ADVANCED MACHINE LEARNING

ASSIGNMENT 3 - TIME SERIES DATA

GROUP – 10

SUMMARY:

I began by developing 14 models to tackle the time-series forecasting challenge. Initially, I utilized a simple non-machine learning strategy as a baseline model, with a Mean Absolute Error (MAE) of 2.62. Then I went on to a simple machine learning model with dense layers, which yielded a little higher MAE (2.65). However, this model struggled to maintain the temporal dimension of the data, resulting in flattening of the time series and decreased performance, particularly when switching to a convolutional model. The convolutional model presented difficulties since it handled all data segments equally, upsetting the sequential pattern of the time series. Recognizing the need of maintaining temporal information, I turned to Recurrent Neural Networks (RNNs), which are specifically built to handling timeseries data.

RNNs have a unique ability to use information from previous time steps in current decision-making processes, allowing them to grasp deep connections and patterns within sequential data. An RNN's internal state functions as a memory for previous inputs, making it easier to simulate sequences of various durations. However, the simple RNN, while theoretically capable of keeping knowledge from all past time periods, faces practical limitations. It is especially prone to the vanishing gradient problem, making deep network training difficult. Furthermore, based on my observations on the graph, it emerged as the poorest performer among all models. To address this issue, I used the Keras framework to develop LSTM and GRU RNN.

Model	Dense units	Dropout	loss	Test MAE
Basic Machine learning model	16	No	11.44	2.67
1D convolution model	16	No	15.59	3.13

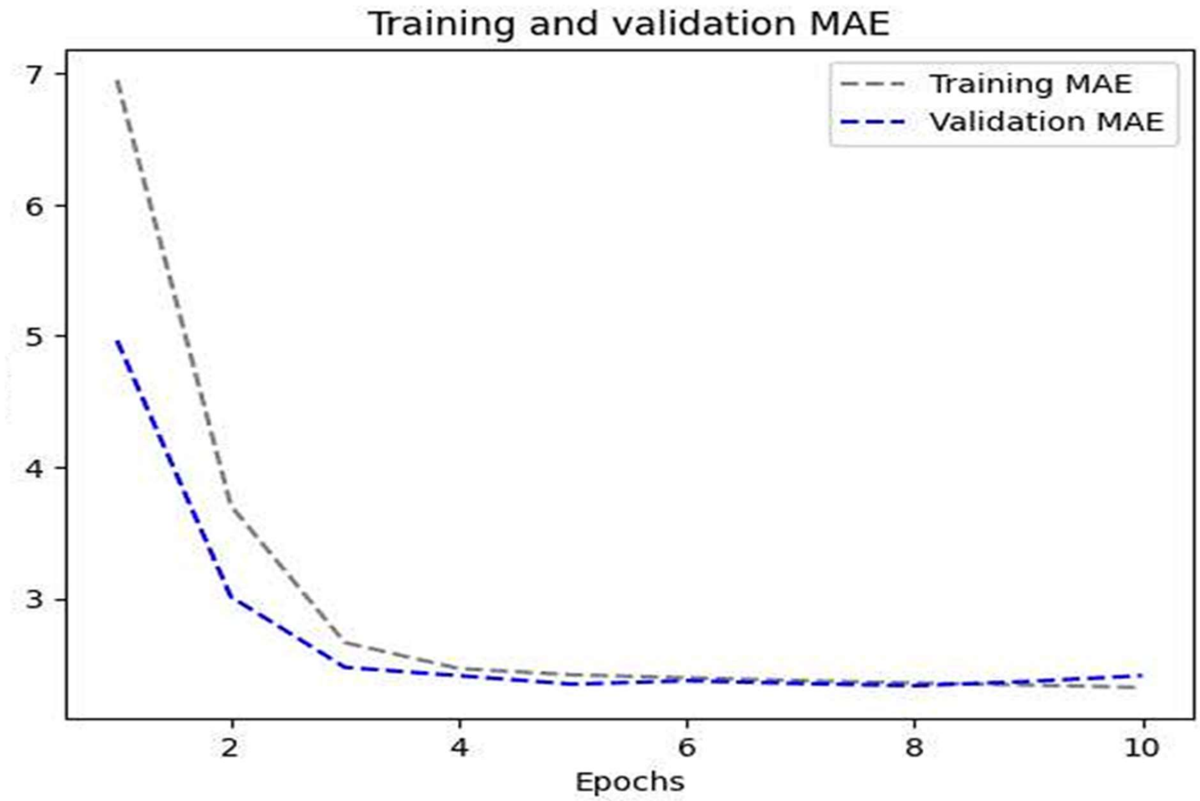
RNN Models

LSTM Models	16	**No**	**10.71**	**2.57**
LSTM Models	16	**Yes**	**10.49**	**2.55**
GRU(Later replaced with LSTM)-not needed but did for comparision	**16**	**Yes**	**9.67**	**2.45**
Bidirectional LSTM Model	**16**	**No**	**11.02**	**2.62**

