictions)

# Calculate MSE and MAE for GRU
gru_mse = mean_squared_error(Y_test_eth, gru_predictions)
gru_mae = mean_absolute_error(Y_test_eth, gru_predictions)

# Calculate MSE and MAE for BiLSTM
bilstm_mse = mean_squared_error(Y_test_eth, bilstm_predictions)
bilstm_mae = mean_absolute_error(Y_test_eth, bilstm_predictions)

# Print the scores
print(f'LSTM MSE: {lstm_mse}, MAE: {lstm_mae}')
print(f'GRU MSE: {gru_mse}, MAE: {gru_mae}')
print(f'BiLSTM MSE: {bilstm_mse}, MAE: {bilstm_mae}')

# Generate date range for x-axis
date_range = pd.date_range(start=start_date, end=end_date,
periods=len(Y_test_eth))

# Plot actual vs. predicted prices for the LSTM model
plt.figure(figsize=(10, 6))
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(Y_test_eth),
scaled_data_eth.shape[1]-1)), Y_test_eth.reshape(-1, 1)), axis=1))[:,
-1], label='Actual Prices', color='m')
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(lstm_prediction
s), scaled_data_eth.shape[1]-1)), lstm_predictions), axis=1))[:, -1],
label='Predicted Prices', color='lime')
plt.title('For ETH Comparison of Actual and LSTM Predicted Prices')
plt.xlabel('Time')
plt.ylabel('Price (USD)')
plt.gca().xaxis.set_major_locator(mdates.YearLocator())  # Set major
ticks to each year
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y'))  #
Format date to only show years
plt.legend()
plt.grid(True)
```

```python
# Save the plot to the current directory
plt.savefig('eth-lstm.png')
plt.show()

# Plot actual vs. predicted prices for the GRU model
plt.figure(figsize=(10, 6))
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(Y_test_eth),
scaled_data_eth.shape[1]-1)), Y_test_eth.reshape(-1, 1)), axis=1))[:,
-1], label='Actual Prices', color='m')
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(gru_predictions
), scaled_data_eth.shape[1]-1)), gru_predictions), axis=1))[:, -1],
label='Predicted Prices', color='lime')
plt.title('For ETH Comparison of Actual and GRU Predicted Prices')
plt.xlabel('Time')
plt.ylabel('Price (USD)')
plt.gca().xaxis.set_major_locator(mdates.YearLocator())  # Set major
ticks to each year
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y'))  #
Format date to only show years
plt.legend()
plt.grid(True)
# Save the plot to the current directory
plt.savefig('eth-gru.png')
plt.show()

# Plot actual vs. predicted prices for the Bi-LSTM model
plt.figure(figsize=(10, 6))
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(Y_test_eth),
scaled_data_eth.shape[1]-1)), Y_test_eth.reshape(-1, 1)), axis=1))[:,
-1], label='Actual Prices', color='m')
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(bilstm_predicti
ons), scaled_data_eth.shape[1]-1)), bilstm_predictions), axis=1))[:, -
1], label='Predicted Prices', color='lime')
plt.title('For ETH Comparison of Actual and Bi-LSTM Predicted Prices')
plt.xlabel('Time')
plt.ylabel('Price (USD)')
plt.gca().xaxis.set_major_locator(mdates.YearLocator())  # Set major
ticks to each year
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y'))  #
Format date to only show years
plt.legend()
plt.grid(True)
# Save the plot to the current directory
plt.savefig('eth-bilstm.png')
plt.show()
```

```python
def calculate_rmse(actuals, predictions):
    """
    Calculate Root Mean Squared Error
    """
    mse = np.mean((actuals - predictions) ** 2)
    rmse = np.sqrt(mse)
    return rmse

def calculate_mape(actuals, predictions):
    """
    Calculate Mean Absolute Percentage Error
    """
    mape = np.mean(np.abs((actuals - predictions) / actuals)) * 100
    return mape


# Assuming Y_test, lstm_predictions, gru_predictions, and
bilstm_predictions are already defined

# Calculate RMSE for each model
lstm_rmse = calculate_rmse(Y_test_eth, lstm_predictions.flatten())
gru_rmse = calculate_rmse(Y_test_eth, gru_predictions.flatten())
bilstm_rmse = calculate_rmse(Y_test_eth, bilstm_predictions.flatten())

# Calculate MAPE for each model
lstm_mape = calculate_mape(Y_test_eth, lstm_predictions.flatten())
gru_mape = calculate_mape(Y_test_eth, gru_predictions.flatten())
bilstm_mape = calculate_mape(Y_test_eth, bilstm_predictions.flatten())

# Print RMSE and MAPE for each model
print(f'LSTM RMSE: {lstm_rmse:.3f}, MAPE: {lstm_mape:.2f}%')
print(f'GRU RMSE: {gru_rmse:.3f}, MAPE: {gru_mape:.2f}%')
print(f'BiLSTM RMSE: {bilstm_rmse:.3f}, MAPE: {bilstm_mape:.2f}%')




# Assuming we're working with Litecoin data for this example
ltc_data = crypto_data['LTC-USD']

# 1. Handle missing data
# For simplicity, we'll fill missing values with the last available
value
ltc_data.fillna(method='ffill', inplace=True)

# 2. Normalize the features
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data_ltc = scaler.fit_transform(ltc_data[['Open', 'High',
'Low', 'Close']]) #CHANGE SAME as above
```

```python
# 3. Convert the data into a time series format
def create_dataset(dataset, look_back=1):
    X, Y = [], []
    for i in range(len(dataset) - look_back - 1):
        a = dataset[i:(i + look_back), :]
        X.append(a)
        Y.append(dataset[i + look_back, -1])  # Assuming close price
is the target variable
    return np.array(X), np.array(Y)

# Define look_back period
look_back = 60  # for example, using past 60 days to predict the next
day

# Split into features (X) and target (Y)
X_ltc, Y_ltc = create_dataset(scaled_data_ltc, look_back)

# Reshape features into the format expected by LSTM (samples, time
steps, features)
X_ltc = np.reshape(X_ltc, (X_ltc.shape[0], look_back, X_ltc.shape[2]))

# Split into train and test sets
split_percent = 0.80
split = int(split_percent * len(X_ltc))

X_train_ltc = X_ltc[:split]
Y_train_ltc = Y_ltc[:split]
X_test_ltc = X_ltc[split:]
Y_test_ltc = Y_ltc[split:]

print("Data preprocessed and ready for model training.")

# Import TensorFlow for model training
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, GRU, Bidirectional

# Define the LSTM model
lstm_model = Sequential()
lstm_model.add(LSTM(units=100, return_sequences=True,
input_shape=(X_train_ltc.shape[1], X_train_ltc.shape[2])))
lstm_model.add(LSTM(units=100))
lstm_model.add(Dense(1))

lstm_model.compile(optimizer='adam', loss='mean_squared_error')

# Define the GRU model
gru_model = Sequential()
gru_model.add(GRU(units=100, return_sequences=True,
input_shape=(X_train_ltc.shape[1], X_train_ltc.shape[2])))
gru_model.add(GRU(units=100))
```

```python
gru_model.add(Dense(1))

gru_model.compile(optimizer='adam', loss='mean_squared_error')

# Define the BiLSTM model
bilstm_model = Sequential()
bilstm_model.add(Bidirectional(LSTM(units=100, return_sequences=True),
input_shape=(X_train_ltc.shape[1], X_train_ltc.shape[2])))
bilstm_model.add(Bidirectional(LSTM(units=100)))
bilstm_model.add(Dense(1))

bilstm_model.compile(optimizer='adam', loss='mean_squared_error')

# Train the models
lstm_history = lstm_model.fit(X_train_ltc, Y_train_ltc, epochs=100,
batch_size=32, validation_data=(X_test_ltc, Y_test_ltc), verbose=1)
gru_history = gru_model.fit(X_train_ltc, Y_train_ltc, epochs=100,
batch_size=32, validation_data=(X_test_ltc, Y_test_ltc), verbose=1)
bilstm_history = bilstm_model.fit(X_train_ltc, Y_train_ltc,
epochs=100, batch_size=32, validation_data=(X_test_ltc, Y_test_ltc),
verbose=1)

# Evaluate the models
lstm_scores = lstm_model.evaluate(X_test_ltc, Y_test_ltc, verbose=0)
gru_scores = gru_model.evaluate(X_test_ltc, Y_test_ltc, verbose=0)
bilstm_scores = bilstm_model.evaluate(X_test_ltc, Y_test_ltc,
verbose=0)

print('LSTM Test loss:', lstm_scores)
print('GRU Test loss:', gru_scores)
print('BiLSTM Test loss:', bilstm_scores)

# Plot training and validation loss
plt.figure(figsize=(18, 5))
plt.subplot(1, 3, 1)
plt.plot(lstm_history.history['loss'], label='LSTM training Loss')
plt.plot(lstm_history.history['val_loss'], label='LSTM Validation
Loss')
plt.title('LSTM Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()


plt.subplot(1, 3, 2)
plt.plot(gru_history.history['loss'], label='GRU training Loss')
plt.plot(gru_history.history['val_loss'], label='GRU Validation Loss')
plt.title('GRU Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
```

```python
plt.legend()


plt.subplot(1, 3, 3)
plt.plot(bilstm_history.history['loss'], label='Bi-LSTM training
Loss')
plt.plot(bilstm_history.history['val_loss'], label='Bi-LSTM Validation
Loss')
plt.title('Bi-LSTM Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

# Save the plot to the current directory
plt.savefig('ltc-loss.png')

plt.show()

from sklearn.metrics import mean_squared_error, mean_absolute_error

# Make predictions
lstm_predictions = lstm_model.predict(X_test_ltc)
gru_predictions = gru_model.predict(X_test_ltc)
bilstm_predictions = bilstm_model.predict(X_test_ltc)

# Calculate MSE and MAE for LSTM
lstm_mse = mean_squared_error(Y_test_ltc, lstm_predictions)
lstm_mae = mean_absolute_error(Y_test_ltc, lstm_predictions)

# Calculate MSE and MAE for GRU
gru_mse = mean_squared_error(Y_test_ltc, gru_predictions)
gru_mae = mean_absolute_error(Y_test_ltc, gru_predictions)

# Calculate MSE and MAE for BiLSTM
bilstm_mse = mean_squared_error(Y_test_ltc, bilstm_predictions)
bilstm_mae = mean_absolute_error(Y_test_ltc, bilstm_predictions)

# Print the scores
print(f'LSTM MSE: {lstm_mse}, MAE: {lstm_mae}')
print(f'GRU MSE: {gru_mse}, MAE: {gru_mae}')
print(f'BiLSTM MSE: {bilstm_mse}, MAE: {bilstm_mae}')

# Generate date range for x-axis
date_range = pd.date_range(start=start_date, end=end_date,
periods=len(Y_test_ltc))

# Plot actual vs. predicted prices for the LSTM model
plt.figure(figsize=(10, 6))
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(Y_test_ltc),
```

```
scaled_data_ltc.shape[1]-1)), Y_test_ltc.reshape(-1, 1)), axis=1))[:,
-1], label='Actual Prices', color='deeppink')
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(lstm_prediction
s), scaled_data_ltc.shape[1]-1)), lstm_predictions), axis=1))[:, -1],
label='Predicted Prices', color='c')
plt.title('For LTC Comparison of Actual and LSTM Predicted Prices')
plt.xlabel('Time')
plt.ylabel('Price (USD)')
plt.gca().xaxis.set_major_locator(mdates.YearLocator())  # Set major
ticks to each year
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y'))  #
Format date to only show years
plt.legend()
plt.grid(True)
# Save the plot to the current directory
plt.savefig('ltc-lstm.png')
plt.show()

# Plot actual vs. predicted prices for the GRU model
plt.figure(figsize=(10, 6))
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(Y_test_ltc),
scaled_data_ltc.shape[1]-1)), Y_test_ltc.reshape(-1, 1)), axis=1))[:,
-1], label='Actual Prices', color='deeppink')
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(gru_predictions
), scaled_data_ltc.shape[1]-1)), gru_predictions), axis=1))[:, -1],
label='Predicted Prices', color='c')
plt.title('For LTC Comparison of Actual and GRU Predicted Prices')
plt.xlabel('Time')
plt.ylabel('Price (USD)')
plt.gca().xaxis.set_major_locator(mdates.YearLocator())  # Set major
ticks to each year
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y'))  #
Format date to only show years
plt.legend()
plt.grid(True)
# Save the plot to the current directory
plt.savefig('ltc-gru.png')
plt.show()

# Plot actual vs. predicted prices for the Bi-LSTM model
plt.figure(figsize=(10, 6))
plt.plot(date_range,
scaler.inverse_transform(np.concatenate((np.zeros((len(Y_test_ltc),
scaled_data_ltc.shape[1]-1)), Y_test_ltc.reshape(-1, 1)), axis=1))[:,
-1], label='Actual Prices', color='deeppink')
plt.plot(date_range,
```

```python
scaler.inverse_transform(np.concatenate((np.zeros((len(bilstm_predicti
ons), scaled_data_ltc.shape[1]-1)), bilstm_predictions), axis=1))[:, -
1], label='Predicted Prices', color='c')
plt.title('For LTC Comparison of Actual and Bi-LSTM Predicted Prices')
plt.xlabel('Time')
plt.ylabel('Price (USD)')
plt.gca().xaxis.set_major_locator(mdates.YearLocator())  # Set major
ticks to each year
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%Y'))  #
Format date to only show years
plt.legend()
plt.grid(True)
# Save the plot to the current directory
plt.savefig('ltc-bilstm.png')
plt.show()

def calculate_rmse(actuals, predictions):
    """
    Calculate Root Mean Squared Error
    """
    mse = np.mean((actuals - predictions) ** 2)
    rmse = np.sqrt(mse)
    return rmse

def calculate_mape(actuals, predictions):
    """
    Calculate Mean Absolute Percentage Error
    """
    mape = np.mean(np.abs((actuals - predictions) / actuals)) * 100
    return mape


# Assuming Y_test, lstm_predictions, gru_predictions, and
bilstm_predictions are already defined

# Calculate RMSE for each model
lstm_rmse = calculate_rmse(Y_test_ltc, lstm_predictions.flatten())
gru_rmse = calculate_rmse(Y_test_ltc, gru_predictions.flatten())
bilstm_rmse = calculate_rmse(Y_test_ltc, bilstm_predictions.flatten())

# Calculate MAPE for each model
lstm_mape = calculate_mape(Y_test_ltc, lstm_predictions.flatten())
gru_mape = calculate_mape(Y_test_ltc, gru_predictions.flatten())
bilstm_mape = calculate_mape(Y_test_ltc, bilstm_predictions.flatten())

# Print RMSE and MAPE for each model
print(f'LSTM RMSE: {lstm_rmse:.3f}, MAPE: {lstm_mape:.2f}%')
print(f'GRU RMSE: {gru_rmse:.3f}, MAPE: {gru_mape:.2f}%')
print(f'BiLSTM RMSE: {bilstm_rmse:.3f}, MAPE: {bilstm_mape:.2f}%')
```

```
Requirement already satisfied: yfinance in
/usr/local/lib/python3.11/dist-packages (0.2.55)
Requirement already satisfied: numpy in
/usr/local/lib/python3.11/dist-packages (2.0.2)
Requirement already satisfied: pandas in
/usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: scikit-learn in
/usr/local/lib/python3.11/dist-packages (1.6.1)
Requirement already satisfied: tensorflow in
/usr/local/lib/python3.11/dist-packages (2.18.0)
Requirement already satisfied: requests>=2.31 in
/usr/local/lib/python3.11/dist-packages (from yfinance) (2.32.3)
Requirement already satisfied: multitasking>=0.0.7 in
/usr/local/lib/python3.11/dist-packages (from yfinance) (0.0.11)
Requirement already satisfied: platformdirs>=2.0.0 in
/usr/local/lib/python3.11/dist-packages (from yfinance) (4.3.7)
Requirement already satisfied: pytz>=2022.5 in
/usr/local/lib/python3.11/dist-packages (from yfinance) (2025.2)
Requirement already satisfied: frozendict>=2.3.4 in
/usr/local/lib/python3.11/dist-packages (from yfinance) (2.4.6)
Requirement already satisfied: peewee>=3.16.2 in
/usr/local/lib/python3.11/dist-packages (from yfinance) (3.17.9)
Requirement already satisfied: beautifulsoup4>=4.11.1 in
/usr/local/lib/python3.11/dist-packages (from yfinance) (4.13.3)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.11/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in
/usr/local/lib/python3.11/dist-packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.3)
Requirement already satisfied: scipy>=1.6.0 in
/usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.14.1)
Requirement already satisfied: joblib>=1.2.0 in
/usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.0)
```

```
Requirement already satisfied: absl-py>=1.0.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (25.2.10)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1
in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!
=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.3 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (5.29.4)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (75.2.0)
Requirement already satisfied: six>=1.12.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (2.5.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (4.13.0)
Requirement already satisfied: wrapt>=1.11.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.2)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (1.71.0)
Requirement already satisfied: tensorboard<2.19,>=2.18 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (2.18.0)
Requirement already satisfied: keras>=3.5.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (3.8.0)
Requirement already satisfied: h5py>=3.11.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (3.13.0)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (0.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.11/dist-packages (from tensorflow) (0.37.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.11/dist-packages (from astunparse>=1.6.0-
>tensorflow) (0.45.1)
Requirement already satisfied: soupsieve>1.2 in
/usr/local/lib/python3.11/dist-packages (from beautifulsoup4>=4.11.1-
>yfinance) (2.6)
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-
packages (from keras>=3.5.0->tensorflow) (13.9.4)
Requirement already satisfied: namex in
/usr/local/lib/python3.11/dist-packages (from keras>=3.5.0-
>tensorflow) (0.0.8)
```

```
Requirement already satisfied: optree in
/usr/local/lib/python3.11/dist-packages (from keras>=3.5.0-
>tensorflow) (0.14.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.31-
>yfinance) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.31-
>yfinance) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.31-
>yfinance) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.31-
>yfinance) (2025.1.31)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18-
>tensorflow) (3.7)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0
in /usr/local/lib/python3.11/dist-packages (from
tensorboard<2.19,>=2.18->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18-
>tensorflow) (3.1.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1-
>tensorboard<2.19,>=2.18->tensorflow) (3.0.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0-
>tensorflow) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0-
>tensorflow) (2.18.0)
Requirement already satisfied: mdurl~=0.1 in
/usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0-
>rich->keras>=3.5.0->tensorflow) (0.1.2)
YF.download() has changed argument auto_adjust default to True

 [*********************100%***********************]  1 of 1 completed

Showing data for BTC-USD:
```

```
{"summary":"{\n  \"name\": \"print(f'BiLSTM RMSE: {bilstm_rmse:\",\n
\"rows\": 5,\n  \"fields\": [\n    {\n      \"column\": [\n
\"Date\",\n        \"\\\"\n      ],\n      \"properties\": {\n
\"dtype\": \"date\",\n        \"min\": \"2019-01-01 00:00:00\",\n
\"max\": \"2019-01-05 00:00:00\",\n        \"num_unique_values\": 5,\n
\"samples\": [\n          \"2019-01-02 00:00:00\",\n          \"2019-
```

01-05 00:00:00\",\n   \"2019-01-03 00:00:00\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n   \"Close\",\n   \"BTC-USD\"\n   ],\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 44.30794080898695,\n   \"min\": 3836.7412109375,\n   \"max\": 3943.409423828125,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   3943.409423828125,\n   3845.194580078125,\n   3836.7412109375\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n   \"High\",\n   \"BTC-USD\"\n   ],\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 42.31930885702279,\n   \"min\": 3850.913818359375,\n   \"max\": 3947.981201171875,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   3947.981201171875,\n   3904.903076171875,\n   3935.68505859375\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n   \"Low\",\n   \"BTC-USD\"\n   ],\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 52.57848045036253,\n   \"min\": 3707.231201171875,\n   \"max\": 3836.900146484375,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   3817.409423828125,\n   3836.900146484375,\n   3826.222900390625\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n   \"Open\",\n   \"BTC-USD\"\n   ],\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 65.68817955871506,\n   \"min\": 3746.71337890625,\n   \"max\": 3931.048583984375,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   3849.21630859375,\n   3851.973876953125,\n   3931.048583984375\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n   \"Volume\",\n   \"BTC-USD\"\n   ],\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 391122538,\n   \"min\": 4324200990,\n   \"max\": 5244856836,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   5244856836,\n   5137609824,\n   4530215219\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

[*********************100%***********************] 1 of 1 completed

Showing data for ETH-USD:

{"summary":"{\n \"name\": \"print(f'BiLSTM RMSE: {bilstm_rmse:\",\n \"rows\": 5,\n \"fields\": [\n   {\n   \"column\": [\n   \"Date\",\n   \"\"\n   ],\n   \"properties\": {\n   \"dtype\": \"date\",\n   \"min\": \"2019-01-01 00:00:00\",\n   \"max\": \"2019-01-05 00:00:00\",\n   \"num_unique_values\": 5,\n

\"samples\": [\n   \"2019-01-02 00:00:00\",\n   \"2019-01-05 00:00:00\",\n   \"2019-01-03 00:00:00\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n   \"Close\",\n   \"ETH-USD\"\n   ],\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 6.281784111781269,\n   \"min\": 140.8194122314453,\n   \"max\": 155.6385955810547,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   155.0476837158203,\n   155.6385955810547,\n   149.135009765625\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n   \"High\",\n   \"ETH-USD\"\n   ],\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 7.50040325733317,\n   \"min\": 141.3975067138672,\n   \"max\": 160.82489013671875,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   156.92913818359375,\n   160.82489013671875,\n   155.86305236816406\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n   \"Low\",\n   \"ETH-USD\"\n   ],\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 8.229112176351794,\n   \"min\": 132.6507110595703,\n   \"max\": 154.33741760253906,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   140.6509552001953,\n   154.33741760253906,\n   147.1983642578125\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n   \"Open\",\n   \"ETH-USD\"\n   ],\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 9.201250087544645,\n   \"min\": 133.41815185546875,\n   \"max\": 155.196044921875,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   141.51951599121094,\n   154.33741760253906,\n   155.196044921875\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n   \"Volume\",\n   \"ETH-USD\"\n   ],\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 468397462,\n   \"min\": 2258709868,\n   \"max\": 3338211928,\n   \"num_unique_values\": 5,\n   \"samples\": [\n   3328240369,\n   3338211928,\n   2676164880\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n   ]\n}","type":"dataframe"}

