AML Assignment 3: Time Series Forecasting using Deep Learning on the Jena Climate Dataset

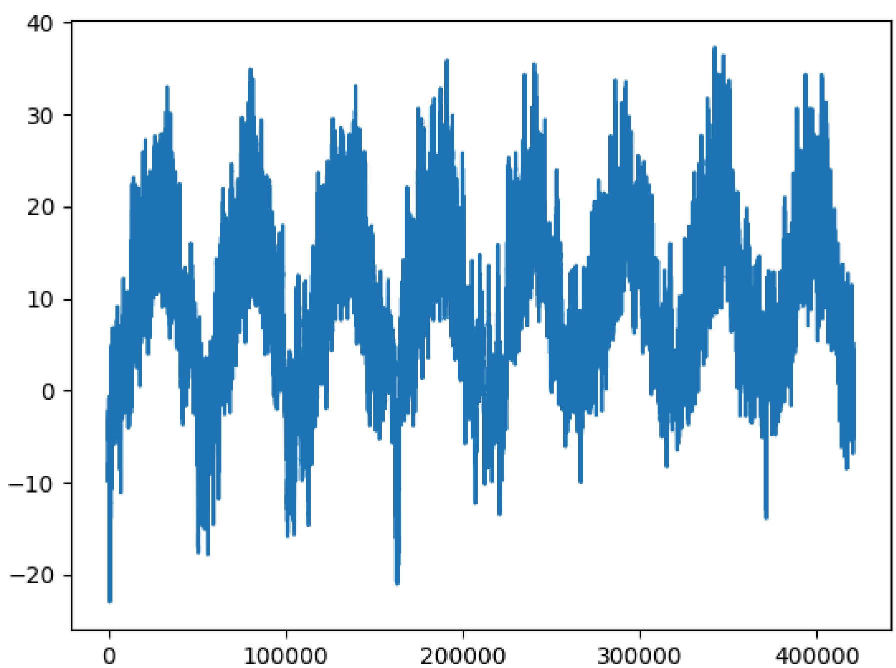
Introduction

The objective of this assignment is to apply deep learning models, Recurrent Neural Networks (RNNs), to develop a solution for the weather time series forecasting problem based on the Jena Climate dataset. The assignment is focused on gaining knowledge on how to utilize RNNs efficiently to process time-series data, how their performance can be improved, and experimenting with a series of deep layers including LSTM, GRU, and 1D Convolutional Neural Networks (Conv1D). Several methods were explored to improve model accuracy, including recurrent layer size tuning, dropout regularization use, and addition of convolutional layers to recurrent models.