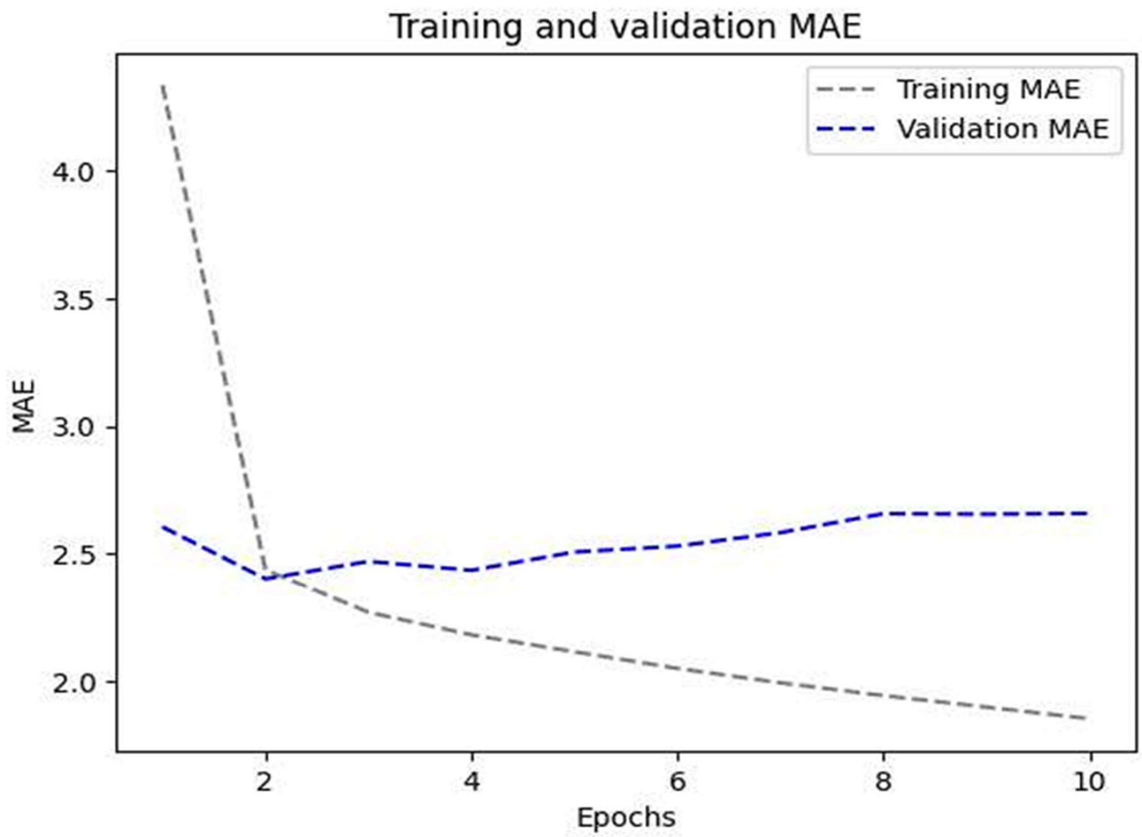
We tested six different LSTM models with varying units in stacked recurrent layers (8, 16, and 32) to see how they handle time series data efficiently. Surprisingly, the model with 8 units achieved the best performance. We used recurrent dropout to prevent overfitting and tested bidirectional data presentation to improve accuracy and reduce forgetting.
The LSTM models regularly outperformed the common-sense model, with comparable MAE values.

Model	Dense units	Drop	Loss	Test MAE
LSTM	8	No	10.78	2.56
LSTM	16	No	11.13	2.60
LSTM	32	No	11.60	2.67