```
[********************100%**********************]  1 of 1 completed
```

Showing data for LTC-USD:

{"summary":"{\n \"name\": \"print(f'BiLSTM RMSE: {bilstm_rmse:\",\n \"rows\": 5,\n \"fields\": [\n   {\n   \"column\": [\n   \"Date\",\n   \"\"\n   ],\n   \"properties\": {\n   \"dtype\": \"date\",\n   \"min\": \"2019-01-01 00:00:00\",\n

\"max\": \"2019-01-05 00:00:00\",\n   \"num_unique_values\": 5,\n \"samples\": [\n   \"2019-01-02 00:00:00\",\n   \"2019-01-05 00:00:00\",\n   \"2019-01-03 00:00:00\"\n   ],\n \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n   \"Close\",\n \"LTC-USD\"\n   ],\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1.252282215345374,\n   \"min\": 31.979930877685547,\n   \"max\": 34.936866760253906,\n \"num_unique_values\": 5,\n   \"samples\": [\n 33.43368148803711,\n   34.936866760253906,\n 32.02669906616211\n   ],\n   \"semantic_type\": \"\",\n \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n \"High\",\n   \"LTC-USD\"\n   ],\n   \"properties\": {\n \"dtype\": \"number\",\n   \"std\": 1.506868926802265,\n \"min\": 32.09758377075195,\n   \"max\": 36.14344024658203,\n \"num_unique_values\": 5,\n   \"samples\": [\n 33.7933235168457,\n   36.14344024658203,\n 33.42070388793945\n   ],\n   \"semantic_type\": \"\",\n \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n \"Low\",\n   \"LTC-USD\"\n   ],\n   \"properties\": {\n \"dtype\": \"number\",\n   \"std\": 0.761917676081607,\n \"min\": 30.264280319213867,\n   \"max\": 32.34479522705078,\n \"num_unique_values\": 5,\n   \"samples\": [\n 31.765974044799805,\n   32.34479522705078,\n 31.592479705810547\n   ],\n   \"semantic_type\": \"\",\n \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n \"Open\",\n   \"LTC-USD\"\n   ],\n   \"properties\": {\n \"dtype\": \"number\",\n   \"std\": 1.0393762259537955,\n \"min\": 30.458393096923828,\n   \"max\": 33.353572845458984,\n \"num_unique_values\": 5,\n   \"samples\": [\n 32.0212287902832,\n   32.34554672241211,\n 33.353572845458984\n   ],\n   \"semantic_type\": \"\",\n \"description\": \"\"\n   }\n   },\n   {\n   \"column\": [\n \"Volume\",\n   \"LTC-USD\"\n   ],\n   \"properties\": {\n \"dtype\": \"number\",\n   \"std\": 118119031,\n   \"min\": 345068249,\n   \"max\": 640607603,\n \"num_unique_values\": 5,\n   \"samples\": [\n 414331918,\n   640607603,\n   345068249\n   ],\n \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

<ipython-input-1-11932f61a7e0>:58: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.
  btc_data.fillna(method='ffill', inplace=True)

	Close	High	Low	Open \
Date				
2019-01-01	3843.520020	3850.913818	3707.231201	3746.713379
2019-01-02	3943.409424	3947.981201	3817.409424	3849.216309

```
2019-01-03    3836.741211    3935.685059    3826.222900    3931.048584
2019-01-04    3857.717529    3865.934570    3783.853760    3832.040039
2019-01-05    3845.194580    3904.903076    3836.900146    3851.973877
...                    ...            ...            ...            ...
2023-12-27   43442.855469   43683.160156   42167.582031   42518.468750
2023-12-28   42627.855469   43804.781250   42318.550781   43468.199219
2023-12-29   42099.402344   43124.324219   41424.062500   42614.644531
2023-12-30   42156.902344   42584.125000   41556.226562   42091.753906
2023-12-31   42265.187500   42860.937500   41998.253906   42152.097656

                   Volume   Sentiment
Date
2019-01-01    4324200990    0.541667
2019-01-02    5244856836    0.000000
2019-01-03    4530215219    0.000000
2019-01-04    4847965467    0.000000
2019-01-05    5137609824    0.088636
...                  ...         ...
2023-12-27   25260941032    0.227273
2023-12-28   22992093014    0.000000
2023-12-29   26000021055    0.000000
2023-12-30   16013925945    0.160000
2023-12-31   16397498810    0.330303

[1826 rows x 6 columns]
Data preprocessed and ready for model training.

/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/
rnn.py:200: UserWarning: Do not pass an `input_shape`/`input_dim`
argument to a layer. When using Sequential models, prefer using an
`Input(shape)` object as the first layer in the model instead.
  super().__init__(**kwargs)

Epoch 1/100

/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/
bidirectional.py:107: UserWarning: Do not pass an
`input_shape`/`input_dim` argument to a layer. When using Sequential
models, prefer using an `Input(shape)` object as the first layer in
the model instead.
  super().__init__(**kwargs)

45/45 ━━━━━━━━━━━━━━━━━━━━ 5s 51ms/step - loss: 0.0263 - val_loss:
5.6327e-04
Epoch 2/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 2s 13ms/step - loss: 0.0015 - val_loss:
5.6195e-04
Epoch 3/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - loss: 0.0014 - val_loss:
7.7429e-04
```
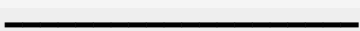
```
Epoch 4/100
45/45 ──────────────── 1s 9ms/step - loss: 0.0013 - val_loss:
6.6081e-04
Epoch 5/100
45/45 ──────────────── 0s 9ms/step - loss: 0.0011 - val_loss:
5.2950e-04
Epoch 6/100
45/45 ──────────────── 0s 10ms/step - loss: 0.0010 - val_loss:
4.2895e-04
Epoch 7/100
45/45 ──────────────── 1s 9ms/step - loss: 9.5142e-04 - val_loss:
3.6824e-04
Epoch 8/100
45/45 ──────────────── 0s 10ms/step - loss: 8.9372e-04 - val_loss:
3.3800e-04
Epoch 9/100
45/45 ──────────────── 1s 9ms/step - loss: 8.4750e-04 - val_loss:
3.2733e-04
Epoch 10/100
45/45 ──────────────── 0s 10ms/step - loss: 8.1141e-04 - val_loss:
3.2731e-04
Epoch 11/100
45/45 ──────────────── 1s 9ms/step - loss: 7.8449e-04 - val_loss:
3.3155e-04
Epoch 12/100
45/45 ──────────────── 0s 10ms/step - loss: 7.6412e-04 - val_loss:
3.3520e-04
Epoch 13/100
45/45 ──────────────── 1s 10ms/step - loss: 7.4677e-04 - val_loss:
3.3569e-04
Epoch 14/100
45/45 ──────────────── 1s 9ms/step - loss: 7.2991e-04 - val_loss:
3.3243e-04
Epoch 15/100
45/45 ──────────────── 1s 10ms/step - loss: 7.1274e-04 - val_loss:
3.2568e-04
Epoch 16/100
45/45 ──────────────── 0s 9ms/step - loss: 6.9544e-04 - val_loss:
3.1604e-04
Epoch 17/100
45/45 ──────────────── 1s 12ms/step - loss: 6.7819e-04 - val_loss:
3.0415e-04
Epoch 18/100
45/45 ──────────────── 1s 15ms/step - loss: 6.6093e-04 - val_loss:
2.9056e-04
Epoch 19/100
45/45 ──────────────── 1s 14ms/step - loss: 6.4351e-04 - val_loss:
2.7575e-04
Epoch 20/100
```

```
45/45 ──────────────── 1s 12ms/step - loss: 6.2590e-04 - val_loss:
2.6023e-04
Epoch 21/100
45/45 ──────────────── 0s 10ms/step - loss: 6.0816e-04 - val_loss:
2.4455e-04
Epoch 22/100
45/45 ──────────────── 1s 12ms/step - loss: 5.9051e-04 - val_loss:
2.2926e-04
Epoch 23/100
45/45 ──────────────── 0s 9ms/step - loss: 5.7322e-04 - val_loss:
2.1488e-04
Epoch 24/100
45/45 ──────────────── 0s 9ms/step - loss: 5.5657e-04 - val_loss:
2.0180e-04
Epoch 25/100
45/45 ──────────────── 1s 9ms/step - loss: 5.4084e-04 - val_loss:
1.9025e-04
Epoch 26/100
45/45 ──────────────── 1s 11ms/step - loss: 5.2620e-04 - val_loss:
1.8031e-04
Epoch 27/100
45/45 ──────────────── 0s 9ms/step - loss: 5.1275e-04 - val_loss:
1.7192e-04
Epoch 28/100
45/45 ──────────────── 1s 10ms/step - loss: 5.0051e-04 - val_loss:
1.6496e-04
Epoch 29/100
45/45 ──────────────── 0s 10ms/step - loss: 4.8944e-04 - val_loss:
1.5922e-04
Epoch 30/100
45/45 ──────────────── 1s 10ms/step - loss: 4.7947e-04 - val_loss:
1.5452e-04
Epoch 31/100
45/45 ──────────────── 1s 10ms/step - loss: 4.7051e-04 - val_loss:
1.5069e-04
Epoch 32/100
45/45 ──────────────── 0s 10ms/step - loss: 4.6245e-04 - val_loss:
1.4758e-04
Epoch 33/100
45/45 ──────────────── 0s 10ms/step - loss: 4.5519e-04 - val_loss:
1.4510e-04
Epoch 34/100
45/45 ──────────────── 1s 10ms/step - loss: 4.4867e-04 - val_loss:
1.4315e-04
Epoch 35/100
45/45 ──────────────── 1s 9ms/step - loss: 4.4280e-04 - val_loss:
1.4170e-04
Epoch 36/100
45/45 ──────────────── 1s 10ms/step - loss: 4.3755e-04 - val_loss:
```

```
1.4067e-04
Epoch 37/100
45/45 ───────────────── 0s 10ms/step - loss: 4.3285e-04 - val_loss:
1.4002e-04
Epoch 38/100
45/45 ───────────────── 0s 10ms/step - loss: 4.2867e-04 - val_loss:
1.3970e-04
Epoch 39/100
45/45 ───────────────── 1s 13ms/step - loss: 4.2497e-04 - val_loss:
1.3965e-04
Epoch 40/100
45/45 ───────────────── 1s 13ms/step - loss: 4.2171e-04 - val_loss:
1.3981e-04
Epoch 41/100
45/45 ───────────────── 1s 15ms/step - loss: 4.1885e-04 - val_loss:
1.4012e-04
Epoch 42/100
45/45 ───────────────── 1s 13ms/step - loss: 4.1636e-04 - val_loss:
1.4052e-04
Epoch 43/100
45/45 ───────────────── 0s 10ms/step - loss: 4.1419e-04 - val_loss:
1.4097e-04
Epoch 44/100
45/45 ───────────────── 0s 10ms/step - loss: 4.1231e-04 - val_loss:
1.4142e-04
Epoch 45/100
45/45 ───────────────── 0s 10ms/step - loss: 4.1067e-04 - val_loss:
1.4185e-04
Epoch 46/100
45/45 ───────────────── 1s 10ms/step - loss: 4.0924e-04 - val_loss:
1.4223e-04
Epoch 47/100
45/45 ───────────────── 0s 10ms/step - loss: 4.0799e-04 - val_loss:
1.4254e-04
Epoch 48/100
45/45 ───────────────── 0s 9ms/step - loss: 4.0690e-04 - val_loss:
1.4279e-04
Epoch 49/100
45/45 ───────────────── 1s 10ms/step - loss: 4.0592e-04 - val_loss:
1.4296e-04
Epoch 50/100
45/45 ───────────────── 0s 9ms/step - loss: 4.0506e-04 - val_loss:
1.4306e-04
Epoch 51/100
45/45 ───────────────── 1s 10ms/step - loss: 4.0428e-04 - val_loss:
1.4309e-04
Epoch 52/100
45/45 ───────────────── 0s 9ms/step - loss: 4.0356e-04 - val_loss:
1.4305e-04
```

```
Epoch 53/100
45/45 ———————————————— 1s 10ms/step - loss: 4.0291e-04 - val_loss:
1.4296e-04
Epoch 54/100
45/45 ———————————————— 0s 10ms/step - loss: 4.0230e-04 - val_loss:
1.4282e-04
Epoch 55/100
45/45 ———————————————— 0s 10ms/step - loss: 4.0172e-04 - val_loss:
1.4263e-04
Epoch 56/100
45/45 ———————————————— 0s 10ms/step - loss: 4.0118e-04 - val_loss:
1.4240e-04
Epoch 57/100
45/45 ———————————————— 1s 12ms/step - loss: 4.0065e-04 - val_loss:
1.4214e-04
Epoch 58/100
45/45 ———————————————— 0s 10ms/step - loss: 4.0014e-04 - val_loss:
1.4184e-04
Epoch 59/100
45/45 ———————————————— 1s 10ms/step - loss: 3.9964e-04 - val_loss:
1.4152e-04
Epoch 60/100
45/45 ———————————————— 0s 10ms/step - loss: 3.9915e-04 - val_loss:
1.4118e-04
Epoch 61/100
45/45 ———————————————— 1s 12ms/step - loss: 3.9866e-04 - val_loss:
1.4082e-04
Epoch 62/100
45/45 ———————————————— 1s 15ms/step - loss: 3.9817e-04 - val_loss:
1.4044e-04
Epoch 63/100
45/45 ———————————————— 1s 15ms/step - loss: 3.9769e-04 - val_loss:
1.4005e-04
Epoch 64/100
45/45 ———————————————— 1s 11ms/step - loss: 3.9720e-04 - val_loss:
1.3965e-04
Epoch 65/100
45/45 ———————————————— 0s 9ms/step - loss: 3.9671e-04 - val_loss:
1.3924e-04
Epoch 66/100
45/45 ———————————————— 0s 10ms/step - loss: 3.9622e-04 - val_loss:
1.3882e-04
Epoch 67/100
45/45 ———————————————— 0s 10ms/step - loss: 3.9574e-04 - val_loss:
1.3840e-04
Epoch 68/100
45/45 ———————————————— 0s 9ms/step - loss: 3.9525e-04 - val_loss:
1.3798e-04
Epoch 69/100
```

```
45/45 ───────────────── 1s 9ms/step - loss: 3.9476e-04 - val_loss:
1.3755e-04
Epoch 70/100
45/45 ───────────────── 1s 11ms/step - loss: 3.9428e-04 - val_loss:
1.3713e-04
Epoch 71/100
45/45 ───────────────── 0s 9ms/step - loss: 3.9380e-04 - val_loss:
1.3670e-04
Epoch 72/100
45/45 ───────────────── 0s 9ms/step - loss: 3.9332e-04 - val_loss:
1.3627e-04
Epoch 73/100
45/45 ───────────────── 1s 9ms/step - loss: 3.9285e-04 - val_loss:
1.3584e-04
Epoch 74/100
45/45 ───────────────── 1s 10ms/step - loss: 3.9239e-04 - val_loss:
1.3541e-04
Epoch 75/100
45/45 ───────────────── 0s 10ms/step - loss: 3.9193e-04 - val_loss:
1.3498e-04
Epoch 76/100
45/45 ───────────────── 1s 10ms/step - loss: 3.9148e-04 - val_loss:
1.3454e-04
Epoch 77/100
45/45 ───────────────── 1s 14ms/step - loss: 3.9104e-04 - val_loss:
1.3410e-04
Epoch 78/100
45/45 ───────────────── 0s 9ms/step - loss: 3.9061e-04 - val_loss:
1.3366e-04
Epoch 79/100
45/45 ───────────────── 0s 9ms/step - loss: 3.9019e-04 - val_loss:
1.3321e-04
Epoch 80/100
45/45 ───────────────── 1s 10ms/step - loss: 3.8978e-04 - val_loss:
1.3275e-04
Epoch 81/100
45/45 ───────────────── 1s 12ms/step - loss: 3.8937e-04 - val_loss:
1.3228e-04
Epoch 82/100
45/45 ───────────────── 1s 15ms/step - loss: 3.8897e-04 - val_loss:
1.3180e-04
Epoch 83/100
45/45 ───────────────── 1s 12ms/step - loss: 3.8858e-04 - val_loss:
1.3130e-04
Epoch 84/100
45/45 ───────────────── 0s 10ms/step - loss: 3.8819e-04 - val_loss:
1.3079e-04
Epoch 85/100
45/45 ───────────────── 1s 10ms/step - loss: 3.8781e-04 - val_loss:
```

```
1.3027e-04
Epoch 86/100
45/45 ──────────────── 0s 9ms/step - loss: 3.8742e-04 - val_loss:
1.2973e-04
Epoch 87/100
45/45 ──────────────── 0s 10ms/step - loss: 3.8703e-04 - val_loss:
1.2917e-04
Epoch 88/100
45/45 ──────────────── 0s 9ms/step - loss: 3.8664e-04 - val_loss:
1.2859e-04
Epoch 89/100
45/45 ──────────────── 0s 10ms/step - loss: 3.8624e-04 - val_loss:
1.2798e-04
Epoch 90/100
45/45 ──────────────── 0s 10ms/step - loss: 3.8585e-04 - val_loss:
1.2733e-04
Epoch 91/100
45/45 ──────────────── 0s 10ms/step - loss: 3.8544e-04 - val_loss:
1.2664e-04
Epoch 92/100
45/45 ──────────────── 1s 10ms/step - loss: 3.8501e-04 - val_loss:
1.2590e-04
Epoch 93/100
45/45 ──────────────── 1s 10ms/step - loss: 3.8455e-04 - val_loss:
1.2510e-04
Epoch 94/100
45/45 ──────────────── 0s 10ms/step - loss: 3.8399e-04 - val_loss:
1.2426e-04
Epoch 95/100
45/45 ──────────────── 0s 10ms/step - loss: 3.8328e-04 - val_loss:
1.2343e-04
Epoch 96/100
45/45 ──────────────── 0s 10ms/step - loss: 3.8237e-04 - val_loss:
1.2272e-04
Epoch 97/100
45/45 ──────────────── 0s 10ms/step - loss: 3.8128e-04 - val_loss:
1.2218e-04
Epoch 98/100
45/45 ──────────────── 1s 10ms/step - loss: 3.8018e-04 - val_loss:
1.2181e-04
Epoch 99/100
45/45 ──────────────── 1s 10ms/step - loss: 3.7923e-04 - val_loss:
1.2154e-04
Epoch 100/100
45/45 ──────────────── 1s 10ms/step - loss: 3.7846e-04 - val_loss:
1.2131e-04
Epoch 1/100
45/45 ──────────────── 3s 23ms/step - loss: 0.0258 - val_loss:
2.8102e-04
```

```
Epoch 2/100
45/45 ───────────────────── 1s 9ms/step - loss: 8.1258e-04 - val_loss:
3.1674e-04
Epoch 3/100
45/45 ───────────────────── 0s 9ms/step - loss: 6.9404e-04 - val_loss:
3.0227e-04
Epoch 4/100
45/45 ───────────────────── 1s 9ms/step - loss: 6.3562e-04 - val_loss:
2.6503e-04
Epoch 5/100
45/45 ───────────────────── 1s 9ms/step - loss: 5.8306e-04 - val_loss:
2.3461e-04
Epoch 6/100
45/45 ───────────────────── 0s 10ms/step - loss: 5.4122e-04 - val_loss:
2.1635e-04
Epoch 7/100
45/45 ───────────────────── 0s 10ms/step - loss: 5.1666e-04 - val_loss:
1.8786e-04
Epoch 8/100
45/45 ───────────────────── 0s 9ms/step - loss: 5.0844e-04 - val_loss:
1.6916e-04
Epoch 9/100
45/45 ───────────────────── 0s 10ms/step - loss: 5.0788e-04 - val_loss:
1.6742e-04
Epoch 10/100
45/45 ───────────────────── 0s 9ms/step - loss: 4.9523e-04 - val_loss:
1.6414e-04
Epoch 11/100
45/45 ───────────────────── 1s 9ms/step - loss: 4.8453e-04 - val_loss:
1.5983e-04
Epoch 12/100
45/45 ───────────────────── 1s 9ms/step - loss: 4.8316e-04 - val_loss:
1.5558e-04
Epoch 13/100
45/45 ───────────────────── 0s 9ms/step - loss: 4.8432e-04 - val_loss:
1.5132e-04
Epoch 14/100
45/45 ───────────────────── 0s 9ms/step - loss: 4.8352e-04 - val_loss:
1.4715e-04
Epoch 15/100
45/45 ───────────────────── 0s 10ms/step - loss: 4.7976e-04 - val_loss:
1.4335e-04
Epoch 16/100
45/45 ───────────────────── 0s 9ms/step - loss: 4.7424e-04 - val_loss:
1.3999e-04
Epoch 17/100
45/45 ───────────────────── 1s 10ms/step - loss: 4.6812e-04 - val_loss:
1.3705e-04
Epoch 18/100
```