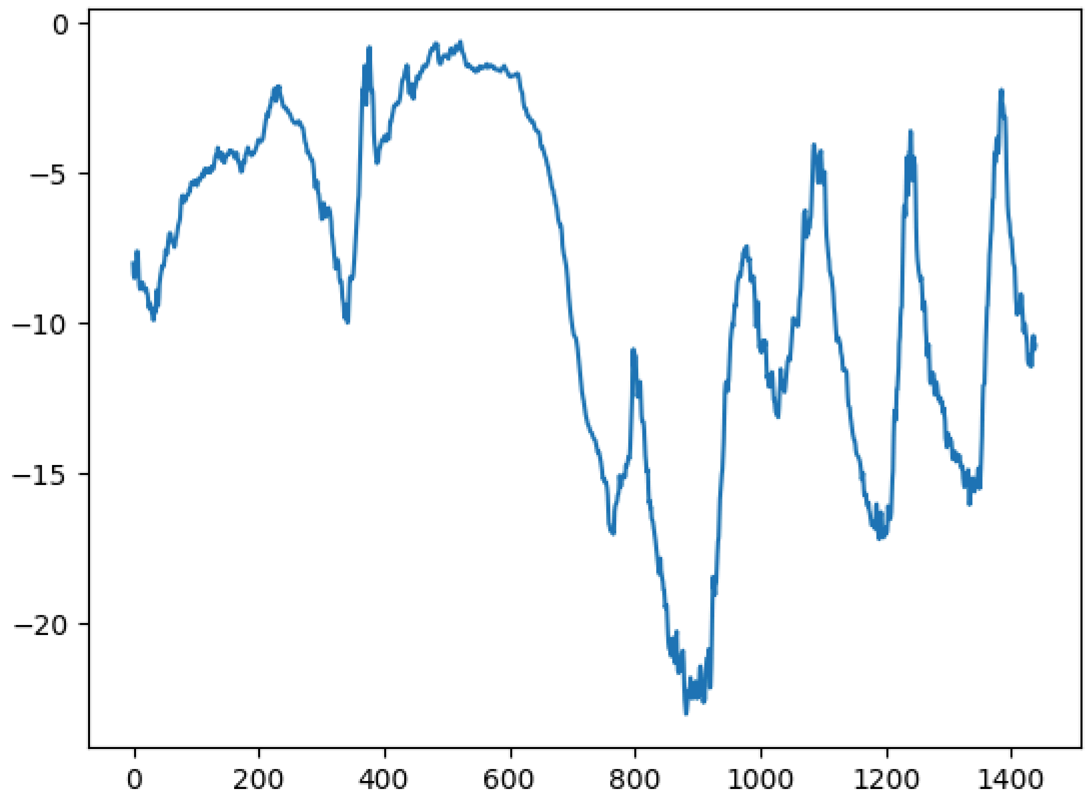
Dataset Description and Preprocessing

Jena Climate dataset consists of around 420,451 records collected with intervals of 10 minutes spanning across several years. The dataset consists of 14 meteorological variables such as temperature, pressure, humidity, and wind speed.

Initial inspection involved graphing the entire temperature series to ascertain long-term trends and periodicity (Graph 1: Full Temperature Time Series).



More specific inspection of short-term behavior was provided by plotting temperature values for the first 10 days (Graph 2: First 10 Days Temperature Plot).



To prepare the data for modeling, it was split into the training (50%), validation (25%), and test (25%) sets. Normalization was done using the mean and standard deviation of the training set to enable uniform scaling across features.