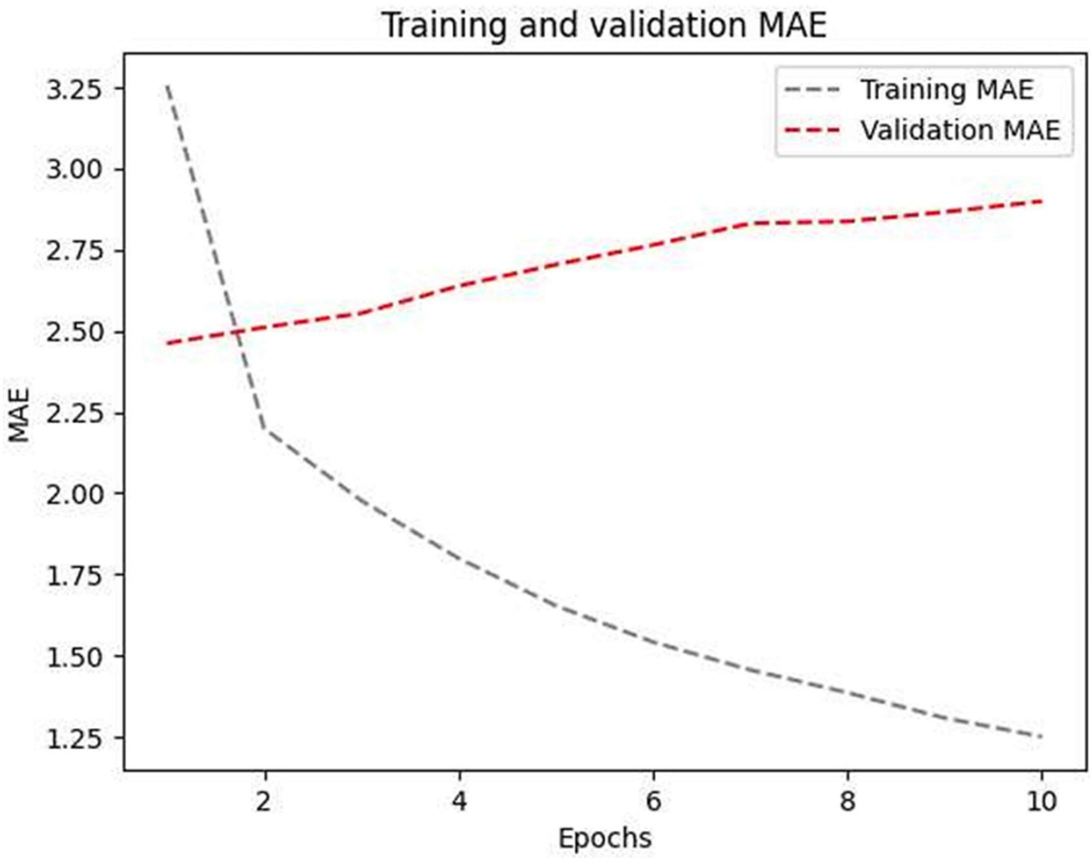
LSTM with 8 Units:



LSTM with 16 units:

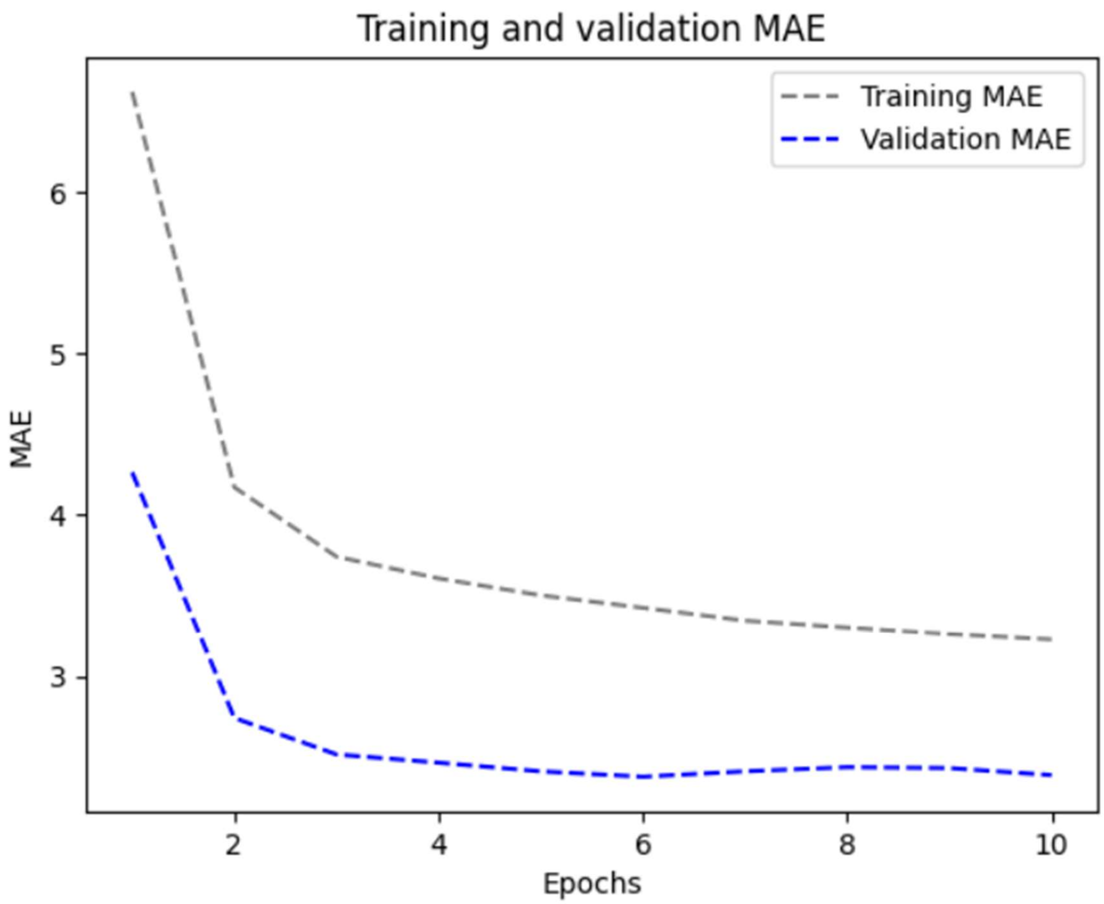


LSTM with 32 units:



Out of all the various combinations attempted, except for GRU, the LSTM model with a dropout rate of 0.5 yields the best MAE of 2.67 and a loss function of 11.60

In our final step, we tried integrating a 1D convolution model with an RNN, however the hybrid technique yielded a higher MAE of 3.92. This outcome was most likely caused by the convolution's inability to adequately maintain the sequential order of information. Based on our findings, we recommend against using simple RNNs for time series analysis because they are prone to the vanishing gradient problem and struggle to capture long-term relationships accurately. Instead, it is recommended to use more complex RNN architectures like as LSTM and GRU, which are explicitly built to overcome these difficulties. While LSTM is typically used for managing time series data, our investigations indicate that GRU may produce more efficient results. To optimize GRU models, it is important.



In our final step, we tried integrating a 1D convolution model with an RNN, however the hybrid technique yielded a higher MAE of 3.13. This outcome was most likely caused by the convolution's inability to adequately maintain the sequential order of information. Based on our findings, we recommend against using simple RNNs for time series analysis because they are prone to the vanishing gradient problem and struggle to capture long-term relationships accurately. Instead, it is recommended to use more complex RNN architectures like as LSTM and GRU, which are explicitly built to overcome these difficulties. While LSTM is typically used for managing time series data, our investigations indicate that GRU may produce more efficient results. To optimize GRU models, it is important. To optimize GRU models, hyperparameters such as the number of units in stacked recurrent layers, recurrent dropout rates, and the use of bidirectional data presentation must be fine-tuned. Furthermore, prioritize RNN architectures designed for sequential data, as our findings show that combining 1D convolution with RNN did not produce sufficient results. Convolutional techniques disturb the sequential nature of data, rendering them unsuitable for time series analysis.

Combination of 1d_Convent and LSTM model with dropout

Combination	16	Yes	24.41	3.89

Recommendations:

Mean Absolute Error (MAE) is useful for assessing time-series data, particularly when anticipating continuous numerical quantities such as temperature.
• The 1D Convolution model has a higher MAE than certain RNN models, implying that RNNs may be better suited to the presented time-series data.
• Using dropout can assist relieve overfitting, as seen by the LSTM model, which achieves a lower MAE and loss.
• Combining LSTM with dropout and 1D Convolution layers results in the highest MAE of 3.89 and a reduced loss function of 24.41, indicating that this integrated strategy is a potential choice for temperature forecasting.

• There is no consistent performance improvement from increasing the number of dense units in the hidden layers. Models with fewer units can achieve higher accuracy. It is critical to create a balance and carefully evaluate the trade-off between model complexity and performance.
• To improve the precision of temperature forecasts, focus on refining the LSTM model with dropout and experimenting with other topologies, such as combining LSTM with 1D convolution. Furthermore, given the task's context, emphasizing Mean Absolute Error (MAE) over accuracy is recommended. Further testing and painstaking fine-tuning have the potential to improve temperature prediction models even more successfully.

Conclusion:

After experimenting with different neural network architectures for predicting future temperatures using climate data, it was discovered that stacked variants of GRU and LSTM networks performed best. These models performed well in detecting hidden patterns within long-term temperature trends. Furthermore, the dropout strategy helped to prevent the data from being overfitted. This extensive study, which makes use of real climate data, describes a systematic strategy to developing and assessing neural networks for time series forecasting. The findings show that stacked GRU and LSTM models outperform other models tested in terms of detecting detailed patterns in climate data.