# Assignment 3 – Time Series

**Objective -** The objective is to explore the application of Recurrent Neural Networks (RNNs) to time-series data, investigate strategies for improving the performance of RNN-based models when dealing with time-series data, and examine the application of various deep learning layers to enhance the analysis of time-series data.

**Model Building** – In total we have built total of 15 models, we have divided the data in training validation and testing sets of 50, 25 and 25 percent, we are currently building and testing a densely connected neural network model for time series forecasting. The Jena temperature dataset is utilized to train the model, and the MAE is employed as the evaluation measure.

The model has a single input layer with 14 features (the number of features in the Jena temperature dataset), a single hidden layer with 16 units, and a single output layer with 1 unit (the target variable, which is the temperature).

The validation MAE is reported as 2.44, and the test MAE is reported as 2.62. The MAE is a measure of the average absolute difference between predicted and actual values. These results provide a baseline performance that any subsequent model should aim to surpass. If a more sophisticated model achieves a lower MAE on the validation and test sets, it indicates an improvement over this simple baseline.

In this case, it is likely that the dense layer model is overfitting the training data. This is because the dense layer model has a relatively large number of parameters (16 units in the hidden layer) and a relatively small amount of training data (819 samples).

In order to reduce the problem of Overfitting we need to proceed with regularization techniques, such as dropout or L1/L2 regularization.

The **1D Convolutional Neural Network** (CNN) model, after 10 epochs, yielded a test Mean Absolute Error (MAE) of 3.04. This result is marginally higher than the test MAE obtained by the densely connected neural network model, which recorded a value of 2.59. One difference could be attributed to the use of pooling layers in CNNs. Although these layers are designed to reduce data dimensionality and capture spatial features, they may inadvertently eliminate long-term dependencies present in time series data. Consequently, the densely connected neural network, without such pooling layers, might have been more effective in preserving these temporal relationships, contributing to its lower test MAE.

RNNs are a suitable choice for time series data because they are specifically designed to handle sequences and can capture the temporal relationships within the data. **Simple RNN** model achieved a test MAE of 9.91. This is significantly higher than the test MAE of the densely connected neural network model (2.59) and the CNN model (3.04). training results indicate that the model's validation MAE decreases slightly over the epochs, but the final test MAE is around 9.91 suggesting that the model might not be performing well on the task. This is the least performing model among the 15 due to their vanishing gradient" problem, which makes them less effective for many sequential data tasks compared to more advanced RNN variants like LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit).

While the inclusion of additional layers in a neural network has the potential to enhance its capacity to include intricate patterns, the introduction of stacked Simple RNN layers hasn't notably improved performance. The mean absolute error (MAE) on the test dataset persists at approximately 9.48, indicating that the model struggles to effectively capture the underlying data patterns.

In comparison, the Simple GRU model represents a fundamental recurrent neural network (RNN) architecture featuring a single GRU layer. The LSTM-Simple model, on the other hand, adopts a more intricate RNN structure, utilizing a single LSTM layer. Furthermore, the LSTM (RNN) - dropout model introduces regularization through dropout, aiming to mitigate overfitting. Despite these variations, the test dataset's relatively high MAE across these models suggests ongoing challenges in uncovering and learning the intrinsic patterns within the data.

All three models achieve similar test MAE scores, but the LSTM-Simple model takes slightly longer to train. The LSTM (RNN) - dropout model takes even longer to train, but it achieves the best test MAE score.

