Assignment 3 Time-Series Data Report

SriAnu Vunnam

811362334

Introduction:

In this project, we analyze some deep learning models for the temperature forecasting from time-series data based on the Jena Climate dataset. These models take weather observations like temperature, humidity, pressure, and wind, and they are sampled at 10-minute intervals. We aim to create and compare a number of deep learning models and see which one performs most effectively in catching temporal patterns and forecasting future temperatures.

Data Preparation:

The dataset (jena_climate_2009_2016.csv) consists of climate measurements from 2009 to 2016.

Preprocessing Steps:

- Feature Extraction: Temperature was selected as the target variable.
- Normalization: All features were normalized using the training set mean and standard deviation.
- Train-Validation-Test Split: The dataset was split into 50% training, 25% validation, and 25% testing, ensuring temporal consistency.
- Windowing: Data was prepared in sliding sequences using timeseries_dataset_from_array with a lookback window of 120 time steps and a prediction delay equivalent to 24 hours.

Exploratory Data Analysis (EDA):

EDA revealed:

- Strong seasonal trends in temperature over the years.
- Cyclic patterns in humidity and wind speed.
- Inverse correlation between temperature and humidity, providing insights into feature interactions.

Visualizations helped confirm that the problem is well-suited for recurrent and convolutional sequence models.

Methodology:

The following models were implemented and trained:

Model 1: Dense Model (Fully Connected)

A simple network using GlobalAveragePooling1D followed by dense layers. It acts as a baseline for performance comparison.

Model 2: Conv1D Model

A 1D convolutional neural network (CNN) model to capture local patterns in time sequences using multiple convolution and pooling layers.

Model 3: LSTM Model

A unidirectional LSTM network with 16 units to capture long-term dependencies in temperature data.

Model 4: GRU Model

A GRU network trained similarly to the LSTM, aimed at faster convergence with reduced computational cost.

Model 5: Conv1D + LSTM Hybrid

A hybrid model combining Conv1D and LSTM layers to simultaneously capture short- and long-term dependencies.

Model 6: Bidirectional LSTM

Uses a bidirectional LSTM layer to capture both past and future context during training, improving temporal understanding.

Model Optimization:

Key strategies:

- Tuning sequence length, batch size, and delay window for meaningful time dependencies.
- Batch normalization and dropout layers were explored but not emphasized.
- Loss Function: Mean Squared Error (MSE).
- Metric: Mean Absolute Error (MAE).

Results and Discussion:

Each model's performance was evaluated based on Mean Absolute Error (MAE) on the test dataset. Additionally, training and validation MAEs were tracked over epochs to assess generalization.

Performance Summary (Test MAE):

Model	Test MAE
Naive Baseline	*~2.60*
Dense Model	*~2.63*
Conv1D Model	*~3.21*
LSTM Model	*~2.59*
GRU Model	~2.53
Conv1D + LSTM Model	*~2.57*
Bidirectional LSTM Model	*~2.55*

Key Findings:

- The Dense and Conv1D-only models underperformed, struggling to learn complex temporal patterns.
- GRU slightly outperformed LSTM while also being computationally more efficient.
- Conv1D + LSTM and Bidirectional LSTM offered robust performance by blending local and global context.
- GRU was the best overall performer in terms of accuracy and training time.

Conclusion:

This project demonstrated the effectiveness of RNN-based architectures for time-series forecasting. GRU model performed the best, Conv1D + LSTM and Bidirectional LSTM hybrids being very close to them. The results stress the importance of detecting short-term oscillation as well as long-term trend in time-series data.

Potential avenues for future enhancement are:

- Hyperparameter optimization (e.g., learning rate schedules, dropout, extra units)
- Application of external features such as season or time of day
- Experimenting with attention-based models or transformer models for long context learning