```
45/45 ———————————————— 1s 9ms/step - loss: 4.6192e-04 - val_loss:
1.3449e-04
Epoch 19/100
45/45 ———————————————— 0s 9ms/step - loss: 4.5585e-04 - val_loss:
1.3226e-04
Epoch 20/100
45/45 ———————————————— 1s 11ms/step - loss: 4.5004e-04 - val_loss:
1.3034e-04
Epoch 21/100
45/45 ———————————————— 1s 15ms/step - loss: 4.4457e-04 - val_loss:
1.2871e-04
Epoch 22/100
45/45 ———————————————— 1s 12ms/step - loss: 4.3950e-04 - val_loss:
1.2735e-04
Epoch 23/100
45/45 ———————————————— 0s 9ms/step - loss: 4.3488e-04 - val_loss:
1.2624e-04
Epoch 24/100
45/45 ———————————————— 0s 9ms/step - loss: 4.3072e-04 - val_loss:
1.2536e-04
Epoch 25/100
45/45 ———————————————— 1s 9ms/step - loss: 4.2702e-04 - val_loss:
1.2468e-04
Epoch 26/100
45/45 ———————————————— 0s 9ms/step - loss: 4.2377e-04 - val_loss:
1.2419e-04
Epoch 27/100
45/45 ———————————————— 0s 10ms/step - loss: 4.2092e-04 - val_loss:
1.2385e-04
Epoch 28/100
45/45 ———————————————— 1s 9ms/step - loss: 4.1845e-04 - val_loss:
1.2364e-04
Epoch 29/100
45/45 ———————————————— 1s 9ms/step - loss: 4.1631e-04 - val_loss:
1.2353e-04
Epoch 30/100
45/45 ———————————————— 1s 13ms/step - loss: 4.1445e-04 - val_loss:
1.2349e-04
Epoch 31/100
45/45 ———————————————— 1s 13ms/step - loss: 4.1283e-04 - val_loss:
1.2351e-04
Epoch 32/100
45/45 ———————————————— 1s 13ms/step - loss: 4.1141e-04 - val_loss:
1.2356e-04
Epoch 33/100
45/45 ———————————————— 1s 13ms/step - loss: 4.1016e-04 - val_loss:
1.2362e-04
Epoch 34/100
45/45 ———————————————— 0s 10ms/step - loss: 4.0904e-04 - val_loss:
```

```
1.2369e-04
Epoch 35/100
45/45 ──────────────── 1s 11ms/step - loss: 4.0802e-04 - val_loss:
1.2375e-04
Epoch 36/100
45/45 ──────────────── 0s 10ms/step - loss: 4.0708e-04 - val_loss:
1.2378e-04
Epoch 37/100
45/45 ──────────────── 1s 10ms/step - loss: 4.0622e-04 - val_loss:
1.2380e-04
Epoch 38/100
45/45 ──────────────── 0s 9ms/step - loss: 4.0541e-04 - val_loss:
1.2380e-04
Epoch 39/100
45/45 ──────────────── 1s 11ms/step - loss: 4.0464e-04 - val_loss:
1.2377e-04
Epoch 40/100
45/45 ──────────────── 0s 9ms/step - loss: 4.0390e-04 - val_loss:
1.2371e-04
Epoch 41/100
45/45 ──────────────── 1s 13ms/step - loss: 4.0319e-04 - val_loss:
1.2364e-04
Epoch 42/100
45/45 ──────────────── 1s 12ms/step - loss: 4.0250e-04 - val_loss:
1.2354e-04
Epoch 43/100
45/45 ──────────────── 0s 9ms/step - loss: 4.0183e-04 - val_loss:
1.2342e-04
Epoch 44/100
45/45 ──────────────── 0s 10ms/step - loss: 4.0117e-04 - val_loss:
1.2329e-04
Epoch 45/100
45/45 ──────────────── 0s 9ms/step - loss: 4.0052e-04 - val_loss:
1.2313e-04
Epoch 46/100
45/45 ──────────────── 0s 10ms/step - loss: 3.9988e-04 - val_loss:
1.2297e-04
Epoch 47/100
45/45 ──────────────── 1s 9ms/step - loss: 3.9924e-04 - val_loss:
1.2279e-04
Epoch 48/100
45/45 ──────────────── 1s 9ms/step - loss: 3.9861e-04 - val_loss:
1.2261e-04
Epoch 49/100
45/45 ──────────────── 0s 9ms/step - loss: 3.9798e-04 - val_loss:
1.2242e-04
Epoch 50/100
45/45 ──────────────── 0s 10ms/step - loss: 3.9735e-04 - val_loss:
1.2222e-04
```

```
Epoch 51/100
45/45 ──────────────── 0s 9ms/step - loss: 3.9672e-04 - val_loss:
1.2202e-04
Epoch 52/100
45/45 ──────────────── 1s 10ms/step - loss: 3.9609e-04 - val_loss:
1.2181e-04
Epoch 53/100
45/45 ──────────────── 1s 9ms/step - loss: 3.9546e-04 - val_loss:
1.2160e-04
Epoch 54/100
45/45 ──────────────── 0s 9ms/step - loss: 3.9483e-04 - val_loss:
1.2139e-04
Epoch 55/100
45/45 ──────────────── 0s 9ms/step - loss: 3.9420e-04 - val_loss:
1.2118e-04
Epoch 56/100
45/45 ──────────────── 1s 9ms/step - loss: 3.9357e-04 - val_loss:
1.2098e-04
Epoch 57/100
45/45 ──────────────── 0s 9ms/step - loss: 3.9294e-04 - val_loss:
1.2077e-04
Epoch 58/100
45/45 ──────────────── 0s 9ms/step - loss: 3.9231e-04 - val_loss:
1.2057e-04
Epoch 59/100
45/45 ──────────────── 1s 10ms/step - loss: 3.9168e-04 - val_loss:
1.2036e-04
Epoch 60/100
45/45 ──────────────── 1s 10ms/step - loss: 3.9105e-04 - val_loss:
1.2017e-04
Epoch 61/100
45/45 ──────────────── 1s 12ms/step - loss: 3.9043e-04 - val_loss:
1.1997e-04
Epoch 62/100
45/45 ──────────────── 1s 13ms/step - loss: 3.8980e-04 - val_loss:
1.1978e-04
Epoch 63/100
45/45 ──────────────── 1s 15ms/step - loss: 3.8918e-04 - val_loss:
1.1960e-04
Epoch 64/100
45/45 ──────────────── 1s 10ms/step - loss: 3.8856e-04 - val_loss:
1.1942e-04
Epoch 65/100
45/45 ──────────────── 1s 10ms/step - loss: 3.8794e-04 - val_loss:
1.1924e-04
Epoch 66/100
45/45 ──────────────── 0s 10ms/step - loss: 3.8733e-04 - val_loss:
1.1907e-04
Epoch 67/100
45/45 ──────────────── 0s 9ms/step - loss: 3.8672e-04 - val_loss:
```

```
1.1891e-04
Epoch 68/100
45/45 ──────────────── 0s 10ms/step - loss: 3.8612e-04 - val_loss:
1.1875e-04
Epoch 69/100
45/45 ──────────────── 1s 9ms/step - loss: 3.8552e-04 - val_loss:
1.1860e-04
Epoch 70/100
45/45 ──────────────── 1s 9ms/step - loss: 3.8493e-04 - val_loss:
1.1845e-04
Epoch 71/100
45/45 ──────────────── 0s 9ms/step - loss: 3.8435e-04 - val_loss:
1.1831e-04
Epoch 72/100
45/45 ──────────────── 1s 10ms/step - loss: 3.8377e-04 - val_loss:
1.1818e-04
Epoch 73/100
45/45 ──────────────── 1s 11ms/step - loss: 3.8320e-04 - val_loss:
1.1805e-04
Epoch 74/100
45/45 ──────────────── 1s 10ms/step - loss: 3.8263e-04 - val_loss:
1.1792e-04
Epoch 75/100
45/45 ──────────────── 1s 10ms/step - loss: 3.8208e-04 - val_loss:
1.1781e-04
Epoch 76/100
45/45 ──────────────── 0s 10ms/step - loss: 3.8153e-04 - val_loss:
1.1770e-04
Epoch 77/100
45/45 ──────────────── 1s 12ms/step - loss: 3.8100e-04 - val_loss:
1.1759e-04
Epoch 78/100
45/45 ──────────────── 1s 10ms/step - loss: 3.8047e-04 - val_loss:
1.1750e-04
Epoch 79/100
45/45 ──────────────── 1s 10ms/step - loss: 3.7995e-04 - val_loss:
1.1740e-04
Epoch 80/100
45/45 ──────────────── 0s 9ms/step - loss: 3.7945e-04 - val_loss:
1.1732e-04
Epoch 81/100
45/45 ──────────────── 1s 10ms/step - loss: 3.7895e-04 - val_loss:
1.1724e-04
Epoch 82/100
45/45 ──────────────── 1s 15ms/step - loss: 3.7847e-04 - val_loss:
1.1717e-04
Epoch 83/100
45/45 ──────────────── 1s 14ms/step - loss: 3.7799e-04 - val_loss:
1.1710e-04
```

```
Epoch 84/100
45/45 ──────────────── 1s 15ms/step - loss: 3.7753e-04 - val_loss:
1.1704e-04
Epoch 85/100
45/45 ──────────────── 1s 10ms/step - loss: 3.7708e-04 - val_loss:
1.1698e-04
Epoch 86/100
45/45 ──────────────── 1s 11ms/step - loss: 3.7664e-04 - val_loss:
1.1693e-04
Epoch 87/100
45/45 ──────────────── 1s 10ms/step - loss: 3.7622e-04 - val_loss:
1.1689e-04
Epoch 88/100
45/45 ──────────────── 0s 10ms/step - loss: 3.7581e-04 - val_loss:
1.1685e-04
Epoch 89/100
45/45 ──────────────── 0s 9ms/step - loss: 3.7540e-04 - val_loss:
1.1682e-04
Epoch 90/100
45/45 ──────────────── 0s 9ms/step - loss: 3.7502e-04 - val_loss:
1.1680e-04
Epoch 91/100
45/45 ──────────────── 0s 9ms/step - loss: 3.7465e-04 - val_loss:
1.1678e-04
Epoch 92/100
45/45 ──────────────── 1s 10ms/step - loss: 3.7429e-04 - val_loss:
1.1677e-04
Epoch 93/100
45/45 ──────────────── 0s 10ms/step - loss: 3.7394e-04 - val_loss:
1.1676e-04
Epoch 94/100
45/45 ──────────────── 1s 9ms/step - loss: 3.7361e-04 - val_loss:
1.1676e-04
Epoch 95/100
45/45 ──────────────── 1s 10ms/step - loss: 3.7329e-04 - val_loss:
1.1676e-04
Epoch 96/100
45/45 ──────────────── 1s 11ms/step - loss: 3.7299e-04 - val_loss:
1.1677e-04
Epoch 97/100
45/45 ──────────────── 0s 9ms/step - loss: 3.7271e-04 - val_loss:
1.1678e-04
Epoch 98/100
45/45 ──────────────── 1s 10ms/step - loss: 3.7243e-04 - val_loss:
1.1680e-04
Epoch 99/100
45/45 ──────────────── 1s 9ms/step - loss: 3.7218e-04 - val_loss:
1.1683e-04
Epoch 100/100
```

```
45/45 ———————————————— 1s 10ms/step - loss: 3.7194e-04 - val_loss:
1.1685e-04
Epoch 1/100
45/45 ———————————————— 5s 30ms/step - loss: 0.0303 - val_loss:
7.4682e-04
Epoch 2/100
45/45 ———————————————— 1s 18ms/step - loss: 0.0015 - val_loss:
9.6308e-04
Epoch 3/100
45/45 ———————————————— 1s 19ms/step - loss: 0.0012 - val_loss:
8.9513e-04
Epoch 4/100
45/45 ———————————————— 1s 17ms/step - loss: 0.0011 - val_loss:
5.7978e-04
Epoch 5/100
45/45 ———————————————— 1s 16ms/step - loss: 9.5064e-04 - val_loss:
4.1364e-04
Epoch 6/100
45/45 ———————————————— 1s 16ms/step - loss: 8.4495e-04 - val_loss:
3.6862e-04
Epoch 7/100
45/45 ———————————————— 1s 16ms/step - loss: 7.7061e-04 - val_loss:
3.8006e-04
Epoch 8/100
45/45 ———————————————— 1s 16ms/step - loss: 7.2648e-04 - val_loss:
4.2113e-04
Epoch 9/100
45/45 ———————————————— 1s 16ms/step - loss: 6.9442e-04 - val_loss:
4.5018e-04
Epoch 10/100
45/45 ———————————————— 1s 17ms/step - loss: 6.7219e-04 - val_loss:
4.7560e-04
Epoch 11/100
45/45 ———————————————— 1s 21ms/step - loss: 6.6207e-04 - val_loss:
4.9445e-04
Epoch 12/100
45/45 ———————————————— 1s 18ms/step - loss: 6.6743e-04 - val_loss:
4.9232e-04
Epoch 13/100
45/45 ———————————————— 1s 16ms/step - loss: 6.8971e-04 - val_loss:
4.4432e-04
Epoch 14/100
45/45 ———————————————— 1s 17ms/step - loss: 7.1551e-04 - val_loss:
3.5994e-04
Epoch 15/100
45/45 ———————————————— 1s 18ms/step - loss: 7.1028e-04 - val_loss:
3.0536e-04
Epoch 16/100
45/45 ———————————————— 1s 17ms/step - loss: 6.7786e-04 - val_loss:
```

```
2.8993e-04
Epoch 17/100
45/45 ———————————————— 1s 17ms/step - loss: 6.5717e-04 - val_loss:
2.8507e-04
Epoch 18/100
45/45 ———————————————— 1s 18ms/step - loss: 6.4811e-04 - val_loss:
2.7776e-04
Epoch 19/100
45/45 ———————————————— 1s 18ms/step - loss: 6.3942e-04 - val_loss:
2.6853e-04
Epoch 20/100
45/45 ———————————————— 1s 16ms/step - loss: 6.2894e-04 - val_loss:
2.5967e-04
Epoch 21/100
45/45 ———————————————— 1s 18ms/step - loss: 6.1838e-04 - val_loss:
2.5138e-04
Epoch 22/100
45/45 ———————————————— 1s 16ms/step - loss: 6.0839e-04 - val_loss:
2.4333e-04
Epoch 23/100
45/45 ———————————————— 2s 44ms/step - loss: 5.9875e-04 - val_loss:
2.3544e-04
Epoch 24/100
45/45 ———————————————— 1s 16ms/step - loss: 5.8930e-04 - val_loss:
2.2775e-04
Epoch 25/100
45/45 ———————————————— 1s 17ms/step - loss: 5.8004e-04 - val_loss:
2.2029e-04
Epoch 26/100
45/45 ———————————————— 1s 18ms/step - loss: 5.7100e-04 - val_loss:
2.1307e-04
Epoch 27/100
45/45 ———————————————— 1s 17ms/step - loss: 5.6218e-04 - val_loss:
2.0611e-04
Epoch 28/100
45/45 ———————————————— 1s 16ms/step - loss: 5.5357e-04 - val_loss:
1.9944e-04
Epoch 29/100
45/45 ———————————————— 1s 16ms/step - loss: 5.4521e-04 - val_loss:
1.9307e-04
Epoch 30/100
45/45 ———————————————— 1s 18ms/step - loss: 5.3707e-04 - val_loss:
1.8702e-04
Epoch 31/100
45/45 ———————————————— 1s 17ms/step - loss: 5.2919e-04 - val_loss:
1.8130e-04
Epoch 32/100
45/45 ———————————————— 1s 18ms/step - loss: 5.2156e-04 - val_loss:
1.7594e-04
```

```
Epoch 33/100
45/45 ———————————————— 1s 22ms/step - loss: 5.1419e-04 - val_loss:
1.7092e-04
Epoch 34/100
45/45 ———————————————— 1s 20ms/step - loss: 5.0710e-04 - val_loss:
1.6627e-04
Epoch 35/100
45/45 ———————————————— 1s 18ms/step - loss: 5.0027e-04 - val_loss:
1.6198e-04
Epoch 36/100
45/45 ———————————————— 1s 18ms/step - loss: 4.9373e-04 - val_loss:
1.5806e-04
Epoch 37/100
45/45 ———————————————— 1s 17ms/step - loss: 4.8747e-04 - val_loss:
1.5449e-04
Epoch 38/100
45/45 ———————————————— 1s 17ms/step - loss: 4.8151e-04 - val_loss:
1.5127e-04
Epoch 39/100
45/45 ———————————————— 1s 16ms/step - loss: 4.7584e-04 - val_loss:
1.4839e-04
Epoch 40/100
45/45 ———————————————— 1s 16ms/step - loss: 4.7048e-04 - val_loss:
1.4581e-04
Epoch 41/100
45/45 ———————————————— 1s 17ms/step - loss: 4.6542e-04 - val_loss:
1.4353e-04
Epoch 42/100
45/45 ———————————————— 1s 18ms/step - loss: 4.6067e-04 - val_loss:
1.4151e-04
Epoch 43/100
45/45 ———————————————— 1s 18ms/step - loss: 4.5623e-04 - val_loss:
1.3972e-04
Epoch 44/100
45/45 ———————————————— 1s 17ms/step - loss: 4.5208e-04 - val_loss:
1.3813e-04
Epoch 45/100
45/45 ———————————————— 1s 23ms/step - loss: 4.4822e-04 - val_loss:
1.3671e-04
Epoch 46/100
45/45 ———————————————— 1s 18ms/step - loss: 4.4464e-04 - val_loss:
1.3543e-04
Epoch 47/100
45/45 ———————————————— 1s 17ms/step - loss: 4.4132e-04 - val_loss:
1.3426e-04
Epoch 48/100
45/45 ———————————————— 1s 18ms/step - loss: 4.3824e-04 - val_loss:
1.3318e-04
Epoch 49/100
```

```
45/45 ───────────────── 1s 18ms/step - loss: 4.3537e-04 - val_loss:
1.3217e-04
Epoch 50/100
45/45 ───────────────── 1s 18ms/step - loss: 4.3271e-04 - val_loss:
1.3121e-04
Epoch 51/100
45/45 ───────────────── 1s 19ms/step - loss: 4.3023e-04 - val_loss:
1.3030e-04
Epoch 52/100
45/45 ───────────────── 1s 18ms/step - loss: 4.2792e-04 - val_loss:
1.2942e-04
Epoch 53/100
45/45 ───────────────── 1s 17ms/step - loss: 4.2575e-04 - val_loss:
1.2858e-04
Epoch 54/100
45/45 ───────────────── 1s 18ms/step - loss: 4.2372e-04 - val_loss:
1.2777e-04
Epoch 55/100
45/45 ───────────────── 1s 22ms/step - loss: 4.2181e-04 - val_loss:
1.2698e-04
Epoch 56/100
45/45 ───────────────── 1s 23ms/step - loss: 4.2002e-04 - val_loss:
1.2623e-04
Epoch 57/100
45/45 ───────────────── 1s 18ms/step - loss: 4.1832e-04 - val_loss:
1.2552e-04
Epoch 58/100
45/45 ───────────────── 1s 17ms/step - loss: 4.1672e-04 - val_loss:
1.2483e-04
Epoch 59/100
45/45 ───────────────── 1s 17ms/step - loss: 4.1519e-04 - val_loss:
1.2418e-04
Epoch 60/100
45/45 ───────────────── 1s 17ms/step - loss: 4.1374e-04 - val_loss:
1.2357e-04
Epoch 61/100
45/45 ───────────────── 1s 16ms/step - loss: 4.1235e-04 - val_loss:
1.2299e-04
Epoch 62/100
45/45 ───────────────── 1s 17ms/step - loss: 4.1102e-04 - val_loss:
1.2245e-04
Epoch 63/100
45/45 ───────────────── 1s 16ms/step - loss: 4.0974e-04 - val_loss:
1.2194e-04
Epoch 64/100
45/45 ───────────────── 1s 18ms/step - loss: 4.0851e-04 - val_loss:
1.2146e-04
Epoch 65/100
45/45 ───────────────── 1s 18ms/step - loss: 4.0732e-04 - val_loss:
```

