**REPORT**

**Assignment-3 Time-Series Data**

Group -24

Shiva Chaitanya Goud Gadila (811252042)

Praneeth Simha Mallenahalli Vijaya (811252718)

**Introduction and Benchmark Model Evaluation**: In the process of analyzing our time series data we have created 14 different models. Using simple techniques, the first model served as a benchmark with a Mean Absolute Error of 2.44. After that we built a simple machine learning model with a dense layer, which produced an MAE of 2.62 which is somewhat higher. This thick layer model flattened the time series data since it was unable to capture the temporal context which led to insufficient performance. We also tested a convolutional model which produced unsatisfactory results since it handled all the data segments equally even after pooling which interfered with the data’s natural sequential order.

**Exploring Machine Learning Models:** From this point on we could see that Recurrent Neural Network provides a better method for working with time series data. The capacity of RNN’s to incorporate knowledge from earlier phases into the decision-making process at hand is one of its primary characteristics. The network can now identify relationships and patterns in sequential data because of this functionality. An RNN can effectively represent sequences of varying lengths because its internal state acts as a memory, retaining information from previous inputs.

**Limitations of Basic Simple RNNs and Introduction to Advanced Variants:** But the basic simple RNN model frequency turns out to be too basic to be useful. Additionally, the simple RNN has a notable drawback as graphical representations show it consistently performs the worst out of all the models. In theory, simple RNN can store data from all previous time steps, but in practice especially in deep networks it often fails because of the well-known vanishing gradient problem. This problem makes it almost hard for the network to train efficiently. More advanced RNN variations including the Gated Recurrent unit and long short-term memory have been created in response to this difficulty and are easily incorporated into frameworks like keras.

**Success with Basic GRU model and Computational Efficiency:** Of all the models we explored, the basic GRU model produced the best results during our experimentation. Its capacity to accurately identify long range connections within sequential data was the main factor in its success. In our study, the GRU model also proved to be more computationally efficient than long short-term memories which increased attractiveness even more.

**Experimentation with Long Short-Term Memory:** Six different long short-term memories models were used in our trials. LSTMs are well known for their ability to manage time series data effectively. These models had different units in stacked recurrent layers and additionally the model with 8 units performed the best. Recurrent dropout techniques were also used to reduce overfitting. To improve accuracy and address the issue of forgetting within the framework of our study, we also investigated bidirectional data display. The MAE values of these long short-term memory models were all comparable and consistently less than those of the commonsense model.

**Challenges with Hybrid RNN and Convolutional Model:** The last stage of the analysis that is conclusion where we attempted to combine an RNN and a 1D convolutional model but unfortunately, the hybrid model produced a greater MAE of 2.67 which can probably be attributed to the fundamental limits of the convolution in terms of maintaining the information’s sequential sequence.

**Findings And Recommendations:** The findings suggest that RNNs should be avoided in time series analysis because they are vulnerable to the vanishing gradient problem and cannot sufficiently capture long-term dependencies. Rather using more advanced RNN architectures such as LSTM and GRU is advised because they are designed to precisely handle these issues. While LSTM is commonly used to handle time series data our findings suggest that GRU may produce more effective results.

**Optimizing Hyperparameters and Recommendations of RNN Design:** Hyperparameters such as the number of units in stacked recurrent layers, recurrent dropout rates and the use of bidirectional data presentation must be adjusted to improve the performance of GRU models. Therefore, it is recommended to give priority to RNN designs that are specifically customized for sequential data, since our attempts to combine 1D convolution with RNN did not produce acceptable outcomes. Convolutional algorithms are less suitable for evaluating time series data since they frequently distort the information’s sequential order.

The scatter plot shows the Mean Absolute Error (MAE) evaluation for different models. Each model is labeled on the x-axis, while the MAE values are shown on the y-axis. The red dots represent the MAE value for each model. Lower MAE values mean better prediction accuracy. We can compare model performance using these MAE values to pick the best model for our needs.