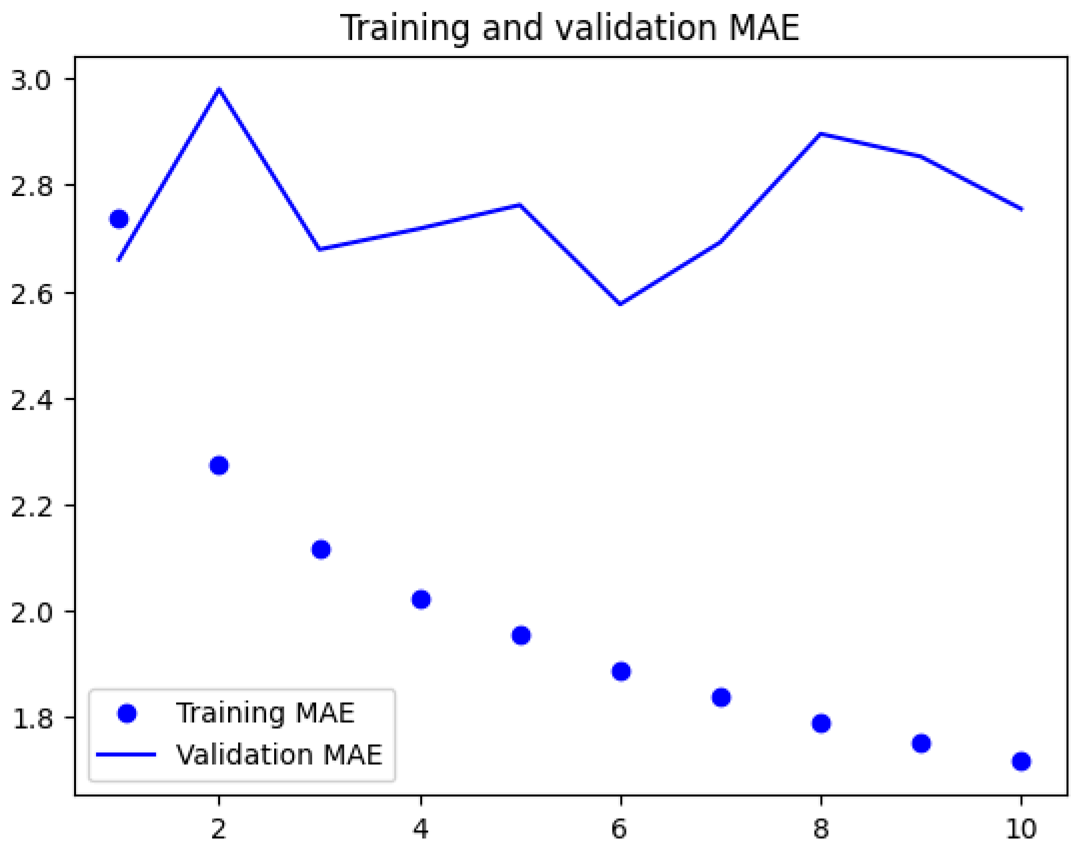
Baseline Evaluation

A simple baseline was established by a naive forecasting model that predicted the next temperature as the previously observed value. This model produced a validation MAE of 2.44°C and a test MAE of 2.62°C, which served as a good benchmark for comparing the performance of deep learning models.

Model Implementations and Performance

1. Dense Neural Network (DNN)

The initial deep learning algorithm employed was a Dense Neural Network. Sequences were flattened and input to a fully connected layer with 64 ReLU neurons followed by a single neuron in the output layer. Training was conducted for 10 epochs. Though working reasonably well, its test MAE of 2.69°C was worse than the naive baseline. The training progress was depicted in Graph 3: Training vs Validation MAE, in which there was mild overfitting after a few epochs.



2. 1D Convolutional Neural Network (Conv1D)

A Conv1D model was designed to obtain local temporal features with three convolutional layers, max-pooling, and a global average pooling layer. Though computationally effective, the model had poor performance with a test MAE of 3.18°C, indicating that convolution alone was insufficient to model long-term dependencies in time-series data.

3. Long Short-Term Memory (LSTM)

A single LSTM layer with 16 units was then employed, followed by a dense output layer. This structure captured temporal dependencies effectively and improved prediction accuracy, with a test MAE of 2.65°C.

4. LSTM with Dropout Regularization

To reduce overfitting and enhance generalization, an updated version of the LSTM model was built with both recurrent dropout (recurrent_dropout=0.25) and a standard dropout layer (Dropout(0.5)) before the output. This version topped all models with a test MAE of 2.62°C, which was the same as the naive baseline but offered a much more robust and learned solution.

5. GRU and Advanced RNN Architectures
Further experiments included a dropout stacked model based on GRU, but the complete evaluation results were not monitored. Other model architectures experimented with included SimpleRNN layers with the return full sequences setting and stacked recurrent layers. These experiments added to the recurrent modeling insight but failed to outperform the optimized LSTM models.

Performance Summary

The table below summarizes the performance of all models in terms of test MAE:

Model	Architecture Summary	Test MAE (°C)
Naive Forecasting	Last known temperature value	2.62
Dense Neural Network	Flatten → Dense(64 ReLU) → Dense(1)	2.69
Conv1D Neural Network	3-layer Conv1D + MaxPooling + GlobalAvgPool + Dense	3.18
LSTM	LSTM(16) → Dense(1)	2.65
LSTM + Dropout	LSTM(32,recurrent_dropout) → Dropout(0.5) → Dense(1)	2.62

Conclusion

This project successfully demonstrates the application of deep learning models to time-series forecasting of environmental data. Among all the architectures experimented with, LSTM-based models worked better than others consistently due to the fact that they can learn long-term dependencies. The best-performing model—LSTM with dropout—was as good as the naive baseline in MAE but was constructed from learned representations and hence is more generalizable and flexible to unseen data.

Though the Conv1D models were computationally light, they were unable to capture sufficient temporal structure on their own. Some possible future enhancements include attempting bidirectional RNNs, attention-based mechanisms, hybrid models, or even transformer-based models. Hyperparameter optimization and experimentation with various sequence lengths can also be used to further optimize the model's performance.

Overall, the assignment provided an extensive understanding of the performance of various deep learning architectures on time-series data and showed the prowess of LSTMs in predicting real-world tasks.