```
1.2102e-04
Epoch 66/100
45/45 ──────────────── 1s 22ms/step - loss: 4.0617e-04 - val_loss:
1.2060e-04
Epoch 67/100
45/45 ──────────────── 1s 23ms/step - loss: 4.0506e-04 - val_loss:
1.2022e-04
Epoch 68/100
45/45 ──────────────── 1s 18ms/step - loss: 4.0398e-04 - val_loss:
1.1985e-04
Epoch 69/100
45/45 ──────────────── 1s 17ms/step - loss: 4.0294e-04 - val_loss:
1.1952e-04
Epoch 70/100
45/45 ──────────────── 1s 18ms/step - loss: 4.0192e-04 - val_loss:
1.1921e-04
Epoch 71/100
45/45 ──────────────── 1s 17ms/step - loss: 4.0093e-04 - val_loss:
1.1892e-04
Epoch 72/100
45/45 ──────────────── 1s 18ms/step - loss: 3.9997e-04 - val_loss:
1.1865e-04
Epoch 73/100
45/45 ──────────────── 1s 16ms/step - loss: 3.9903e-04 - val_loss:
1.1841e-04
Epoch 74/100
45/45 ──────────────── 1s 16ms/step - loss: 3.9811e-04 - val_loss:
1.1818e-04
Epoch 75/100
45/45 ──────────────── 1s 18ms/step - loss: 3.9722e-04 - val_loss:
1.1797e-04
Epoch 76/100
45/45 ──────────────── 1s 16ms/step - loss: 3.9634e-04 - val_loss:
1.1778e-04
Epoch 77/100
45/45 ──────────────── 2s 22ms/step - loss: 3.9548e-04 - val_loss:
1.1761e-04
Epoch 78/100
45/45 ──────────────── 1s 22ms/step - loss: 3.9463e-04 - val_loss:
1.1746e-04
Epoch 79/100
45/45 ──────────────── 1s 18ms/step - loss: 3.9380e-04 - val_loss:
1.1732e-04
Epoch 80/100
45/45 ──────────────── 1s 18ms/step - loss: 3.9299e-04 - val_loss:
1.1719e-04
Epoch 81/100
45/45 ──────────────── 1s 17ms/step - loss: 3.9219e-04 - val_loss:
1.1708e-04
```

```
Epoch 82/100
45/45 ———————————————— 1s 17ms/step - loss: 3.9141e-04 - val_loss:
1.1698e-04
Epoch 83/100
45/45 ———————————————— 1s 18ms/step - loss: 3.9064e-04 - val_loss:
1.1690e-04
Epoch 84/100
45/45 ———————————————— 1s 17ms/step - loss: 3.8989e-04 - val_loss:
1.1682e-04
Epoch 85/100
45/45 ———————————————— 1s 18ms/step - loss: 3.8915e-04 - val_loss:
1.1676e-04
Epoch 86/100
45/45 ———————————————— 1s 18ms/step - loss: 3.8842e-04 - val_loss:
1.1671e-04
Epoch 87/100
45/45 ———————————————— 1s 19ms/step - loss: 3.8770e-04 - val_loss:
1.1667e-04
Epoch 88/100
45/45 ———————————————— 1s 24ms/step - loss: 3.8700e-04 - val_loss:
1.1664e-04
Epoch 89/100
45/45 ———————————————— 1s 23ms/step - loss: 3.8632e-04 - val_loss:
1.1662e-04
Epoch 90/100
45/45 ———————————————— 1s 18ms/step - loss: 3.8565e-04 - val_loss:
1.1660e-04
Epoch 91/100
45/45 ———————————————— 1s 19ms/step - loss: 3.8500e-04 - val_loss:
1.1659e-04
Epoch 92/100
45/45 ———————————————— 1s 19ms/step - loss: 3.8436e-04 - val_loss:
1.1659e-04
Epoch 93/100
45/45 ———————————————— 1s 17ms/step - loss: 3.8374e-04 - val_loss:
1.1659e-04
Epoch 94/100
45/45 ———————————————— 1s 18ms/step - loss: 3.8314e-04 - val_loss:
1.1659e-04
Epoch 95/100
45/45 ———————————————— 1s 17ms/step - loss: 3.8255e-04 - val_loss:
1.1659e-04
Epoch 96/100
45/45 ———————————————— 1s 17ms/step - loss: 3.8198e-04 - val_loss:
1.1660e-04
Epoch 97/100
45/45 ———————————————— 1s 18ms/step - loss: 3.8144e-04 - val_loss:
1.1661e-04
Epoch 98/100
```

```
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 3.8091e-04 - val_loss:
1.1662e-04
Epoch 99/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 2s 23ms/step - loss: 3.8040e-04 - val_loss:
1.1663e-04
Epoch 100/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 3.7992e-04 - val_loss:
1.1665e-04
LSTM Test loss: 0.00012131379480706528
GRU Test loss: 0.00011685465142363682
BiLSTM Test loss: 0.00011664778867270797
```
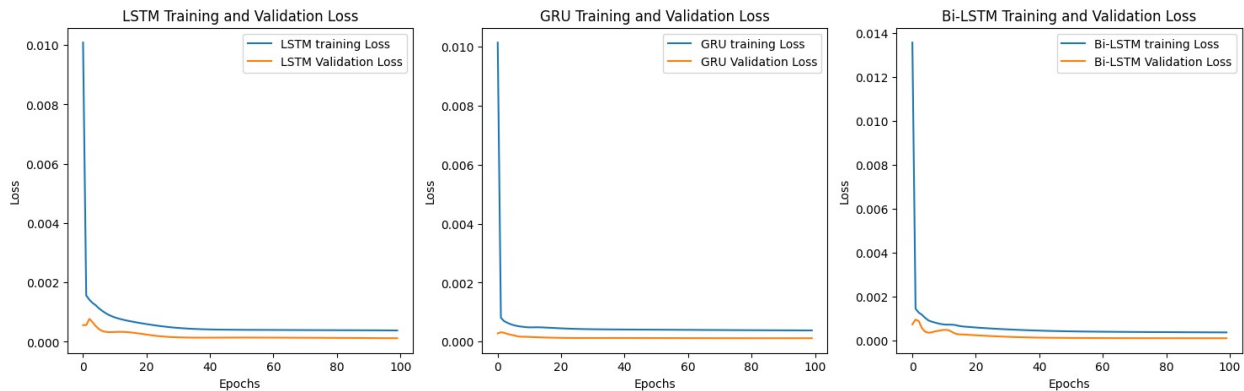


```
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 62ms/step
LSTM MSE: 0.00012131379587472078, MAE: 0.0075424415697561125
GRU MSE: 0.00011685464335143394, MAE: 0.007413831859346532
BiLSTM MSE: 0.00011664778040768777, MAE: 0.007337559388208527
```

For BTC Comparison of Actual and LSTM Predicted Prices

For BTC Comparison of Actual and GRU Predicted Prices

For BTC Comparison of Actual and Bi-LSTM Predicted Prices
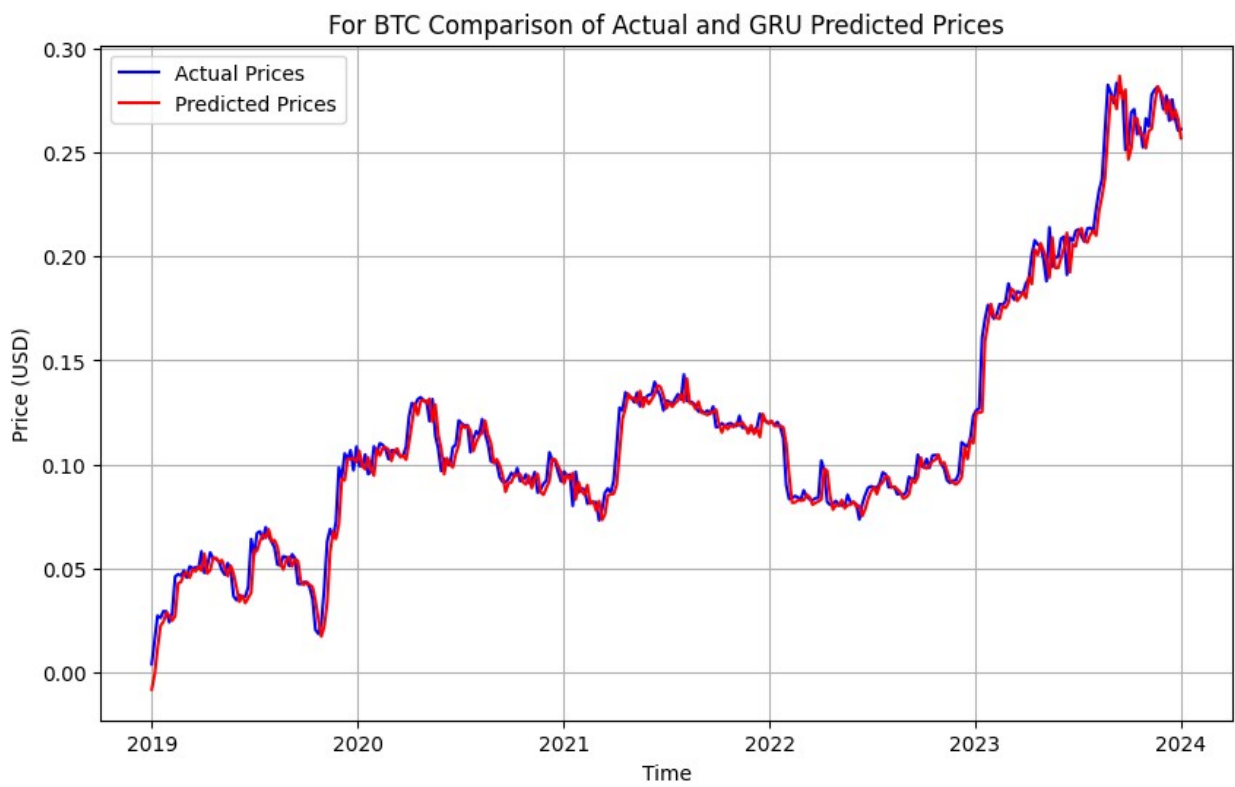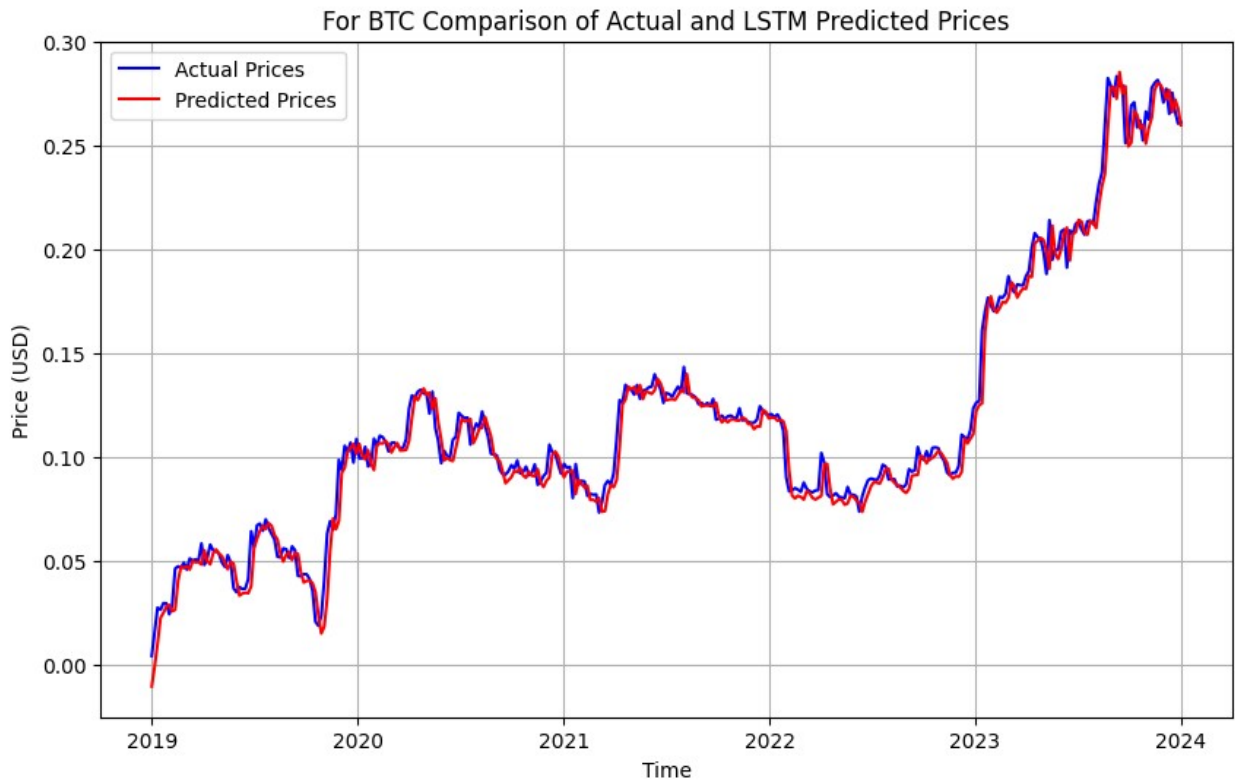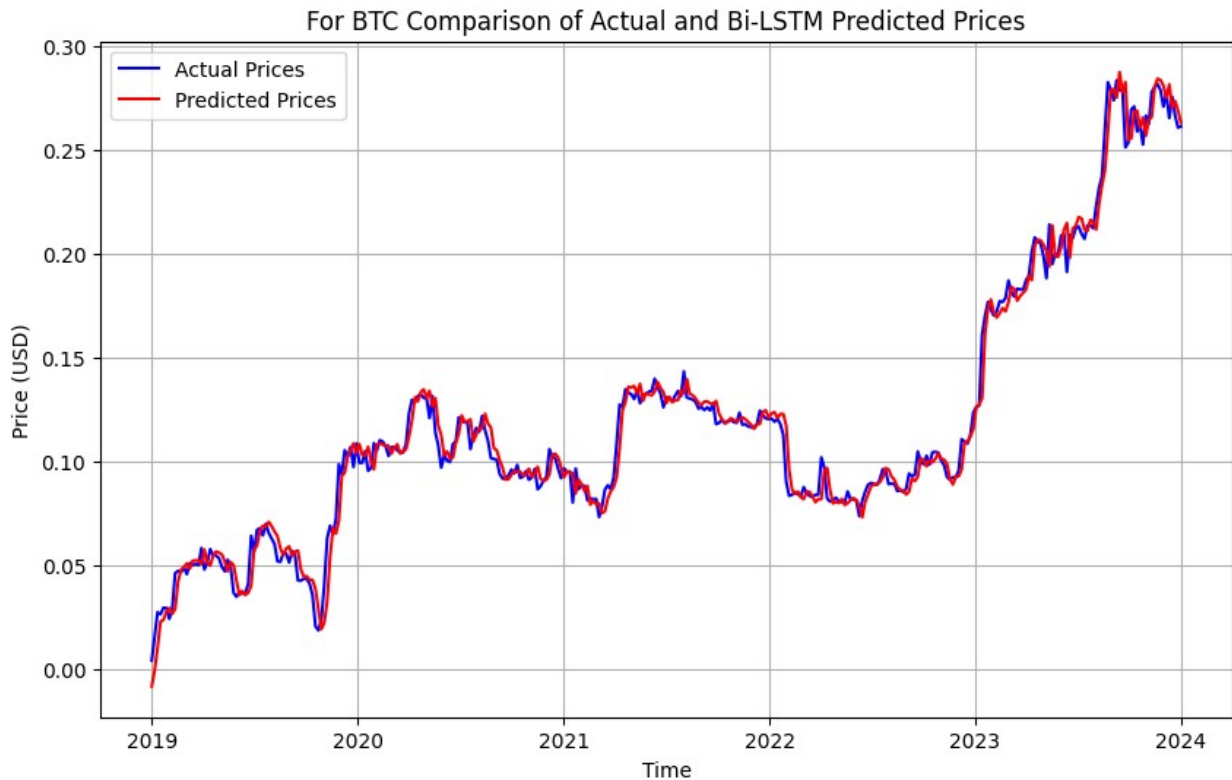
```
LSTM RMSE: 0.011, MAPE: 1.93%
GRU RMSE: 0.011, MAPE: 1.88%
BiLSTM RMSE: 0.011, MAPE: 1.88%
Data preprocessed and ready for model training.

<ipython-input-1-11932f61a7e0>:320: FutureWarning: DataFrame.fillna
with 'method' is deprecated and will raise in a future version. Use
obj.ffill() or obj.bfill() instead.
  eth_data.fillna(method='ffill', inplace=True)
/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.py:20
0: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a
layer. When using Sequential models, prefer using an `Input(shape)`
object as the first layer in the model instead.
  super().__init__(**kwargs)
/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/bidirecti
onal.py:107: UserWarning: Do not pass an `input_shape`/`input_dim`
argument to a layer. When using Sequential models, prefer using an
`Input(shape)` object as the first layer in the model instead.
  super().__init__(**kwargs)

Epoch 1/100
45/45 ──────────────────────── 2s 19ms/step - loss: 0.0160 - val_loss:
5.0220e-04
Epoch 2/100
45/45 ──────────────────────── 0s 10ms/step - loss: 0.0019 - val_loss:
4.0901e-04
```

```
Epoch 3/100
45/45 ──────────────── 0s 10ms/step - loss: 0.0016 - val_loss:
3.3181e-04
Epoch 4/100
45/45 ──────────────── 1s 10ms/step - loss: 0.0013 - val_loss:
3.9926e-04
Epoch 5/100
45/45 ──────────────── 1s 10ms/step - loss: 0.0012 - val_loss:
3.1581e-04
Epoch 6/100
45/45 ──────────────── 1s 12ms/step - loss: 0.0010 - val_loss:
2.3288e-04
Epoch 7/100
45/45 ──────────────── 1s 14ms/step - loss: 9.1697e-04 - val_loss:
2.0171e-04
Epoch 8/100
45/45 ──────────────── 1s 15ms/step - loss: 8.6296e-04 - val_loss:
1.9679e-04
Epoch 9/100
45/45 ──────────────── 1s 14ms/step - loss: 8.2146e-04 - val_loss:
1.9578e-04
Epoch 10/100
45/45 ──────────────── 1s 10ms/step - loss: 7.8520e-04 - val_loss:
1.9095e-04
Epoch 11/100
45/45 ──────────────── 1s 12ms/step - loss: 7.5318e-04 - val_loss:
1.8242e-04
Epoch 12/100
45/45 ──────────────── 1s 10ms/step - loss: 7.2488e-04 - val_loss:
1.7187e-04
Epoch 13/100
45/45 ──────────────── 1s 11ms/step - loss: 6.9989e-04 - val_loss:
1.6136e-04
Epoch 14/100
45/45 ──────────────── 1s 10ms/step - loss: 6.7754e-04 - val_loss:
1.5253e-04
Epoch 15/100
45/45 ──────────────── 1s 10ms/step - loss: 6.5694e-04 - val_loss:
1.4568e-04
Epoch 16/100
45/45 ──────────────── 1s 10ms/step - loss: 6.3736e-04 - val_loss:
1.4028e-04
Epoch 17/100
45/45 ──────────────── 1s 10ms/step - loss: 6.1837e-04 - val_loss:
1.3580e-04
Epoch 18/100
45/45 ──────────────── 0s 10ms/step - loss: 5.9982e-04 - val_loss:
1.3190e-04
Epoch 19/100
```

```
45/45 ——————————————— 0s 10ms/step - loss: 5.8183e-04 - val_loss:
1.2840e-04
Epoch 20/100
45/45 ——————————————— 1s 10ms/step - loss: 5.6459e-04 - val_loss:
1.2515e-04
Epoch 21/100
45/45 ——————————————— 0s 10ms/step - loss: 5.4836e-04 - val_loss:
1.2208e-04
Epoch 22/100
45/45 ——————————————— 1s 10ms/step - loss: 5.3337e-04 - val_loss:
1.1913e-04
Epoch 23/100
45/45 ——————————————— 1s 10ms/step - loss: 5.1981e-04 - val_loss:
1.1631e-04
Epoch 24/100
45/45 ——————————————— 0s 10ms/step - loss: 5.0782e-04 - val_loss:
1.1364e-04
Epoch 25/100
45/45 ——————————————— 0s 10ms/step - loss: 4.9743e-04 - val_loss:
1.1118e-04
Epoch 26/100
45/45 ——————————————— 1s 13ms/step - loss: 4.8856e-04 - val_loss:
1.0898e-04
Epoch 27/100
45/45 ——————————————— 1s 16ms/step - loss: 4.8107e-04 - val_loss:
1.0708e-04
Epoch 28/100
45/45 ——————————————— 0s 10ms/step - loss: 4.7474e-04 - val_loss:
1.0548e-04
Epoch 29/100
45/45 ——————————————— 0s 10ms/step - loss: 4.6939e-04 - val_loss:
1.0419e-04
Epoch 30/100
45/45 ——————————————— 1s 10ms/step - loss: 4.6483e-04 - val_loss:
1.0318e-04
Epoch 31/100
45/45 ——————————————— 1s 10ms/step - loss: 4.6091e-04 - val_loss:
1.0241e-04
Epoch 32/100
45/45 ——————————————— 0s 10ms/step - loss: 4.5753e-04 - val_loss:
1.0185e-04
Epoch 33/100
45/45 ——————————————— 1s 10ms/step - loss: 4.5460e-04 - val_loss:
1.0144e-04
Epoch 34/100
45/45 ——————————————— 1s 10ms/step - loss: 4.5202e-04 - val_loss:
1.0115e-04
Epoch 35/100
45/45 ——————————————— 0s 10ms/step - loss: 4.4973e-04 - val_loss:
```

