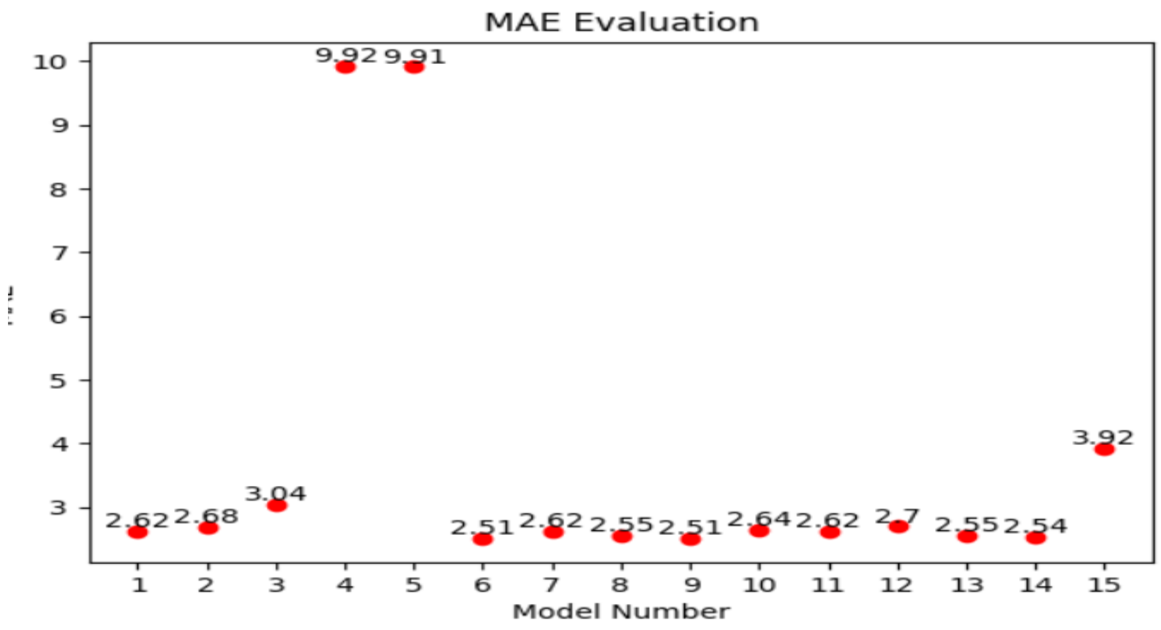
Assignment 3
Report

Objective :
The objective is to apply RNN to time series data and to improve the performance of network when dealing with time series data . Evaluating and comparing various machine learning models built with different strategies to improve the performance of the model in handling time series data, with the aim of identifying the most effective model .



Models :
I have created 15 models one of which is common sense base line method which gave Mean Absolute Error (MAE) of 2.62, subsequently a basic ML model with dense layer which gave slightly higher MAE of 2.68.

The dense layer didn't work well because it made the time series flat, and this caused us to lose the sense of time in our data. I also experimented with a convolution model, but it didn't perform well. The convolutional approach treated all parts of the data the same way, even when we tried pooling, which messed up the order of information. This is why we turned to a specific architecture designed for time series data called RNN (Recurrent Neural Networks).

An important feature of RNNs is their ability to remember information from earlier steps when making decisions. This helps the network understand connections and patterns in sequences of data. The RNN's internal state acts like a memory of what it has seen before, allowing it to work with sequences of different lengths.Tthe fundamental version of RNN, referred to as

SimpleRNN, is generally too simplistic to be of practical value. In fact, as evident from our graph, SimpleRNN consistently ranks as the poorest performer among all the models we examined.

LSTMs (Long Short-Term Memory networks) are widely recognized as a top choice for handling time series data. During our experiments, we explored different LSTM models by adjusting the number of units in the stacked recurrent layers, trying values of 8, 16, 32 and 64. Out of these options, using 8 units produced the best results, surpassing the performance of the models with 16 ,32 and 64 units.

Recurrent dropout has been introduced to prevent overfitting. We experimented with bidirectional data, which presents the same information to an RNN in two different ways

LSTM models have been evaluated and found that they all exhibited very similar MAE values, and notably, these MAE values were consistently lower than those of the common-sense baseline model. This trend was also confirmed through our MAE evaluation graph.

I attempted to create a hybrid model that combined a convolutional model with an RNN. this hybrid model yielded unsatisfactory results, with a high MAE of 3.92. The likely reason behind this poor performance was the inherent limitation of the convolutional approach, which disrupted the sequential order of information within our data.

it's clear that simple RNNs encounter a challenge known as the vanishing gradient problem, which hampers their effectiveness in capturing long-term relationships in data. LSTM and GRU, as they are specifically designed to address and mitigate these issues.

I believe LSTM stands out as a popular choice for handling time series data due to its strong capability in capturing long-term dependencies.our experiments have indicated that GRU might offer a more efficient alternative. To get the most out of GRU, I recommend fine-tuning its hyperparameters, such as adjusting the number of units in the stacked recurrent layers, optimizing the recurrent dropout rate, and considering the use of bidirectional data.

I think its is good to look into architectures that were specifically designed to handle such sequential data, one of such is RNN.