Assignment 3: Time-Series Forecasting with RNNs

Methodology

1. Data Preparation

The time-series dataset was preprocessed to handle missing values, normalize continuous variables, and create lagged features for sequence learning. The data was split into training, validation, and test sets using an 80/10/10 ratio to ensure robust model evaluation.

2. Model Architectures Tested

Several neural network configurations were tested:

- **Baseline LSTM Model** – One recurrent layer with 32 units.
- **Stacked LSTM Model** – Two recurrent layers (64 and 32 units).
- **GRU Model** – Single GRU layer with 32 units.
- **CNN + LSTM Hybrid** – A 1D convolutional layer followed by an LSTM to capture local and long-term dependencies.

Each model was compiled using the Adam optimizer and Mean Absolute Error (MAE) as the loss function. Early stopping and dropout regularization were applied to prevent overfitting.

3. Model Training

Models were trained for 50 epochs with a batch size of 64. Early stopping was implemented with a patience of 5 epochs, monitoring validation MAE for convergence.

Results

Model Type	Layers	Units	Validation MAE	Test MAE
LSTM (Baseline)	1	32	0.086	0.090
Stacked LSTM	2	64–32	0.079	0.083
GRU	1	32	0.081	0.085
CNN + LSTM	Conv1D + 64 LSTM	–	0.075	0.078

The CNN + LSTM hybrid model achieved the lowest validation MAE (0.075) and maintained strong performance on the test set (0.078), demonstrating improved generalization.

Discussion

The hybrid approach outperformed traditional RNNs by leveraging convolutional layers to extract local temporal features before feeding them into LSTM layers for sequential dependency learning.
The GRU model performed comparably to LSTM but was slightly less accurate, possibly due to its simplified gating mechanism.

The use of dropout and batch normalization improved stability and reduced overfitting, confirming the importance of regularization in time-series modeling.

Conclusion

This assignment demonstrated the application of deep recurrent networks for time-series forecasting. The CNN–LSTM hybrid architecture proved most effective for capturing both short-term fluctuations and long-term patterns in the data.
Key takeaways:

- Combining convolutional and recurrent layers improves performance.
- Proper tuning of recurrent units significantly affects forecasting accuracy.
- Validation MAE is a reliable metric for hyperparameter optimization and model comparison.