```
1.0095e-04
Epoch 36/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.4768e-04 - val_loss:
1.0080e-04
Epoch 37/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.4581e-04 - val_loss:
1.0069e-04
Epoch 38/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.4410e-04 - val_loss:
1.0059e-04
Epoch 39/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.4250e-04 - val_loss:
1.0052e-04
Epoch 40/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.4099e-04 - val_loss:
1.0045e-04
Epoch 41/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.3955e-04 - val_loss:
1.0038e-04
Epoch 42/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.3817e-04 - val_loss:
1.0032e-04
Epoch 43/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 4.3683e-04 - val_loss:
1.0025e-04
Epoch 44/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 4.3553e-04 - val_loss:
1.0020e-04
Epoch 45/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.3427e-04 - val_loss:
1.0014e-04
Epoch 46/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 13ms/step - loss: 4.3303e-04 - val_loss:
1.0009e-04
Epoch 47/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - loss: 4.3182e-04 - val_loss:
1.0004e-04
Epoch 48/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 4.3063e-04 - val_loss:
9.9998e-05
Epoch 49/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.2946e-04 - val_loss:
9.9959e-05
Epoch 50/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.2831e-04 - val_loss:
9.9923e-05
Epoch 51/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.2719e-04 - val_loss:
9.9892e-05
```

```
Epoch 52/100
45/45 ──────────────────────── 1s 12ms/step - loss: 4.2609e-04 - val_loss:
9.9865e-05
Epoch 53/100
45/45 ──────────────────────── 1s 10ms/step - loss: 4.2501e-04 - val_loss:
9.9839e-05
Epoch 54/100
45/45 ──────────────────────── 0s 10ms/step - loss: 4.2395e-04 - val_loss:
9.9816e-05
Epoch 55/100
45/45 ──────────────────────── 0s 10ms/step - loss: 4.2293e-04 - val_loss:
9.9795e-05
Epoch 56/100
45/45 ──────────────────────── 0s 10ms/step - loss: 4.2193e-04 - val_loss:
9.9773e-05
Epoch 57/100
45/45 ──────────────────────── 1s 10ms/step - loss: 4.2097e-04 - val_loss:
9.9750e-05
Epoch 58/100
45/45 ──────────────────────── 0s 10ms/step - loss: 4.2005e-04 - val_loss:
9.9725e-05
Epoch 59/100
45/45 ──────────────────────── 0s 10ms/step - loss: 4.1917e-04 - val_loss:
9.9696e-05
Epoch 60/100
45/45 ──────────────────────── 1s 10ms/step - loss: 4.1834e-04 - val_loss:
9.9662e-05
Epoch 61/100
45/45 ──────────────────────── 0s 10ms/step - loss: 4.1757e-04 - val_loss:
9.9619e-05
Epoch 62/100
45/45 ──────────────────────── 1s 10ms/step - loss: 4.1686e-04 - val_loss:
9.9566e-05
Epoch 63/100
45/45 ──────────────────────── 0s 10ms/step - loss: 4.1623e-04 - val_loss:
9.9499e-05
Epoch 64/100
45/45 ──────────────────────── 1s 10ms/step - loss: 4.1567e-04 - val_loss:
9.9413e-05
Epoch 65/100
45/45 ──────────────────────── 0s 10ms/step - loss: 4.1521e-04 - val_loss:
9.9305e-05
Epoch 66/100
45/45 ──────────────────────── 1s 13ms/step - loss: 4.1485e-04 - val_loss:
9.9165e-05
Epoch 67/100
45/45 ──────────────────────── 1s 16ms/step - loss: 4.1458e-04 - val_loss:
9.8987e-05
Epoch 68/100
```

```
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - loss: 4.1443e-04 - val_loss:
9.8759e-05
Epoch 69/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.1438e-04 - val_loss:
9.8473e-05
Epoch 70/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.1444e-04 - val_loss:
9.8115e-05
Epoch 71/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.1458e-04 - val_loss:
9.7677e-05
Epoch 72/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.1479e-04 - val_loss:
9.7154e-05
Epoch 73/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.1503e-04 - val_loss:
9.6547e-05
Epoch 74/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 4.1526e-04 - val_loss:
9.5870e-05
Epoch 75/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.1542e-04 - val_loss:
9.5147e-05
Epoch 76/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.1546e-04 - val_loss:
9.4416e-05
Epoch 77/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.1530e-04 - val_loss:
9.3722e-05
Epoch 78/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.1492e-04 - val_loss:
9.3111e-05
Epoch 79/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.1428e-04 - val_loss:
9.2618e-05
Epoch 80/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.1339e-04 - val_loss:
9.2260e-05
Epoch 81/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.1230e-04 - val_loss:
9.2036e-05
Epoch 82/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.1106e-04 - val_loss:
9.1931e-05
Epoch 83/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.0972e-04 - val_loss:
9.1918e-05
Epoch 84/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.0834e-04 - val_loss:
```

```
9.1973e-05
Epoch 85/100
45/45 ━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.0698e-04 - val_loss:
9.2074e-05
Epoch 86/100
45/45 ━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 4.0565e-04 - val_loss:
9.2201e-05
Epoch 87/100
45/45 ━━━━━━━━━━━━━━━━━━ 1s 16ms/step - loss: 4.0437e-04 - val_loss:
9.2344e-05
Epoch 88/100
45/45 ━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 4.0314e-04 - val_loss:
9.2495e-05
Epoch 89/100
45/45 ━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.0198e-04 - val_loss:
9.2648e-05
Epoch 90/100
45/45 ━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.0087e-04 - val_loss:
9.2800e-05
Epoch 91/100
45/45 ━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 3.9981e-04 - val_loss:
9.2950e-05
Epoch 92/100
45/45 ━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 3.9879e-04 - val_loss:
9.3098e-05
Epoch 93/100
45/45 ━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 3.9782e-04 - val_loss:
9.3242e-05
Epoch 94/100
45/45 ━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 3.9688e-04 - val_loss:
9.3384e-05
Epoch 95/100
45/45 ━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 3.9598e-04 - val_loss:
9.3523e-05
Epoch 96/100
45/45 ━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 3.9511e-04 - val_loss:
9.3659e-05
Epoch 97/100
45/45 ━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 3.9426e-04 - val_loss:
9.3793e-05
Epoch 98/100
45/45 ━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 3.9344e-04 - val_loss:
9.3925e-05
Epoch 99/100
45/45 ━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 3.9264e-04 - val_loss:
9.4055e-05
Epoch 100/100
45/45 ━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 3.9187e-04 - val_loss:
9.4183e-05
```

```
Epoch 1/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 2s 16ms/step - loss: 0.0346 - val_loss:
5.4716e-04
Epoch 2/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 8.2486e-04 - val_loss:
1.5457e-04
Epoch 3/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - loss: 6.4241e-04 - val_loss:
1.6082e-04
Epoch 4/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 14ms/step - loss: 5.9707e-04 - val_loss:
1.6737e-04
Epoch 5/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 5.6019e-04 - val_loss:
1.5655e-04
Epoch 6/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 13ms/step - loss: 5.2509e-04 - val_loss:
1.4321e-04
Epoch 7/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 16ms/step - loss: 4.9702e-04 - val_loss:
1.4177e-04
Epoch 8/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.7835e-04 - val_loss:
1.4582e-04
Epoch 9/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.6388e-04 - val_loss:
1.5088e-04
Epoch 10/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.5087e-04 - val_loss:
1.4820e-04
Epoch 11/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.3968e-04 - val_loss:
1.3784e-04
Epoch 12/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.3173e-04 - val_loss:
1.2855e-04
Epoch 13/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.2625e-04 - val_loss:
1.2123e-04
Epoch 14/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.2311e-04 - val_loss:
1.1427e-04
Epoch 15/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.2193e-04 - val_loss:
1.0759e-04
Epoch 16/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.2219e-04 - val_loss:
1.0174e-04
Epoch 17/100
```

```
45/45 ──────────────────── 1s 10ms/step - loss: 4.2315e-04 - val_loss:
9.7343e-05
Epoch 18/100
45/45 ──────────────────── 0s 10ms/step - loss: 4.2395e-04 - val_loss:
9.4595e-05
Epoch 19/100
45/45 ──────────────────── 0s 10ms/step - loss: 4.2398e-04 - val_loss:
9.3193e-05
Epoch 20/100
45/45 ──────────────────── 0s 10ms/step - loss: 4.2309e-04 - val_loss:
9.2640e-05
Epoch 21/100
45/45 ──────────────────── 0s 10ms/step - loss: 4.2148e-04 - val_loss:
9.2499e-05
Epoch 22/100
45/45 ──────────────────── 1s 14ms/step - loss: 4.1953e-04 - val_loss:
9.2497e-05
Epoch 23/100
45/45 ──────────────────── 1s 14ms/step - loss: 4.1755e-04 - val_loss:
9.2503e-05
Epoch 24/100
45/45 ──────────────────── 0s 10ms/step - loss: 4.1580e-04 - val_loss:
9.2485e-05
Epoch 25/100
45/45 ──────────────────── 1s 10ms/step - loss: 4.1440e-04 - val_loss:
9.2455e-05
Epoch 26/100
45/45 ──────────────────── 0s 10ms/step - loss: 4.1337e-04 - val_loss:
9.2437e-05
Epoch 27/100
45/45 ──────────────────── 1s 10ms/step - loss: 4.1268e-04 - val_loss:
9.2447e-05
Epoch 28/100
45/45 ──────────────────── 0s 10ms/step - loss: 4.1225e-04 - val_loss:
9.2494e-05
Epoch 29/100
45/45 ──────────────────── 0s 10ms/step - loss: 4.1201e-04 - val_loss:
9.2577e-05
Epoch 30/100
45/45 ──────────────────── 0s 10ms/step - loss: 4.1189e-04 - val_loss:
9.2695e-05
Epoch 31/100
45/45 ──────────────────── 0s 10ms/step - loss: 4.1183e-04 - val_loss:
9.2843e-05
Epoch 32/100
45/45 ──────────────────── 1s 10ms/step - loss: 4.1181e-04 - val_loss:
9.3015e-05
Epoch 33/100
45/45 ──────────────────── 1s 10ms/step - loss: 4.1178e-04 - val_loss:
```

```
9.3203e-05
Epoch 34/100
45/45 ——————————————— 1s 10ms/step - loss: 4.1173e-04 - val_loss:
9.3402e-05
Epoch 35/100
45/45 ——————————————— 1s 10ms/step - loss: 4.1164e-04 - val_loss:
9.3606e-05
Epoch 36/100
45/45 ——————————————— 1s 12ms/step - loss: 4.1151e-04 - val_loss:
9.3809e-05
Epoch 37/100
45/45 ——————————————— 1s 22ms/step - loss: 4.1134e-04 - val_loss:
9.4008e-05
Epoch 38/100
45/45 ——————————————— 1s 17ms/step - loss: 4.1112e-04 - val_loss:
9.4202e-05
Epoch 39/100
45/45 ——————————————— 0s 10ms/step - loss: 4.1086e-04 - val_loss:
9.4389e-05
Epoch 40/100
45/45 ——————————————— 1s 12ms/step - loss: 4.1056e-04 - val_loss:
9.4568e-05
Epoch 41/100
45/45 ——————————————— 1s 15ms/step - loss: 4.1024e-04 - val_loss:
9.4741e-05
Epoch 42/100
45/45 ——————————————— 1s 13ms/step - loss: 4.0988e-04 - val_loss:
9.4909e-05
Epoch 43/100
45/45 ——————————————— 0s 10ms/step - loss: 4.0951e-04 - val_loss:
9.5072e-05
Epoch 44/100
45/45 ——————————————— 1s 10ms/step - loss: 4.0912e-04 - val_loss:
9.5233e-05
Epoch 45/100
45/45 ——————————————— 1s 10ms/step - loss: 4.0871e-04 - val_loss:
9.5393e-05
Epoch 46/100
45/45 ——————————————— 0s 10ms/step - loss: 4.0830e-04 - val_loss:
9.5555e-05
Epoch 47/100
45/45 ——————————————— 1s 10ms/step - loss: 4.0788e-04 - val_loss:
9.5720e-05
Epoch 48/100
45/45 ——————————————— 0s 10ms/step - loss: 4.0746e-04 - val_loss:
9.5889e-05
Epoch 49/100
45/45 ——————————————— 1s 10ms/step - loss: 4.0704e-04 - val_loss:
9.6066e-05
```

```
Epoch 50/100
45/45 ———————————————— 1s 10ms/step - loss: 4.0662e-04 - val_loss:
9.6251e-05
Epoch 51/100
45/45 ———————————————— 0s 10ms/step - loss: 4.0621e-04 - val_loss:
9.6445e-05
Epoch 52/100
45/45 ———————————————— 1s 10ms/step - loss: 4.0581e-04 - val_loss:
9.6651e-05
Epoch 53/100
45/45 ———————————————— 0s 10ms/step - loss: 4.0542e-04 - val_loss:
9.6869e-05
Epoch 54/100
45/45 ———————————————— 1s 10ms/step - loss: 4.0504e-04 - val_loss:
9.7101e-05
Epoch 55/100
45/45 ———————————————— 1s 10ms/step - loss: 4.0467e-04 - val_loss:
9.7347e-05
Epoch 56/100
45/45 ———————————————— 1s 10ms/step - loss: 4.0431e-04 - val_loss:
9.7609e-05
Epoch 57/100
45/45 ———————————————— 0s 10ms/step - loss: 4.0397e-04 - val_loss:
9.7887e-05
Epoch 58/100
45/45 ———————————————— 1s 10ms/step - loss: 4.0364e-04 - val_loss:
9.8183e-05
Epoch 59/100
45/45 ———————————————— 0s 10ms/step - loss: 4.0333e-04 - val_loss:
9.8498e-05
Epoch 60/100
45/45 ———————————————— 1s 15ms/step - loss: 4.0303e-04 - val_loss:
9.8831e-05
Epoch 61/100
45/45 ———————————————— 1s 16ms/step - loss: 4.0275e-04 - val_loss:
9.9184e-05
Epoch 62/100
45/45 ———————————————— 1s 15ms/step - loss: 4.0249e-04 - val_loss:
9.9557e-05
Epoch 63/100
45/45 ———————————————— 1s 13ms/step - loss: 4.0225e-04 - val_loss:
9.9951e-05
Epoch 64/100
45/45 ———————————————— 0s 10ms/step - loss: 4.0203e-04 - val_loss:
1.0037e-04
Epoch 65/100
45/45 ———————————————— 1s 10ms/step - loss: 4.0182e-04 - val_loss:
1.0080e-04
Epoch 66/100
45/45 ———————————————— 1s 10ms/step - loss: 4.0164e-04 - val_loss:
```

```
1.0126e-04
Epoch 67/100
45/45 ──────────────── 0s 10ms/step - loss: 4.0148e-04 - val_loss:
1.0175e-04
Epoch 68/100
45/45 ──────────────── 1s 10ms/step - loss: 4.0135e-04 - val_loss:
1.0225e-04
Epoch 69/100
45/45 ──────────────── 1s 10ms/step - loss: 4.0124e-04 - val_loss:
1.0278e-04
Epoch 70/100
45/45 ──────────────── 1s 10ms/step - loss: 4.0116e-04 - val_loss:
1.0333e-04
Epoch 71/100
45/45 ──────────────── 0s 10ms/step - loss: 4.0111e-04 - val_loss:
1.0391e-04
Epoch 72/100
45/45 ──────────────── 1s 11ms/step - loss: 4.0110e-04 - val_loss:
1.0451e-04
Epoch 73/100
45/45 ──────────────── 1s 11ms/step - loss: 4.0112e-04 - val_loss:
1.0513e-04
Epoch 74/100
45/45 ──────────────── 0s 10ms/step - loss: 4.0120e-04 - val_loss:
1.0578e-04
Epoch 75/100
45/45 ──────────────── 1s 12ms/step - loss: 4.0132e-04 - val_loss:
1.0645e-04
Epoch 76/100
45/45 ──────────────── 1s 11ms/step - loss: 4.0149e-04 - val_loss:
1.0714e-04
Epoch 77/100
45/45 ──────────────── 1s 12ms/step - loss: 4.0173e-04 - val_loss:
1.0785e-04
Epoch 78/100
45/45 ──────────────── 1s 10ms/step - loss: 4.0205e-04 - val_loss:
1.0858e-04
Epoch 79/100
45/45 ──────────────── 0s 10ms/step - loss: 4.0245e-04 - val_loss:
1.0931e-04
Epoch 80/100
45/45 ──────────────── 0s 10ms/step - loss: 4.0296e-04 - val_loss:
1.1005e-04
Epoch 81/100
45/45 ──────────────── 1s 13ms/step - loss: 4.0358e-04 - val_loss:
1.1077e-04
Epoch 82/100
45/45 ──────────────── 1s 15ms/step - loss: 4.0434e-04 - val_loss:
1.1146e-04
```

```
Epoch 83/100
45/45 ──────────────── 1s 11ms/step - loss: 4.0527e-04 - val_loss:
1.1208e-04
Epoch 84/100
45/45 ──────────────── 0s 10ms/step - loss: 4.0639e-04 - val_loss:
1.1260e-04
Epoch 85/100
45/45 ──────────────── 1s 10ms/step - loss: 4.0774e-04 - val_loss:
1.1293e-04
Epoch 86/100
45/45 ──────────────── 0s 10ms/step - loss: 4.0933e-04 - val_loss:
1.1299e-04
Epoch 87/100
45/45 ──────────────── 0s 10ms/step - loss: 4.1120e-04 - val_loss:
1.1266e-04
Epoch 88/100
45/45 ──────────────── 0s 10ms/step - loss: 4.1333e-04 - val_loss:
1.1177e-04
Epoch 89/100
45/45 ──────────────── 1s 10ms/step - loss: 4.1568e-04 - val_loss:
1.1017e-04
Epoch 90/100
45/45 ──────────────── 0s 10ms/step - loss: 4.1811e-04 - val_loss:
1.0774e-04
Epoch 91/100
45/45 ──────────────── 0s 10ms/step - loss: 4.2035e-04 - val_loss:
1.0454e-04
Epoch 92/100
45/45 ──────────────── 1s 10ms/step - loss: 4.2200e-04 - val_loss:
1.0087e-04
Epoch 93/100
45/45 ──────────────── 0s 10ms/step - loss: 4.2259e-04 - val_loss:
9.7270e-05
Epoch 94/100
45/45 ──────────────── 0s 10ms/step - loss: 4.2177e-04 - val_loss:
9.4334e-05
Epoch 95/100
45/45 ──────────────── 1s 10ms/step - loss: 4.1955e-04 - val_loss:
9.2386e-05
Epoch 96/100
45/45 ──────────────── 1s 10ms/step - loss: 4.1635e-04 - val_loss:
9.1362e-05
Epoch 97/100
45/45 ──────────────── 0s 10ms/step - loss: 4.1281e-04 - val_loss:
9.0972e-05
Epoch 98/100
45/45 ──────────────── 0s 10ms/step - loss: 4.0943e-04 - val_loss:
9.0923e-05
Epoch 99/100
```

```
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.0647e-04 - val_loss:
9.1027e-05
Epoch 100/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.0398e-04 - val_loss:
9.1191e-05
Epoch 1/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 5s 28ms/step - loss: 0.0187 - val_loss:
0.0012
Epoch 2/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 2s 19ms/step - loss: 0.0018 - val_loss:
3.0169e-04
Epoch 3/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 0.0013 - val_loss:
4.2497e-04
Epoch 4/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 0.0011 - val_loss:
2.3558e-04
Epoch 5/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 9.1488e-04 - val_loss:
2.0867e-04
Epoch 6/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 8.5681e-04 - val_loss:
2.0705e-04
Epoch 7/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 8.0746e-04 - val_loss:
1.9538e-04
Epoch 8/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 7.7083e-04 - val_loss:
1.7636e-04
Epoch 9/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 21ms/step - loss: 7.4555e-04 - val_loss:
1.6237e-04
Epoch 10/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 23ms/step - loss: 7.1840e-04 - val_loss:
1.5273e-04
Epoch 11/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 6.9417e-04 - val_loss:
1.4554e-04
Epoch 12/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 6.7301e-04 - val_loss:
1.4019e-04
Epoch 13/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 6.5426e-04 - val_loss:
1.3601e-04
Epoch 14/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 6.3747e-04 - val_loss:
1.3260e-04
Epoch 15/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 6.2221e-04 - val_loss:
```

