DeepAR:

The DeepAR algorithm is a supervised learning algorithm. It is a forecasting time series algorithm based on recurrent neural networks (RNN). Some conventional models like Autoregression (AR), Moving Average (MA), Autoregressive Integrated Moving Average (ARIMA), and Simple Exponential Smoothing (SES) learn from past observations and predict future values using recent history. Whereas, instead of fitting separate models for each time series, DeepAR aims to create a global model that learns using all-time series in the dataset

DeepAR outperforms the standard ARIMA and ETS methods when the data has multiple time series. DeepAR can forecast the data of multiple time series by only single model training [1].

Methodology:

The training input for the DeepAR algorithm is one or more target time series data that is

generated by the same or similar processes. The input data is given as a JSON file or in Parquet format. Input data should contain the start and target fields. Start field is a string with the format YYYY-MM-DD HH:MM: SS. Target field is an array of floating-point values or integers that represent the time series.

Prepared data should be trained with the DeepAR model. The below parameters are required to define the model.

Required Parameters [4]:

* Context\_length: hyperparameter controls how far in the past the network can see
* Prediction\_length: hyperparameter controls how far in the future predictions can be made.
* Epochs: The maximum number of passes over the training data.
* Time\_freq: The granularity of the time series in the dataset.

Based on this input, DeepAR trains a model and uses it forecast the target time series data.

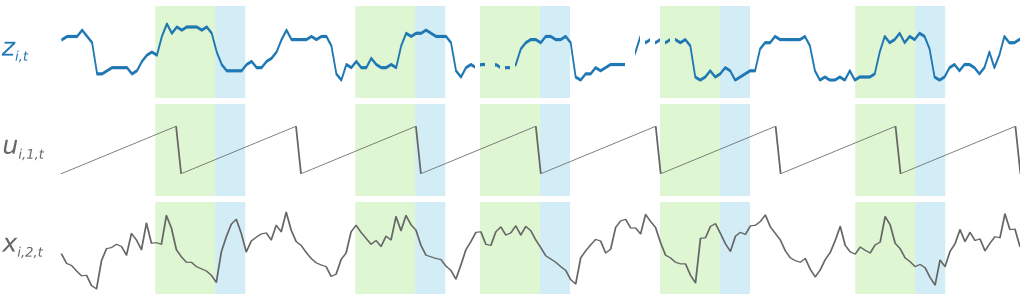


Figure Input-Output

Figure 1 depicts the general overview of how the output of the DeepAR algorithm is lookalike. Here, Z is the target time series, U and X are the features.

Flowchart:

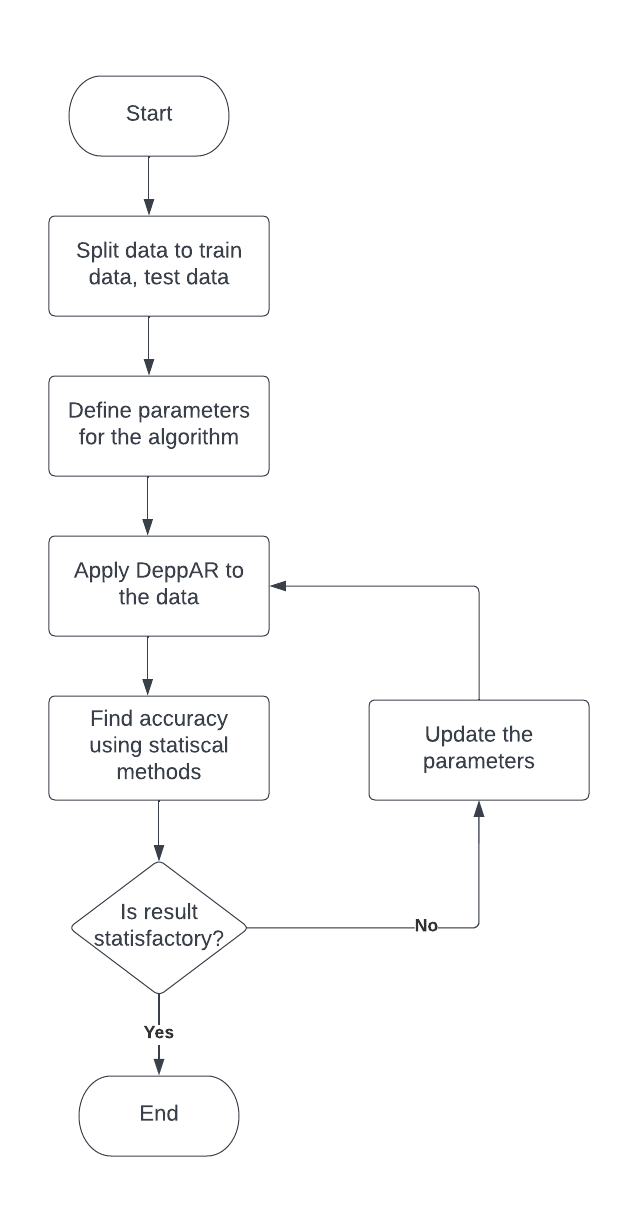


Figure Flowchart

Algorithm on our research work:

After dividing the data into train and test parts, we applied the DeepAR algorithm using the Dart package to our dataset. The forecast visualization of the test data is given below.

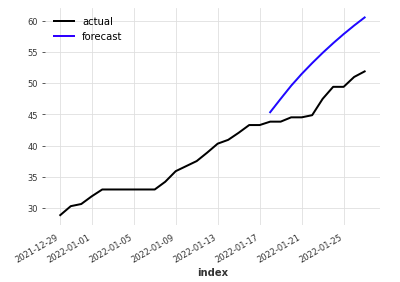


Figure Result

Here, we get the MAPE= 0.32%, which is higher than the LSTM algorithm. The data in our case contains only a single time-series field. However DeepAR is efficient, it is more suitable and robust for multiple time-series forecasting and chunky data. As the error rate of DeepAR is higher, our data is straightforward (contains only one time series with uniform features), so we choose the LSTM over the DeepAR.

Pros and cons:

DeepAR has the advantage of training several hundred or thousands of time-series simultaneously, potentially offering significant model scalability. It also has the following technical benefits:

* Minimal Feature Engineering: The model requires minimal feature engineering, as it learns seasonal behaviour on given covariates across time series.
* Monte Carlo Sampling: It is also possible to compute consistent quantile estimates for the sub-ranges of the function, as DeepAR implements Monte Carlo sampling. This could, for instance, be useful when deciding on safety stock.
* Built-in item supersession: It can predict on items with little history items by learning from similar items
* Variety of likelihood functions: DeepAR does not assume Gaussian noise, and likelihood functions can be adapted to the statistical properties of the data allowing for data flexibility [2].

Reference:

1. <https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html#deepar-inputoutput>
2. <https://towardsdatascience.com/prophet-vs-deepar-forecasting-food-demand-2fdebfb8d282>
3. <https://docs.aws.amazon.com/sagemaker/latest/dg/deepar_how-it-works.html>
4. <https://docs.aws.amazon.com/sagemaker/latest/dg/deepar_hyperparameters.html>