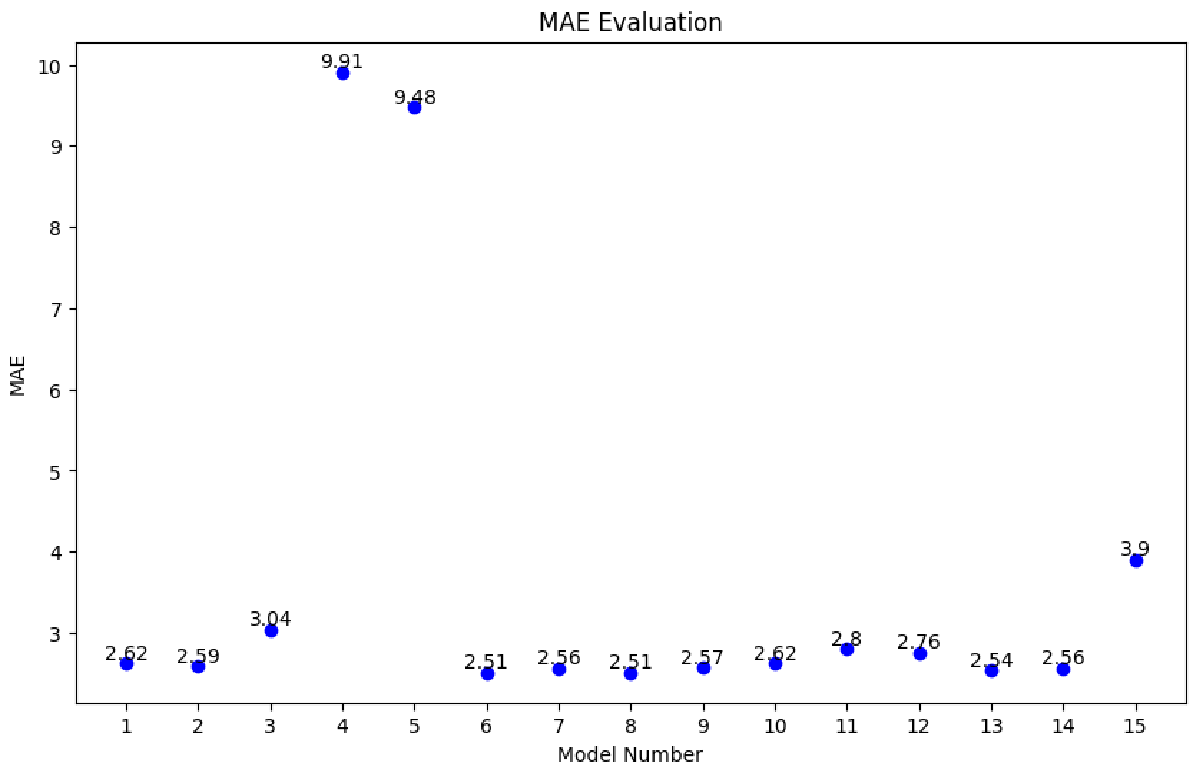
In our evaluation of model performance based on the number of units in stacked LSTM layers, it was observed that models with higher unit counts (32 and 64) exhibited greater Mean Absolute Errors (MAEs) compared to models with fewer units (8 and 16). This suggests a relative sensitivity to the number of units in the LSTM layers.

Bidirectional RNNs were employed to capture dependencies in both the forward and backward directions of the input sequence. These models consistently demonstrated lower MAE values when compared to a basic baseline model, highlighting their effectiveness in time series forecasting.

However, the integration of a convolutional model with an RNN in a hybrid approach resulted in unsatisfactory performance, as indicated by a high MAE. This suboptimal outcome was attributed to the inherent limitations of convolutional methods, which can disrupt the sequential order of information within the data. This underscores the importance of selecting architectures specifically tailored for handling sequential data, such as RNNs.

Another model, combining 1D convolutional layers with an LSTM layer, was designed to capture both local patterns and long-term dependencies in the input sequence. Despite this hybrid approach, the test MAE of 3.90 was higher compared to the bidirectional LSTM model. This discrepancy could be attributed to the increased complexity introduced by the combination of convolutional layers and LSTM.

In the context of time series forecasting, LSTM and GRU models outperformed simple RNN models due to their enhanced ability to capture long-term dependencies. The unsuccessful performance of the hybrid model, integrating a convolutional model with an RNN, is likely attributable to the disruptive impact of convolutional approaches on the sequential order of information in time series data.


MAE Evaluation
MAE
Model Number
9.91
9.48
3.04
2.62
2.59
2.51
2.56
2.51
2.57
2.62
2.8
2.76
2.54
2.56
3.9