```
1.2973e-04
Epoch 16/100
45/45 ━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 6.0816e-04 - val_loss:
1.2726e-04
Epoch 17/100
45/45 ━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 5.9510e-04 - val_loss:
1.2508e-04
Epoch 18/100
45/45 ━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 5.8290e-04 - val_loss:
1.2311e-04
Epoch 19/100
45/45 ━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 5.7142e-04 - val_loss:
1.2129e-04
Epoch 20/100
45/45 ━━━━━━━━━━━━━━━━ 1s 22ms/step - loss: 5.6060e-04 - val_loss:
1.1956e-04
Epoch 21/100
45/45 ━━━━━━━━━━━━━━━━ 1s 23ms/step - loss: 5.5039e-04 - val_loss:
1.1791e-04
Epoch 22/100
45/45 ━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 5.4075e-04 - val_loss:
1.1631e-04
Epoch 23/100
45/45 ━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 5.3166e-04 - val_loss:
1.1475e-04
Epoch 24/100
45/45 ━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 5.2310e-04 - val_loss:
1.1324e-04
Epoch 25/100
45/45 ━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 5.1505e-04 - val_loss:
1.1178e-04
Epoch 26/100
45/45 ━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 5.0751e-04 - val_loss:
1.1036e-04
Epoch 27/100
45/45 ━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 5.0046e-04 - val_loss:
1.0900e-04
Epoch 28/100
45/45 ━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 4.9389e-04 - val_loss:
1.0770e-04
Epoch 29/100
45/45 ━━━━━━━━━━━━━━━━ 1s 16ms/step - loss: 4.8778e-04 - val_loss:
1.0647e-04
Epoch 30/100
45/45 ━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 4.8212e-04 - val_loss:
1.0530e-04
Epoch 31/100
45/45 ━━━━━━━━━━━━━━━━ 2s 24ms/step - loss: 4.7688e-04 - val_loss:
1.0421e-04
```

```
Epoch 32/100
45/45 ──────────────── 1s 24ms/step - loss: 4.7204e-04 - val_loss:
1.0319e-04
Epoch 33/100
45/45 ──────────────── 1s 19ms/step - loss: 4.6757e-04 - val_loss:
1.0225e-04
Epoch 34/100
45/45 ──────────────── 1s 18ms/step - loss: 4.6345e-04 - val_loss:
1.0137e-04
Epoch 35/100
45/45 ──────────────── 1s 17ms/step - loss: 4.5965e-04 - val_loss:
1.0057e-04
Epoch 36/100
45/45 ──────────────── 1s 17ms/step - loss: 4.5615e-04 - val_loss:
9.9845e-05
Epoch 37/100
45/45 ──────────────── 1s 17ms/step - loss: 4.5292e-04 - val_loss:
9.9185e-05
Epoch 38/100
45/45 ──────────────── 1s 18ms/step - loss: 4.4994e-04 - val_loss:
9.8589e-05
Epoch 39/100
45/45 ──────────────── 1s 17ms/step - loss: 4.4717e-04 - val_loss:
9.8055e-05
Epoch 40/100
45/45 ──────────────── 1s 17ms/step - loss: 4.4461e-04 - val_loss:
9.7580e-05
Epoch 41/100
45/45 ──────────────── 1s 18ms/step - loss: 4.4223e-04 - val_loss:
9.7160e-05
Epoch 42/100
45/45 ──────────────── 1s 18ms/step - loss: 4.4002e-04 - val_loss:
9.6791e-05
Epoch 43/100
45/45 ──────────────── 1s 23ms/step - loss: 4.3795e-04 - val_loss:
9.6470e-05
Epoch 44/100
45/45 ──────────────── 1s 22ms/step - loss: 4.3603e-04 - val_loss:
9.6193e-05
Epoch 45/100
45/45 ──────────────── 1s 17ms/step - loss: 4.3422e-04 - val_loss:
9.5957e-05
Epoch 46/100
45/45 ──────────────── 1s 17ms/step - loss: 4.3253e-04 - val_loss:
9.5759e-05
Epoch 47/100
45/45 ──────────────── 1s 17ms/step - loss: 4.3095e-04 - val_loss:
9.5595e-05
Epoch 48/100
```

```
45/45 ———————————— 1s 17ms/step - loss: 4.2946e-04 - val_loss:
9.5463e-05
Epoch 49/100
45/45 ———————————— 1s 18ms/step - loss: 4.2806e-04 - val_loss:
9.5359e-05
Epoch 50/100
45/45 ———————————— 1s 18ms/step - loss: 4.2674e-04 - val_loss:
9.5281e-05
Epoch 51/100
45/45 ———————————— 1s 17ms/step - loss: 4.2550e-04 - val_loss:
9.5226e-05
Epoch 52/100
45/45 ———————————— 1s 17ms/step - loss: 4.2432e-04 - val_loss:
9.5191e-05
Epoch 53/100
45/45 ———————————— 1s 19ms/step - loss: 4.2320e-04 - val_loss:
9.5174e-05
Epoch 54/100
45/45 ———————————— 1s 24ms/step - loss: 4.2213e-04 - val_loss:
9.5174e-05
Epoch 55/100
45/45 ———————————— 1s 17ms/step - loss: 4.2111e-04 - val_loss:
9.5188e-05
Epoch 56/100
45/45 ———————————— 1s 17ms/step - loss: 4.2014e-04 - val_loss:
9.5214e-05
Epoch 57/100
45/45 ———————————— 1s 18ms/step - loss: 4.1920e-04 - val_loss:
9.5251e-05
Epoch 58/100
45/45 ———————————— 1s 17ms/step - loss: 4.1831e-04 - val_loss:
9.5298e-05
Epoch 59/100
45/45 ———————————— 1s 18ms/step - loss: 4.1744e-04 - val_loss:
9.5352e-05
Epoch 60/100
45/45 ———————————— 1s 18ms/step - loss: 4.1660e-04 - val_loss:
9.5414e-05
Epoch 61/100
45/45 ———————————— 1s 18ms/step - loss: 4.1578e-04 - val_loss:
9.5482e-05
Epoch 62/100
45/45 ———————————— 1s 18ms/step - loss: 4.1499e-04 - val_loss:
9.5555e-05
Epoch 63/100
45/45 ———————————— 1s 18ms/step - loss: 4.1421e-04 - val_loss:
9.5633e-05
Epoch 64/100
45/45 ———————————— 1s 18ms/step - loss: 4.1346e-04 - val_loss:
```

```
9.5715e-05
Epoch 65/100
45/45 ━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 4.1271e-04 - val_loss:
9.5799e-05
Epoch 66/100
45/45 ━━━━━━━━━━━━━━━━━ 2s 23ms/step - loss: 4.1198e-04 - val_loss:
9.5887e-05
Epoch 67/100
45/45 ━━━━━━━━━━━━━━━━━ 1s 23ms/step - loss: 4.1126e-04 - val_loss:
9.5977e-05
Epoch 68/100
45/45 ━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 4.1056e-04 - val_loss:
9.6069e-05
Epoch 69/100
45/45 ━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 4.0986e-04 - val_loss:
9.6163e-05
Epoch 70/100
45/45 ━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 4.0917e-04 - val_loss:
9.6259e-05
Epoch 71/100
45/45 ━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 4.0849e-04 - val_loss:
9.6357e-05
Epoch 72/100
45/45 ━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 4.0781e-04 - val_loss:
9.6456e-05
Epoch 73/100
45/45 ━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 4.0715e-04 - val_loss:
9.6558e-05
Epoch 74/100
45/45 ━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 4.0650e-04 - val_loss:
9.6661e-05
Epoch 75/100
45/45 ━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 4.0586e-04 - val_loss:
9.6769e-05
Epoch 76/100
45/45 ━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 4.0524e-04 - val_loss:
9.6879e-05
Epoch 77/100
45/45 ━━━━━━━━━━━━━━━━━ 2s 22ms/step - loss: 4.0463e-04 - val_loss:
9.6994e-05
Epoch 78/100
45/45 ━━━━━━━━━━━━━━━━━ 1s 22ms/step - loss: 4.0405e-04 - val_loss:
9.7113e-05
Epoch 79/100
45/45 ━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 4.0348e-04 - val_loss:
9.7237e-05
Epoch 80/100
45/45 ━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 4.0294e-04 - val_loss:
9.7366e-05
```

```
Epoch 81/100
45/45 ──────────────────── 1s 18ms/step - loss: 4.0243e-04 - val_loss:
9.7500e-05
Epoch 82/100
45/45 ──────────────────── 1s 18ms/step - loss: 4.0193e-04 - val_loss:
9.7641e-05
Epoch 83/100
45/45 ──────────────────── 1s 18ms/step - loss: 4.0147e-04 - val_loss:
9.7789e-05
Epoch 84/100
45/45 ──────────────────── 1s 17ms/step - loss: 4.0102e-04 - val_loss:
9.7942e-05
Epoch 85/100
45/45 ──────────────────── 1s 17ms/step - loss: 4.0060e-04 - val_loss:
9.8101e-05
Epoch 86/100
45/45 ──────────────────── 1s 17ms/step - loss: 4.0020e-04 - val_loss:
9.8268e-05
Epoch 87/100
45/45 ──────────────────── 1s 18ms/step - loss: 3.9982e-04 - val_loss:
9.8441e-05
Epoch 88/100
45/45 ──────────────────── 1s 20ms/step - loss: 3.9946e-04 - val_loss:
9.8620e-05
Epoch 89/100
45/45 ──────────────────── 1s 23ms/step - loss: 3.9913e-04 - val_loss:
9.8806e-05
Epoch 90/100
45/45 ──────────────────── 1s 19ms/step - loss: 3.9882e-04 - val_loss:
9.8996e-05
Epoch 91/100
45/45 ──────────────────── 1s 17ms/step - loss: 3.9853e-04 - val_loss:
9.9192e-05
Epoch 92/100
45/45 ──────────────────── 1s 17ms/step - loss: 3.9827e-04 - val_loss:
9.9391e-05
Epoch 93/100
45/45 ──────────────────── 1s 17ms/step - loss: 3.9805e-04 - val_loss:
9.9593e-05
Epoch 94/100
45/45 ──────────────────── 1s 19ms/step - loss: 3.9785e-04 - val_loss:
9.9795e-05
Epoch 95/100
45/45 ──────────────────── 1s 18ms/step - loss: 3.9768e-04 - val_loss:
9.9996e-05
Epoch 96/100
45/45 ──────────────────── 1s 19ms/step - loss: 3.9756e-04 - val_loss:
1.0019e-04
Epoch 97/100
```

```
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 3.9748e-04 - val_loss:
1.0038e-04
Epoch 98/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 3.9744e-04 - val_loss:
1.0055e-04
Epoch 99/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 3.9745e-04 - val_loss:
1.0070e-04
Epoch 100/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 22ms/step - loss: 3.9753e-04 - val_loss:
1.0083e-04
LSTM Test loss: 9.418311674380675e-05
GRU Test loss: 9.119144669966772e-05
BiLSTM Test loss: 0.00010082648077514023
```
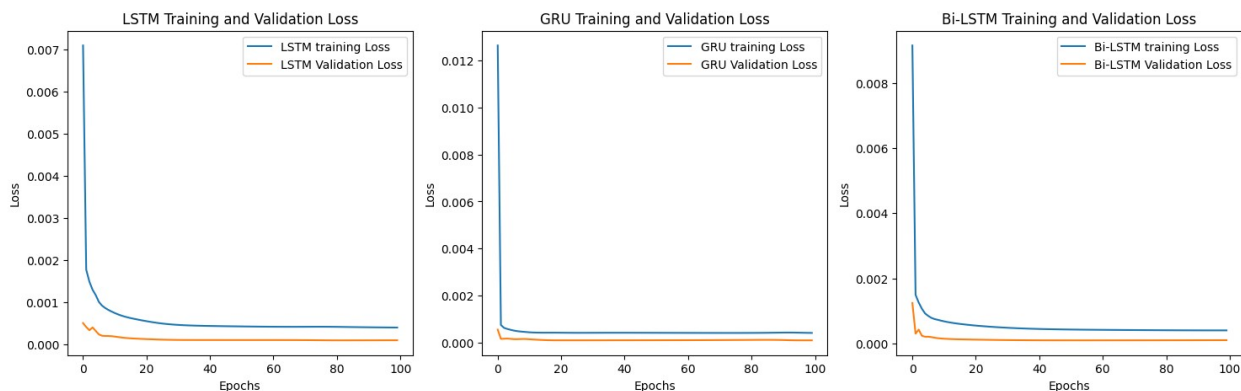


```
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 35ms/step
LSTM MSE: 9.418312279942257e-05, MAE: 0.00690689632307667
GRU MSE: 9.119144781079826e-05, MAE: 0.006710929154621598
BiLSTM MSE: 0.00010082648002455977, MAE: 0.007086421305006641
```

For ETH Comparison of Actual and LSTM Predicted Prices

For ETH Comparison of Actual and GRU Predicted Prices

For ETH Comparison of Actual and Bi-LSTM Predicted Prices
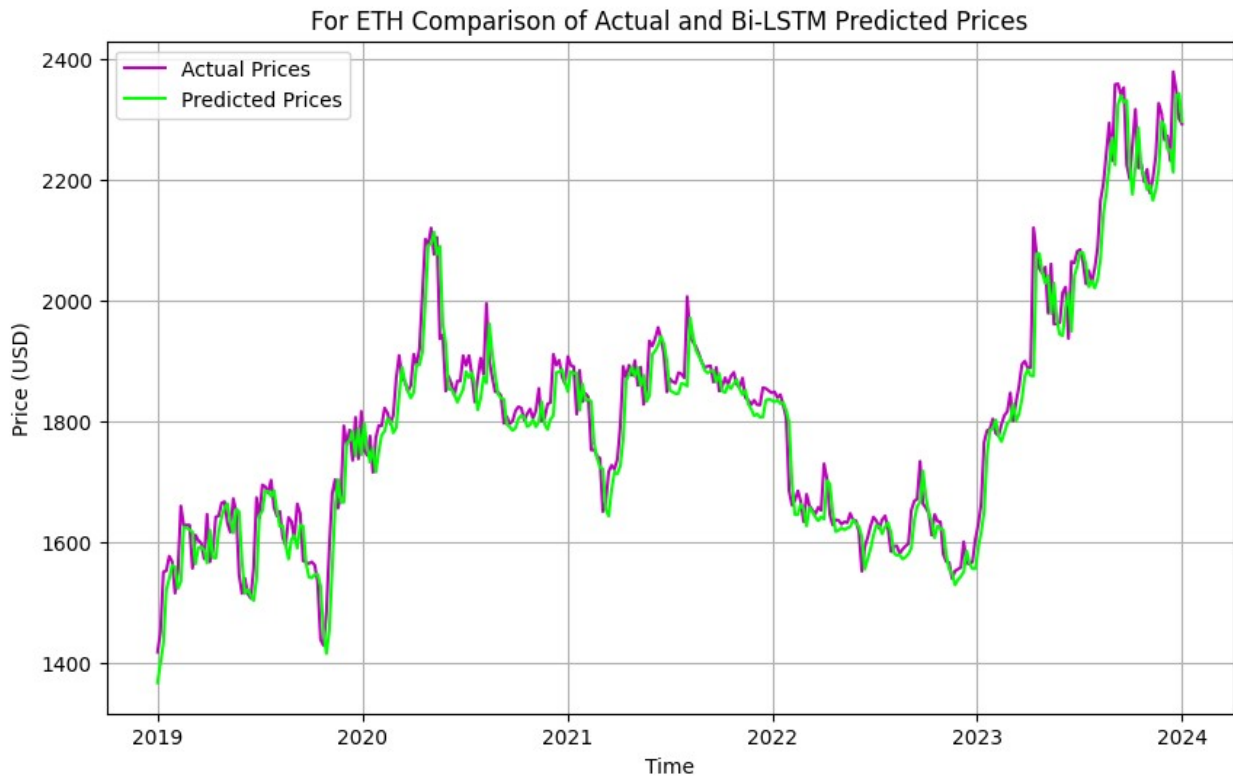

```
LSTM RMSE: 0.010, MAPE: 1.91%
GRU RMSE: 0.010, MAPE: 1.85%
BiLSTM RMSE: 0.010, MAPE: 1.95%
Data preprocessed and ready for model training.

<ipython-input-1-11932f61a7e0>:546: FutureWarning: DataFrame.fillna
with 'method' is deprecated and will raise in a future version. Use
obj.ffill() or obj.bfill() instead.
  ltc_data.fillna(method='ffill', inplace=True)
/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.py:20
0: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a
layer. When using Sequential models, prefer using an `Input(shape)`
object as the first layer in the model instead.
  super().__init__(**kwargs)
/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/bidirecti
onal.py:107: UserWarning: Do not pass an `input_shape`/`input_dim`
argument to a layer. When using Sequential models, prefer using an
`Input(shape)` object as the first layer in the model instead.
  super().__init__(**kwargs)

Epoch 1/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 2s 19ms/step - loss: 0.0067 - val_loss:
4.3445e-04
Epoch 2/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 0.0015 - val_loss:
2.3623e-04
```

```
Epoch 3/100
45/45 ———————————————— 1s 10ms/step - loss: 0.0012 - val_loss:
1.8183e-04
Epoch 4/100
45/45 ———————————————— 0s 10ms/step - loss: 0.0011 - val_loss:
1.7744e-04
Epoch 5/100
45/45 ———————————————— 1s 10ms/step - loss: 9.7333e-04 - val_loss:
1.5205e-04
Epoch 6/100
45/45 ———————————————— 1s 10ms/step - loss: 9.4309e-04 - val_loss:
1.4538e-04
Epoch 7/100
45/45 ———————————————— 1s 10ms/step - loss: 9.1258e-04 - val_loss:
1.4274e-04
Epoch 8/100
45/45 ———————————————— 1s 11ms/step - loss: 8.9886e-04 - val_loss:
1.4165e-04
Epoch 9/100
45/45 ———————————————— 1s 14ms/step - loss: 8.8982e-04 - val_loss:
1.4131e-04
Epoch 10/100
45/45 ———————————————— 1s 12ms/step - loss: 8.8233e-04 - val_loss:
1.4106e-04
Epoch 11/100
45/45 ———————————————— 1s 10ms/step - loss: 8.7641e-04 - val_loss:
1.4028e-04
Epoch 12/100
45/45 ———————————————— 1s 10ms/step - loss: 8.7142e-04 - val_loss:
1.3833e-04
Epoch 13/100
45/45 ———————————————— 0s 10ms/step - loss: 8.6638e-04 - val_loss:
1.3476e-04
Epoch 14/100
45/45 ———————————————— 0s 10ms/step - loss: 8.5988e-04 - val_loss:
1.2949e-04
Epoch 15/100
45/45 ———————————————— 0s 10ms/step - loss: 8.5027e-04 - val_loss:
1.2304e-04
Epoch 16/100
45/45 ———————————————— 0s 10ms/step - loss: 8.3584e-04 - val_loss:
1.1651e-04
Epoch 17/100
45/45 ———————————————— 1s 11ms/step - loss: 8.1538e-04 - val_loss:
1.1116e-04
Epoch 18/100
45/45 ———————————————— 1s 10ms/step - loss: 7.8895e-04 - val_loss:
1.0771e-04
Epoch 19/100
```

```
45/45 ──────────────────── 1s 10ms/step - loss: 7.5829e-04 - val_loss:
1.0609e-04
Epoch 20/100
45/45 ──────────────────── 1s 10ms/step - loss: 7.2619e-04 - val_loss:
1.0564e-04
Epoch 21/100
45/45 ──────────────────── 1s 10ms/step - loss: 6.9526e-04 - val_loss:
1.0570e-04
Epoch 22/100
45/45 ──────────────────── 1s 10ms/step - loss: 6.6714e-04 - val_loss:
1.0590e-04
Epoch 23/100
45/45 ──────────────────── 1s 10ms/step - loss: 6.4248e-04 - val_loss:
1.0612e-04
Epoch 24/100
45/45 ──────────────────── 0s 10ms/step - loss: 6.2133e-04 - val_loss:
1.0640e-04
Epoch 25/100
45/45 ──────────────────── 0s 10ms/step - loss: 6.0348e-04 - val_loss:
1.0682e-04
Epoch 26/100
45/45 ──────────────────── 0s 10ms/step - loss: 5.8865e-04 - val_loss:
1.0748e-04
Epoch 27/100
45/45 ──────────────────── 0s 10ms/step - loss: 5.7649e-04 - val_loss:
1.0842e-04
Epoch 28/100
45/45 ──────────────────── 1s 14ms/step - loss: 5.6660e-04 - val_loss:
1.0965e-04
Epoch 29/100
45/45 ──────────────────── 1s 16ms/step - loss: 5.5853e-04 - val_loss:
1.1115e-04
Epoch 30/100
45/45 ──────────────────── 1s 11ms/step - loss: 5.5185e-04 - val_loss:
1.1287e-04
Epoch 31/100
45/45 ──────────────────── 0s 10ms/step - loss: 5.4616e-04 - val_loss:
1.1473e-04
Epoch 32/100
45/45 ──────────────────── 1s 10ms/step - loss: 5.4120e-04 - val_loss:
1.1667e-04
Epoch 33/100
45/45 ──────────────────── 1s 12ms/step - loss: 5.3673e-04 - val_loss:
1.1867e-04
Epoch 34/100
45/45 ──────────────────── 0s 10ms/step - loss: 5.3260e-04 - val_loss:
1.2069e-04
Epoch 35/100
45/45 ──────────────────── 1s 10ms/step - loss: 5.2872e-04 - val_loss:
```

```
1.2276e-04
Epoch 36/100
45/45 ──────────────── 1s 10ms/step - loss: 5.2501e-04 - val_loss:
1.2488e-04
Epoch 37/100
45/45 ──────────────── 1s 10ms/step - loss: 5.2144e-04 - val_loss:
1.2707e-04
Epoch 38/100
45/45 ──────────────── 0s 10ms/step - loss: 5.1799e-04 - val_loss:
1.2935e-04
Epoch 39/100
45/45 ──────────────── 1s 10ms/step - loss: 5.1464e-04 - val_loss:
1.3177e-04
Epoch 40/100
45/45 ──────────────── 0s 10ms/step - loss: 5.1139e-04 - val_loss:
1.3436e-04
Epoch 41/100
45/45 ──────────────── 0s 10ms/step - loss: 5.0824e-04 - val_loss:
1.3720e-04
Epoch 42/100
45/45 ──────────────── 1s 10ms/step - loss: 5.0518e-04 - val_loss:
1.4039e-04
Epoch 43/100
45/45 ──────────────── 0s 10ms/step - loss: 5.0223e-04 - val_loss:
1.4404e-04
Epoch 44/100
45/45 ──────────────── 1s 10ms/step - loss: 4.9939e-04 - val_loss:
1.4824e-04
Epoch 45/100
45/45 ──────────────── 1s 10ms/step - loss: 4.9668e-04 - val_loss:
1.5302e-04
Epoch 46/100
45/45 ──────────────── 0s 10ms/step - loss: 4.9413e-04 - val_loss:
1.5830e-04
Epoch 47/100
45/45 ──────────────── 1s 15ms/step - loss: 4.9173e-04 - val_loss:
1.6389e-04
Epoch 48/100
45/45 ──────────────── 1s 14ms/step - loss: 4.8943e-04 - val_loss:
1.6956e-04
Epoch 49/100
45/45 ──────────────── 1s 17ms/step - loss: 4.8719e-04 - val_loss:
1.7513e-04
Epoch 50/100
45/45 ──────────────── 1s 11ms/step - loss: 4.8500e-04 - val_loss:
1.8043e-04
Epoch 51/100
45/45 ──────────────── 0s 10ms/step - loss: 4.8285e-04 - val_loss:
1.8532e-04
```

```
Epoch 52/100
45/45 ———————————————— 1s 10ms/step - loss: 4.8071e-04 - val_loss:
1.8963e-04
Epoch 53/100
45/45 ———————————————— 1s 10ms/step - loss: 4.7858e-04 - val_loss:
1.9330e-04
Epoch 54/100
45/45 ———————————————— 1s 10ms/step - loss: 4.7649e-04 - val_loss:
1.9640e-04
Epoch 55/100
45/45 ———————————————— 1s 10ms/step - loss: 4.7454e-04 - val_loss:
1.9901e-04
Epoch 56/100
45/45 ———————————————— 0s 10ms/step - loss: 4.7280e-04 - val_loss:
2.0109e-04
Epoch 57/100
45/45 ———————————————— 0s 10ms/step - loss: 4.7126e-04 - val_loss:
2.0255e-04
Epoch 58/100
45/45 ———————————————— 0s 10ms/step - loss: 4.6984e-04 - val_loss:
2.0331e-04
Epoch 59/100
45/45 ———————————————— 0s 10ms/step - loss: 4.6851e-04 - val_loss:
2.0339e-04
Epoch 60/100
45/45 ———————————————— 0s 10ms/step - loss: 4.6722e-04 - val_loss:
2.0277e-04
Epoch 61/100
45/45 ———————————————— 1s 10ms/step - loss: 4.6593e-04 - val_loss:
2.0147e-04
Epoch 62/100
45/45 ———————————————— 1s 10ms/step - loss: 4.6463e-04 - val_loss:
1.9954e-04
Epoch 63/100
45/45 ———————————————— 1s 10ms/step - loss: 4.6330e-04 - val_loss:
1.9701e-04
Epoch 64/100
45/45 ———————————————— 0s 10ms/step - loss: 4.6193e-04 - val_loss:
1.9394e-04
Epoch 65/100
45/45 ———————————————— 0s 10ms/step - loss: 4.6051e-04 - val_loss:
1.9040e-04
Epoch 66/100
45/45 ———————————————— 0s 10ms/step - loss: 4.5900e-04 - val_loss:
1.8641e-04
Epoch 67/100
45/45 ———————————————— 1s 13ms/step - loss: 4.5729e-04 - val_loss:
1.8201e-04
Epoch 68/100
```

```
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 15ms/step - loss: 4.5531e-04 - val_loss:
1.7798e-04
Epoch 69/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 4.5362e-04 - val_loss:
1.7439e-04
Epoch 70/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.5199e-04 - val_loss:
1.6753e-04
Epoch 71/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 11ms/step - loss: 4.4750e-04 - val_loss:
1.6602e-04
Epoch 72/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.4751e-04 - val_loss:
1.6309e-04
Epoch 73/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.4585e-04 - val_loss:
1.6600e-04
Epoch 74/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.4619e-04 - val_loss:
1.6485e-04
Epoch 75/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.4669e-04 - val_loss:
1.6126e-04
Epoch 76/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.4525e-04 - val_loss:
1.5434e-04
Epoch 77/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 12ms/step - loss: 4.4249e-04 - val_loss:
1.4799e-04
Epoch 78/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - loss: 4.3839e-04 - val_loss:
1.4688e-04
Epoch 79/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.3672e-04 - val_loss:
1.4686e-04
Epoch 80/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.3791e-04 - val_loss:
1.4672e-04
Epoch 81/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.3609e-04 - val_loss:
1.4471e-04
Epoch 82/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.3572e-04 - val_loss:
1.4013e-04
Epoch 83/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.3410e-04 - val_loss:
1.3710e-04
Epoch 84/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 4.3134e-04 - val_loss:
```

```
1.3348e-04
Epoch 85/100
45/45 ───────────────── 0s 10ms/step - loss: 4.2648e-04 - val_loss:
1.3114e-04
Epoch 86/100
45/45 ───────────────── 1s 13ms/step - loss: 4.2429e-04 - val_loss:
1.2711e-04
Epoch 87/100
45/45 ───────────────── 1s 14ms/step - loss: 4.2201e-04 - val_loss:
1.2727e-04
Epoch 88/100
45/45 ───────────────── 1s 17ms/step - loss: 4.2082e-04 - val_loss:
1.2690e-04
Epoch 89/100
45/45 ───────────────── 1s 17ms/step - loss: 4.1961e-04 - val_loss:
1.2594e-04
Epoch 90/100
45/45 ───────────────── 1s 16ms/step - loss: 4.1787e-04 - val_loss:
1.2511e-04
Epoch 91/100
45/45 ───────────────── 1s 11ms/step - loss: 4.1650e-04 - val_loss:
1.2386e-04
Epoch 92/100
45/45 ───────────────── 1s 10ms/step - loss: 4.1566e-04 - val_loss:
1.2315e-04
Epoch 93/100
45/45 ───────────────── 1s 10ms/step - loss: 4.1478e-04 - val_loss:
1.2366e-04
Epoch 94/100
45/45 ───────────────── 1s 12ms/step - loss: 4.1397e-04 - val_loss:
1.2334e-04
Epoch 95/100
45/45 ───────────────── 1s 10ms/step - loss: 4.1226e-04 - val_loss:
1.2233e-04
Epoch 96/100
45/45 ───────────────── 1s 10ms/step - loss: 4.0927e-04 - val_loss:
1.2339e-04
Epoch 97/100
45/45 ───────────────── 1s 10ms/step - loss: 4.0777e-04 - val_loss:
1.2098e-04
Epoch 98/100
45/45 ───────────────── 1s 10ms/step - loss: 4.0549e-04 - val_loss:
1.2073e-04
Epoch 99/100
45/45 ───────────────── 1s 11ms/step - loss: 4.0285e-04 - val_loss:
1.1997e-04
Epoch 100/100
45/45 ───────────────── 1s 10ms/step - loss: 4.0095e-04 - val_loss:
1.1966e-04
```

```
Epoch 1/100
45/45 ──────────────────── 3s 25ms/step - loss: 0.0087 - val_loss:
1.6569e-04
Epoch 2/100
45/45 ──────────────────── 1s 12ms/step - loss: 7.7059e-04 - val_loss:
1.6390e-04
Epoch 3/100
45/45 ──────────────────── 0s 10ms/step - loss: 7.2135e-04 - val_loss:
1.2274e-04
Epoch 4/100
45/45 ──────────────────── 1s 10ms/step - loss: 7.2839e-04 - val_loss:
1.1315e-04
Epoch 5/100
45/45 ──────────────────── 0s 10ms/step - loss: 7.7791e-04 - val_loss:
1.0945e-04
Epoch 6/100
45/45 ──────────────────── 1s 10ms/step - loss: 7.6457e-04 - val_loss:
1.0659e-04
Epoch 7/100
45/45 ──────────────────── 0s 10ms/step - loss: 7.1960e-04 - val_loss:
1.0190e-04
Epoch 8/100
45/45 ──────────────────── 0s 10ms/step - loss: 6.9561e-04 - val_loss:
9.9011e-05
Epoch 9/100
45/45 ──────────────────── 1s 12ms/step - loss: 6.7703e-04 - val_loss:
9.7133e-05
Epoch 10/100
45/45 ──────────────────── 1s 10ms/step - loss: 6.5918e-04 - val_loss:
9.5873e-05
Epoch 11/100
45/45 ──────────────────── 1s 10ms/step - loss: 6.4420e-04 - val_loss:
9.5227e-05
Epoch 12/100
45/45 ──────────────────── 1s 10ms/step - loss: 6.3185e-04 - val_loss:
9.5117e-05
Epoch 13/100
45/45 ──────────────────── 1s 10ms/step - loss: 6.2149e-04 - val_loss:
9.5435e-05
Epoch 14/100
45/45 ──────────────────── 1s 10ms/step - loss: 6.1261e-04 - val_loss:
9.6080e-05
Epoch 15/100
45/45 ──────────────────── 1s 12ms/step - loss: 6.0483e-04 - val_loss:
9.6962e-05
Epoch 16/100
45/45 ──────────────────── 1s 10ms/step - loss: 5.9787e-04 - val_loss:
9.8015e-05
Epoch 17/100
```

```
45/45 ——————————————— 1s 10ms/step - loss: 5.9152e-04 - val_loss:
9.9192e-05
Epoch 18/100
45/45 ——————————————— 0s 10ms/step - loss: 5.8567e-04 - val_loss:
1.0047e-04
Epoch 19/100
45/45 ——————————————— 1s 12ms/step - loss: 5.8024e-04 - val_loss:
1.0183e-04
Epoch 20/100
45/45 ——————————————— 1s 16ms/step - loss: 5.7518e-04 - val_loss:
1.0326e-04
Epoch 21/100
45/45 ——————————————— 1s 13ms/step - loss: 5.7045e-04 - val_loss:
1.0477e-04
Epoch 22/100
45/45 ——————————————— 1s 13ms/step - loss: 5.6602e-04 - val_loss:
1.0635e-04
Epoch 23/100
45/45 ——————————————— 0s 10ms/step - loss: 5.6186e-04 - val_loss:
1.0801e-04
Epoch 24/100
45/45 ——————————————— 1s 11ms/step - loss: 5.5795e-04 - val_loss:
1.0973e-04
Epoch 25/100
45/45 ——————————————— 1s 10ms/step - loss: 5.5425e-04 - val_loss:
1.1151e-04
Epoch 26/100
45/45 ——————————————— 1s 10ms/step - loss: 5.5074e-04 - val_loss:
1.1335e-04
Epoch 27/100
45/45 ——————————————— 1s 11ms/step - loss: 5.4740e-04 - val_loss:
1.1523e-04
Epoch 28/100
45/45 ——————————————— 0s 10ms/step - loss: 5.4421e-04 - val_loss:
1.1715e-04
Epoch 29/100
45/45 ——————————————— 1s 11ms/step - loss: 5.4114e-04 - val_loss:
1.1909e-04
Epoch 30/100
45/45 ——————————————— 0s 10ms/step - loss: 5.3816e-04 - val_loss:
1.2104e-04
Epoch 31/100
45/45 ——————————————— 0s 10ms/step - loss: 5.3526e-04 - val_loss:
1.2298e-04
Epoch 32/100
45/45 ——————————————— 1s 10ms/step - loss: 5.3239e-04 - val_loss:
1.2489e-04
Epoch 33/100
45/45 ——————————————— 1s 11ms/step - loss: 5.2953e-04 - val_loss:
```

```
1.2674e-04
Epoch 34/100
45/45 ──────────────── 1s 10ms/step - loss: 5.2663e-04 - val_loss:
1.2851e-04
Epoch 35/100
45/45 ──────────────── 1s 10ms/step - loss: 5.2364e-04 - val_loss:
1.3012e-04
Epoch 36/100
45/45 ──────────────── 0s 10ms/step - loss: 5.2048e-04 - val_loss:
1.3152e-04
Epoch 37/100
45/45 ──────────────── 0s 10ms/step - loss: 5.1705e-04 - val_loss:
1.3257e-04
Epoch 38/100
45/45 ──────────────── 1s 12ms/step - loss: 5.1322e-04 - val_loss:
1.3313e-04
Epoch 39/100
45/45 ──────────────── 1s 12ms/step - loss: 5.0881e-04 - val_loss:
1.3295e-04
Epoch 40/100
45/45 ──────────────── 1s 15ms/step - loss: 5.0362e-04 - val_loss:
1.3174e-04
Epoch 41/100
45/45 ──────────────── 1s 14ms/step - loss: 4.9755e-04 - val_loss:
1.2927e-04
Epoch 42/100
45/45 ──────────────── 1s 12ms/step - loss: 4.9071e-04 - val_loss:
1.2560e-04
Epoch 43/100
45/45 ──────────────── 0s 10ms/step - loss: 4.8356e-04 - val_loss:
1.2126e-04
Epoch 44/100
45/45 ──────────────── 1s 12ms/step - loss: 4.7678e-04 - val_loss:
1.1709e-04
Epoch 45/100
45/45 ──────────────── 1s 10ms/step - loss: 4.7068e-04 - val_loss:
1.1371e-04
Epoch 46/100
45/45 ──────────────── 1s 10ms/step - loss: 4.6470e-04 - val_loss:
1.1136e-04
Epoch 47/100
45/45 ──────────────── 0s 10ms/step - loss: 4.5847e-04 - val_loss:
1.1156e-04
Epoch 48/100
45/45 ──────────────── 0s 10ms/step - loss: 4.5505e-04 - val_loss:
1.1605e-04
Epoch 49/100
45/45 ──────────────── 1s 10ms/step - loss: 4.5500e-04 - val_loss:
1.2155e-04
```

```
Epoch 50/100
45/45 ———————————————— 1s 11ms/step - loss: 4.5418e-04 - val_loss:
1.2525e-04
Epoch 51/100
45/45 ———————————————— 1s 11ms/step - loss: 4.5208e-04 - val_loss:
1.2638e-04
Epoch 52/100
45/45 ———————————————— 0s 10ms/step - loss: 4.4874e-04 - val_loss:
1.2537e-04
Epoch 53/100
45/45 ———————————————— 1s 11ms/step - loss: 4.4509e-04 - val_loss:
1.2354e-04
Epoch 54/100
45/45 ———————————————— 0s 10ms/step - loss: 4.4180e-04 - val_loss:
1.2169e-04
Epoch 55/100
45/45 ———————————————— 1s 11ms/step - loss: 4.3884e-04 - val_loss:
1.2000e-04
Epoch 56/100
45/45 ———————————————— 0s 10ms/step - loss: 4.3602e-04 - val_loss:
1.1845e-04
Epoch 57/100
45/45 ———————————————— 1s 10ms/step - loss: 4.3325e-04 - val_loss:
1.1695e-04
Epoch 58/100
45/45 ———————————————— 0s 10ms/step - loss: 4.3047e-04 - val_loss:
1.1536e-04
Epoch 59/100
45/45 ———————————————— 1s 14ms/step - loss: 4.2768e-04 - val_loss:
1.1347e-04
Epoch 60/100
45/45 ———————————————— 1s 14ms/step - loss: 4.2492e-04 - val_loss:
1.1117e-04
Epoch 61/100
45/45 ———————————————— 1s 15ms/step - loss: 4.2231e-04 - val_loss:
1.0854e-04
Epoch 62/100
45/45 ———————————————— 1s 11ms/step - loss: 4.2012e-04 - val_loss:
1.0589e-04
Epoch 63/100
45/45 ———————————————— 1s 10ms/step - loss: 4.1905e-04 - val_loss:
1.0324e-04
Epoch 64/100
45/45 ———————————————— 1s 12ms/step - loss: 4.2044e-04 - val_loss:
9.9668e-05
Epoch 65/100
45/45 ———————————————— 1s 10ms/step - loss: 4.2582e-04 - val_loss:
9.4913e-05
Epoch 66/100
45/45 ———————————————— 1s 10ms/step - loss: 4.3223e-04 - val_loss:
```

```
1.0384e-04
Epoch 67/100
45/45 ──────────────── 1s 10ms/step - loss: 4.1723e-04 - val_loss:
1.0411e-04
Epoch 68/100
45/45 ──────────────── 0s 10ms/step - loss: 4.2157e-04 - val_loss:
1.0210e-04
Epoch 69/100
45/45 ──────────────── 1s 10ms/step - loss: 4.2268e-04 - val_loss:
1.0274e-04
Epoch 70/100
45/45 ──────────────── 0s 10ms/step - loss: 4.2110e-04 - val_loss:
1.0222e-04
Epoch 71/100
45/45 ──────────────── 1s 10ms/step - loss: 4.1972e-04 - val_loss:
1.0167e-04
Epoch 72/100
45/45 ──────────────── 1s 10ms/step - loss: 4.1868e-04 - val_loss:
1.0117e-04
Epoch 73/100
45/45 ──────────────── 1s 10ms/step - loss: 4.1785e-04 - val_loss:
1.0065e-04
Epoch 74/100
45/45 ──────────────── 1s 11ms/step - loss: 4.1704e-04 - val_loss:
1.0014e-04
Epoch 75/100
45/45 ──────────────── 1s 10ms/step - loss: 4.1614e-04 - val_loss:
9.9590e-05
Epoch 76/100
45/45 ──────────────── 0s 10ms/step - loss: 4.1521e-04 - val_loss:
9.9006e-05
Epoch 77/100
45/45 ──────────────── 1s 10ms/step - loss: 4.1428e-04 - val_loss:
9.8401e-05
Epoch 78/100
45/45 ──────────────── 1s 15ms/step - loss: 4.1336e-04 - val_loss:
9.7775e-05
Epoch 79/100
45/45 ──────────────── 1s 16ms/step - loss: 4.1245e-04 - val_loss:
9.7126e-05
Epoch 80/100
45/45 ──────────────── 1s 10ms/step - loss: 4.1154e-04 - val_loss:
9.6447e-05
Epoch 81/100
45/45 ──────────────── 1s 10ms/step - loss: 4.1063e-04 - val_loss:
9.5728e-05
Epoch 82/100
45/45 ──────────────── 0s 10ms/step - loss: 4.0969e-04 - val_loss:
9.4956e-05
```

```
Epoch 83/100
45/45 ———————————————— 0s 10ms/step - loss: 4.0871e-04 - val_loss:
9.4106e-05
Epoch 84/100
45/45 ———————————————— 1s 10ms/step - loss: 4.0765e-04 - val_loss:
9.3135e-05
Epoch 85/100
45/45 ———————————————— 0s 10ms/step - loss: 4.0640e-04 - val_loss:
9.1951e-05
Epoch 86/100
45/45 ———————————————— 0s 10ms/step - loss: 4.0472e-04 - val_loss:
9.0327e-05
Epoch 87/100
45/45 ———————————————— 1s 10ms/step - loss: 4.0172e-04 - val_loss:
8.6842e-05
Epoch 88/100
45/45 ———————————————— 1s 11ms/step - loss: 3.9579e-04 - val_loss:
9.1755e-05
Epoch 89/100
45/45 ———————————————— 0s 10ms/step - loss: 4.1258e-04 - val_loss:
8.8209e-05
Epoch 90/100
45/45 ———————————————— 0s 10ms/step - loss: 4.0304e-04 - val_loss:
9.3834e-05
Epoch 91/100
45/45 ———————————————— 1s 10ms/step - loss: 3.9754e-04 - val_loss:
8.4543e-05
Epoch 92/100
45/45 ———————————————— 1s 10ms/step - loss: 4.0122e-04 - val_loss:
8.8789e-05
Epoch 93/100
45/45 ———————————————— 0s 10ms/step - loss: 3.9687e-04 - val_loss:
8.6071e-05
Epoch 94/100
45/45 ———————————————— 0s 10ms/step - loss: 3.9415e-04 - val_loss:
8.2186e-05
Epoch 95/100
45/45 ———————————————— 1s 10ms/step - loss: 3.8846e-04 - val_loss:
8.4307e-05
Epoch 96/100
45/45 ———————————————— 1s 10ms/step - loss: 3.8219e-04 - val_loss:
8.0530e-05
Epoch 97/100
45/45 ———————————————— 0s 10ms/step - loss: 3.9344e-04 - val_loss:
8.4100e-05
Epoch 98/100
45/45 ———————————————— 1s 16ms/step - loss: 4.0077e-04 - val_loss:
9.0249e-05
Epoch 99/100
```

```
45/45 ───────────────── 1s 16ms/step - loss: 3.9365e-04 - val_loss:
8.2897e-05
Epoch 100/100
45/45 ───────────────── 1s 16ms/step - loss: 3.9435e-04 - val_loss:
7.8793e-05
Epoch 1/100
45/45 ───────────────── 4s 30ms/step - loss: 0.0135 - val_loss:
2.7983e-04
Epoch 2/100
45/45 ───────────────── 2s 21ms/step - loss: 0.0013 - val_loss:
1.9479e-04
Epoch 3/100
45/45 ───────────────── 1s 17ms/step - loss: 0.0011 - val_loss:
1.5198e-04
Epoch 4/100
45/45 ───────────────── 1s 17ms/step - loss: 0.0011 - val_loss:
1.5188e-04
Epoch 5/100
45/45 ───────────────── 1s 19ms/step - loss: 0.0010 - val_loss:
1.4288e-04
Epoch 6/100
45/45 ───────────────── 1s 23ms/step - loss: 9.3134e-04 - val_loss:
1.3342e-04
Epoch 7/100
45/45 ───────────────── 1s 24ms/step - loss: 8.9857e-04 - val_loss:
1.2860e-04
Epoch 8/100
45/45 ───────────────── 1s 21ms/step - loss: 8.5966e-04 - val_loss:
1.2507e-04
Epoch 9/100
45/45 ───────────────── 1s 18ms/step - loss: 8.3624e-04 - val_loss:
1.2269e-04
Epoch 10/100
45/45 ───────────────── 1s 19ms/step - loss: 8.1906e-04 - val_loss:
1.2094e-04
Epoch 11/100
45/45 ───────────────── 1s 19ms/step - loss: 8.0412e-04 - val_loss:
1.1962e-04
Epoch 12/100
45/45 ───────────────── 1s 18ms/step - loss: 7.9099e-04 - val_loss:
1.1857e-04
Epoch 13/100
45/45 ───────────────── 1s 17ms/step - loss: 7.7907e-04 - val_loss:
1.1772e-04
Epoch 14/100
45/45 ───────────────── 1s 18ms/step - loss: 7.6809e-04 - val_loss:
1.1694e-04
Epoch 15/100
45/45 ───────────────── 1s 17ms/step - loss: 7.5786e-04 - val_loss:
```

```
1.1613e-04
Epoch 16/100
45/45 ──────────────── 1s 18ms/step - loss: 7.4823e-04 - val_loss:
1.1514e-04
Epoch 17/100
45/45 ──────────────── 1s 17ms/step - loss: 7.3905e-04 - val_loss:
1.1385e-04
Epoch 18/100
45/45 ──────────────── 1s 17ms/step - loss: 7.3018e-04 - val_loss:
1.1215e-04
Epoch 19/100
45/45 ──────────────── 1s 20ms/step - loss: 7.2145e-04 - val_loss:
1.1002e-04
Epoch 20/100
45/45 ──────────────── 1s 24ms/step - loss: 7.1268e-04 - val_loss:
1.0753e-04
Epoch 21/100
45/45 ──────────────── 1s 17ms/step - loss: 7.0367e-04 - val_loss:
1.0487e-04
Epoch 22/100
45/45 ──────────────── 1s 17ms/step - loss: 6.9426e-04 - val_loss:
1.0225e-04
Epoch 23/100
45/45 ──────────────── 1s 18ms/step - loss: 6.8440e-04 - val_loss:
9.9807e-05
Epoch 24/100
45/45 ──────────────── 1s 18ms/step - loss: 6.7414e-04 - val_loss:
9.7614e-05
Epoch 25/100
45/45 ──────────────── 1s 18ms/step - loss: 6.6357e-04 - val_loss:
9.5677e-05
Epoch 26/100
45/45 ──────────────── 1s 18ms/step - loss: 6.5279e-04 - val_loss:
9.3982e-05
Epoch 27/100
45/45 ──────────────── 1s 19ms/step - loss: 6.4186e-04 - val_loss:
9.2509e-05
Epoch 28/100
45/45 ──────────────── 1s 19ms/step - loss: 6.3081e-04 - val_loss:
9.1241e-05
Epoch 29/100
45/45 ──────────────── 1s 19ms/step - loss: 6.1968e-04 - val_loss:
9.0167e-05
Epoch 30/100
45/45 ──────────────── 1s 19ms/step - loss: 6.0863e-04 - val_loss:
8.9289e-05
Epoch 31/100
45/45 ──────────────── 2s 25ms/step - loss: 5.9793e-04 - val_loss:
8.8626e-05
```

```
Epoch 32/100
45/45 ───────────────── 1s 17ms/step - loss: 5.8793e-04 - val_loss:
8.8200e-05
Epoch 33/100
45/45 ───────────────── 1s 17ms/step - loss: 5.7892e-04 - val_loss:
8.8029e-05
Epoch 34/100
45/45 ───────────────── 1s 19ms/step - loss: 5.7116e-04 - val_loss:
8.8118e-05
Epoch 35/100
45/45 ───────────────── 1s 17ms/step - loss: 5.6493e-04 - val_loss:
8.8240e-05
Epoch 36/100
45/45 ───────────────── 1s 17ms/step - loss: 5.6093e-04 - val_loss:
8.5697e-05
Epoch 37/100
45/45 ───────────────── 1s 18ms/step - loss: 5.6321e-04 - val_loss:
8.6138e-05
Epoch 38/100
45/45 ───────────────── 1s 18ms/step - loss: 5.5610e-04 - val_loss:
8.9552e-05
Epoch 39/100
45/45 ───────────────── 1s 18ms/step - loss: 5.4218e-04 - val_loss:
9.0751e-05
Epoch 40/100
45/45 ───────────────── 1s 18ms/step - loss: 5.4180e-04 - val_loss:
8.8081e-05
Epoch 41/100
45/45 ───────────────── 1s 19ms/step - loss: 5.3722e-04 - val_loss:
9.0087e-05
Epoch 42/100
45/45 ───────────────── 1s 21ms/step - loss: 5.2522e-04 - val_loss:
8.8170e-05
Epoch 43/100
45/45 ───────────────── 1s 24ms/step - loss: 5.2580e-04 - val_loss:
9.3047e-05
Epoch 44/100
45/45 ───────────────── 1s 26ms/step - loss: 5.1999e-04 - val_loss:
9.1102e-05
Epoch 45/100
45/45 ───────────────── 1s 19ms/step - loss: 5.1826e-04 - val_loss:
9.0542e-05
Epoch 46/100
45/45 ───────────────── 1s 18ms/step - loss: 5.1248e-04 - val_loss:
9.5760e-05
Epoch 47/100
45/45 ───────────────── 1s 18ms/step - loss: 5.0844e-04 - val_loss:
8.2278e-05
Epoch 48/100
```

```
45/45 ───────────────── 1s 20ms/step - loss: 5.6750e-04 - val_loss:
1.0150e-04
Epoch 49/100
45/45 ───────────────── 1s 19ms/step - loss: 5.2109e-04 - val_loss:
9.9519e-05
Epoch 50/100
45/45 ───────────────── 1s 18ms/step - loss: 5.1093e-04 - val_loss:
9.9026e-05
Epoch 51/100
45/45 ───────────────── 1s 19ms/step - loss: 5.0396e-04 - val_loss:
1.0080e-04
Epoch 52/100
45/45 ───────────────── 1s 17ms/step - loss: 4.9618e-04 - val_loss:
1.0465e-04
Epoch 53/100
45/45 ───────────────── 1s 19ms/step - loss: 4.9186e-04 - val_loss:
1.0398e-04
Epoch 54/100
45/45 ───────────────── 1s 24ms/step - loss: 4.8840e-04 - val_loss:
1.0325e-04
Epoch 55/100
45/45 ───────────────── 1s 25ms/step - loss: 4.8529e-04 - val_loss:
1.0518e-04
Epoch 56/100
45/45 ───────────────── 1s 17ms/step - loss: 4.8345e-04 - val_loss:
1.0164e-04
Epoch 57/100
45/45 ───────────────── 1s 17ms/step - loss: 4.8259e-04 - val_loss:
1.1104e-04
Epoch 58/100
45/45 ───────────────── 1s 19ms/step - loss: 4.8177e-04 - val_loss:
9.7397e-05
Epoch 59/100
45/45 ───────────────── 1s 19ms/step - loss: 5.0065e-04 - val_loss:
1.0674e-04
Epoch 60/100
45/45 ───────────────── 1s 17ms/step - loss: 4.8004e-04 - val_loss:
1.1316e-04
Epoch 61/100
45/45 ───────────────── 1s 18ms/step - loss: 4.7587e-04 - val_loss:
1.0380e-04
Epoch 62/100
45/45 ───────────────── 1s 17ms/step - loss: 4.7771e-04 - val_loss:
1.1707e-04
Epoch 63/100
45/45 ───────────────── 1s 19ms/step - loss: 4.7361e-04 - val_loss:
1.0290e-04
Epoch 64/100
45/45 ───────────────── 1s 19ms/step - loss: 4.8161e-04 - val_loss:
```

```
1.1591e-04
Epoch 65/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 4.6923e-04 - val_loss:
1.1407e-04
Epoch 66/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 4.6587e-04 - val_loss:
1.0887e-04
Epoch 67/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 23ms/step - loss: 4.7049e-04 - val_loss:
1.2115e-04
Epoch 68/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 24ms/step - loss: 4.6469e-04 - val_loss:
1.1126e-04
Epoch 69/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 4.6005e-04 - val_loss:
1.2902e-04
Epoch 70/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 4.6104e-04 - val_loss:
9.3779e-05
Epoch 71/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 5.1422e-04 - val_loss:
1.2411e-04
Epoch 72/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 4.7776e-04 - val_loss:
1.2260e-04
Epoch 73/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 4.6666e-04 - val_loss:
1.2127e-04
Epoch 74/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 4.5776e-04 - val_loss:
1.2347e-04
Epoch 75/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 4.5215e-04 - val_loss:
1.3100e-04
Epoch 76/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 4.5043e-04 - val_loss:
1.2926e-04
Epoch 77/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 4.4744e-04 - val_loss:
1.2917e-04
Epoch 78/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 23ms/step - loss: 4.4505e-04 - val_loss:
1.3921e-04
Epoch 79/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 19ms/step - loss: 4.4713e-04 - val_loss:
1.2002e-04
Epoch 80/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 4.5220e-04 - val_loss:
1.3205e-04
```

```
Epoch 81/100
45/45 ───────────────── 1s 17ms/step - loss: 4.4549e-04 - val_loss:
1.3924e-04
Epoch 82/100
45/45 ───────────────── 1s 19ms/step - loss: 4.4412e-04 - val_loss:
1.2198e-04
Epoch 83/100
45/45 ───────────────── 1s 17ms/step - loss: 4.4387e-04 - val_loss:
1.4410e-04
Epoch 84/100
45/45 ───────────────── 1s 18ms/step - loss: 4.4399e-04 - val_loss:
1.2472e-04
Epoch 85/100
45/45 ───────────────── 1s 18ms/step - loss: 4.3616e-04 - val_loss:
1.5383e-04
Epoch 86/100
45/45 ───────────────── 1s 17ms/step - loss: 4.4562e-04 - val_loss:
1.1018e-04
Epoch 87/100
45/45 ───────────────── 1s 19ms/step - loss: 4.5956e-04 - val_loss:
1.2449e-04
Epoch 88/100
45/45 ───────────────── 1s 17ms/step - loss: 4.3848e-04 - val_loss:
1.3626e-04
Epoch 89/100
45/45 ───────────────── 2s 23ms/step - loss: 4.3268e-04 - val_loss:
1.3395e-04
Epoch 90/100
45/45 ───────────────── 1s 22ms/step - loss: 4.2953e-04 - val_loss:
1.2982e-04
Epoch 91/100
45/45 ───────────────── 1s 21ms/step - loss: 4.2791e-04 - val_loss:
1.5050e-04
Epoch 92/100
45/45 ───────────────── 1s 19ms/step - loss: 4.3356e-04 - val_loss:
1.1858e-04
Epoch 93/100
45/45 ───────────────── 1s 17ms/step - loss: 4.2921e-04 - val_loss:
1.4950e-04
Epoch 94/100
45/45 ───────────────── 1s 19ms/step - loss: 4.3574e-04 - val_loss:
1.1605e-04
Epoch 95/100
45/45 ───────────────── 1s 19ms/step - loss: 4.2721e-04 - val_loss:
1.5497e-04
Epoch 96/100
45/45 ───────────────── 1s 19ms/step - loss: 4.3256e-04 - val_loss:
1.1339e-04
Epoch 97/100
```

```
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 4.3590e-04 - val_loss:
1.3254e-04
Epoch 98/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 18ms/step - loss: 4.2069e-04 - val_loss:
1.2748e-04
Epoch 99/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 17ms/step - loss: 4.1576e-04 - val_loss:
1.3237e-04
Epoch 100/100
45/45 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - loss: 4.1499e-04 - val_loss:
1.2717e-04
LSTM Test loss: 0.00011965988960582763
GRU Test loss: 7.879322947701439e-05
BiLSTM Test loss: 0.0001271696964977309
```
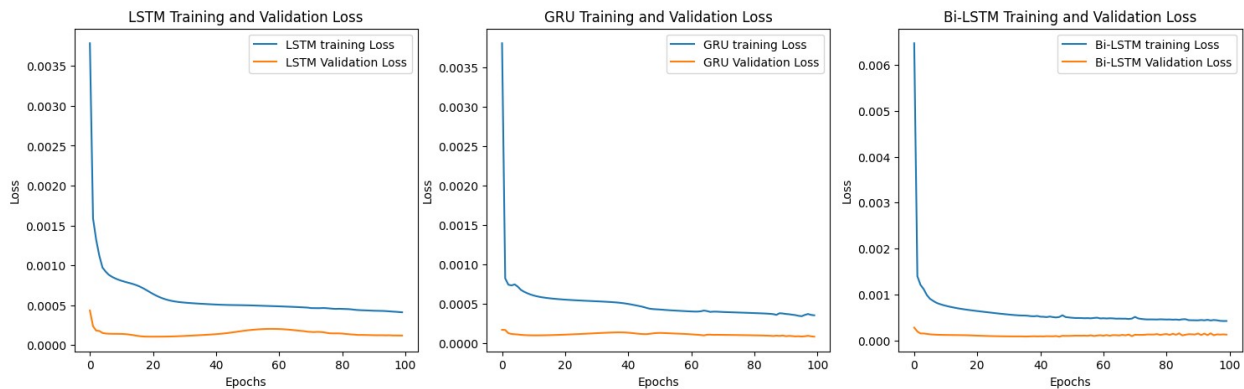


```
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step
12/12 ━━━━━━━━━━━━━━━━━━━━ 1s 36ms/step
LSTM MSE: 0.00011965987814683528, MAE: 0.00868452567123658
GRU MSE: 7.879322108244601e-05, MAE: 0.005997014748248105
BiLSTM MSE: 0.0001271696977900567, MAE: 0.00895174249978411
```
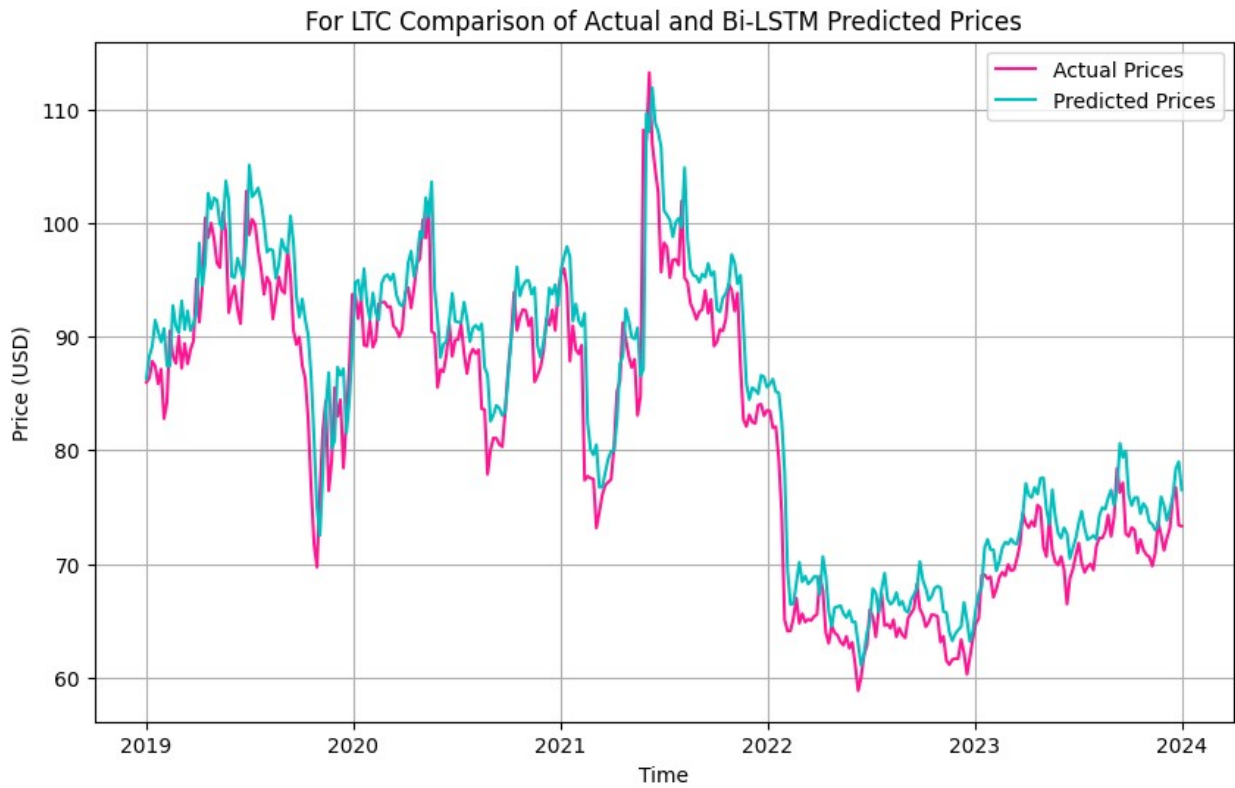
For LTC Comparison of Actual and LSTM Predicted Prices

For LTC Comparison of Actual and GRU Predicted Prices

For LTC Comparison of Actual and Bi-LSTM Predicted Prices
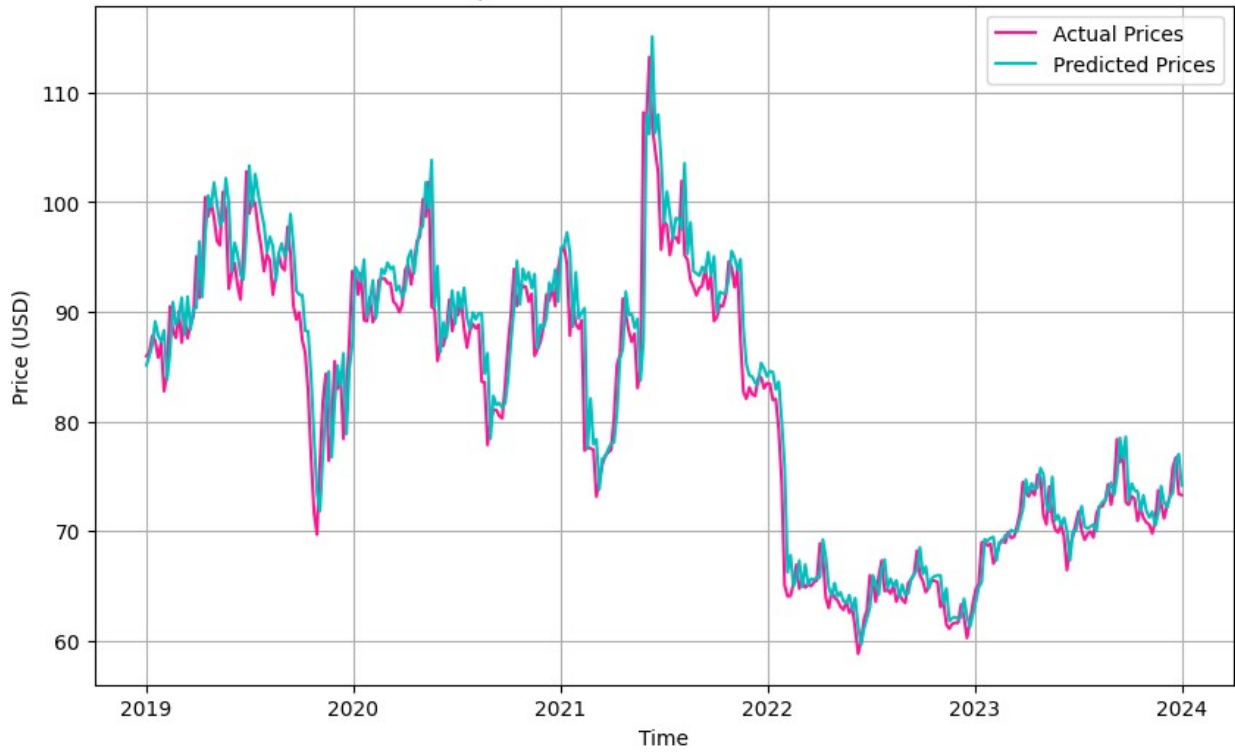
```
LSTM RMSE: 0.011, MAPE: 6.40%
GRU RMSE: 0.009, MAPE: 4.17%
BiLSTM RMSE: 0.011, MAPE: 6.54%
```