IISER PUNE

Semester Project Report

Time series forecasting for climate data using deep neural network models

Name: Barish Sarkhel
Reg no: 20201242
Semester : January 2024 (8th Semester)
Course code:DS 4613
Supervisor: Dr Joy Merwin Monteiro

1 Objective:

The main objective of our project is to implement different type of neural network based time series forecasting model and compare their performance. Along with this we tried to understand what are the regions where one model is performing then the other. I also tried to build a model which will be an ensemble model which should perform better then each individual models. Also we tried to compare it with some other simple statistical machine learning models.

2 Introduction:

This main aim of the project was to understand how deep learning models are used in time series analysis. In my last semester project I understand different type of error matrices and build some machine learning time series models like liner regression model, exponential smoothing model. I tried to understand and analyse the deep learning models using those error metrics.

In this project I took three neural network based time series models to do our analysis and prediction. These models are Artificial Neural Network (ANN), Long Short-Term Memory (LSTM) and Neural Basis Expansion Analysis for Time Series (N-BEATS). I mainly used mse and mae error metrics to analyse these model. But there are other error metrics which I also used for reference. Along with this I computed some probability distributions to understand the performance of the models in different region of the dataset.

3 Data set:

Here we used the temperature data for our model. The data is taken from a point near to Pune. The data was given in each three hour time steps. I averaged them to get the daily averaged temperature value. For the missing value I did linear interpolation. Then I normalized the dataset using min max scaling to convert it in the range of 0 to 1. The formula for this is the following-

$$\text{Min Max Scaling} = \frac{\text{Actual value} - \text{Minimum value}}{\text{Maximum value} - \text{Minimum value}}$$

4 Deep Neural Networks Models for Time Series:

Here I described about different deep neural network models and techniques to use them in time series prediction.

4.1 Artificial Neural Network (ANN)-

Artificial Neural Network (ANN) is a computational model inspired by the human brain's neural networks. ANNs models do forecasting of future values based on past observations in time series data. We usually give some N no of previous time steps value to compute the N+1 th time step value. ANNs consist of interconnected nodes (neurons) organized in layers, including an input layer, one or more hidden layers, and an output layer. As I said the N previous time steps value given as a N size vector and we put the hidden layers along with the appropriate activation functions which brings non linearity in the model. After the training when we get appropriate weights and biases we can use this model to predict the future model. We need to try with different number of hidden layers, different number of neurones in it along with different type of activation functions to get the best model for our dataset.

4.2 Long Short-Term Memory (LSTM)-

The ANN model has the problem of not capturing the sequence in the data and the long term dependency in the dataset. Long Short-Term Memory or LSTM model is a type of recurrent neural network which is used to model any sequential data with long term dependency. LSTMs consist of memory cells and gating mechanisms that enable them to selectively remember or forget information over long sequences which allows them to capture long-term dependencies in time series data. This models have gates which helps in remembering and forgetting informations. There are three type of gates which are input gates, forget gates, and output gates, which control the flow of information in the model. We give the input data to the model in the same way as we gave in ANN model. The model understand the temporal patterns and dependencies in the data and learn the important parameters which then used for prediction.

4.3 Neural Basis Expansion Analysis for Time Series (N-BEATS)-

N-BEATS (Neural Basis Expansion Analysis for Time Series) is a deep learning model designed for time series forecasting tasks.The N-BEATS model is based on the concept of neural basis expansion, which involves decomposing time series data into basis functions. These basis functions capture different patterns and trends present in the time series data. N-BEATS then employs a deep learning architecture to learn the relationships between these basis functions and make accurate predictions.

The key components of the N-BEATS model include:

1. Stacks: N-BEATS consists of multiple stacks, each containing several fully connected layers. These stacks can be either generic (for capturing general patterns) or trend-specific (for capturing long-term trends).

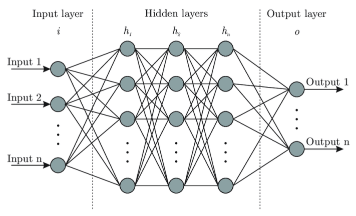
2. Blocks: Each stack is composed of multiple blocks, with each block containing a fully connected feedforward neural network. The blocks are responsible for processing the input data and generating forecasts.

3. Forecasting Horizons: N-BEATS can predict time series data for multiple forecasting horizons, allowing it to make short-term and long-term predictions simultaneously.

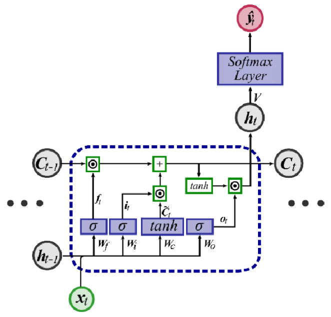
4. Backcast and Forecast: N-BEATS uses a two-step process called backcast and forecast. In the backcast step, the model predicts the future values of the time series based on historical data. In the forecast step, the model refines its predictions using additional information, such as exogenous variables or future observations.

N-BEATS has demonstrated strong performance in time series forecasting tasks then the existing models.

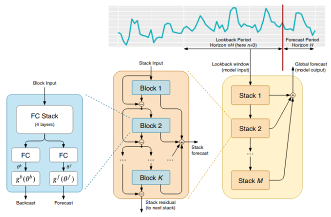
Here in Figure 1 I have attached a basic architecture of the all the three models.



(a) ANN model architecture



(b) LSTM model architecture



(c) N-BEATS model architecture

Figure 1: Basic architecture of all the models

5 Ensemble Models:

Our next target was to make some model which will ensemble all these models. I used some ensemble techniques to ensemble these deep neural language models. I described those techniques here.

5.1 Ensemble trough average model-

This is an ensemble technique where we use simple average method to ensemble the predictions. In this case we take the prediction of each of the model and average at each time steps. We take this as the final prediction and analyse the result.

5.2 Ensemble through model confidence-

This is an ensemble technique where we try to do weighted average of the prediction of different model to build a more accuarte model. Here we use the confidence of the model to calculate the weights for each models. The techniques we used is the following:

a. We compute the variance of predictions for each model on the validation set. Variance represents the spread or uncertainty of predictions.

b. We assign weights inversely proportional to the variance, meaning models with lower variance (higher confidence) will have higher weights.

c. We normalize the weights to ensure they sum up to 1, making them suitable for weighting predictions in an ensemble.

5.3 Ensemble through meta learning-

Here we try to ensemble the model using the technique of meta learning. We followed these following steps these ensemble model:

a. We took the prediction of the validation set and put then in stack. So now we got a dimension vector for each time step in the validation set (one point coming from each of the three deep learning model. We will use this three dimension vector as input of the meta model.

b. Now we build our meta model which is basically a simple ANN model taking 3 inputs and the output try to predict the actual value. We trained the weights and biases of these model.

c. Now we take the test set and converted them in a three dimension vector as we did for the validation set and put it in the meta model to get the prediction for the test set.

6 Chaotic time series:

We also build a chaotic time series model to understand the model. The daily averaged temperature value dataset was basically a sine curve with some deviance. We build this model to understand about the performance of these models when the dataset has more complex pattern then the temperature data.

To make the chaotic time series model we used the Lorenz system with the usual parameter $\sigma = 10$, $\beta = \frac{8}{3}$ and $\rho = 28$. The equations are:

$\frac{dx}{dt} = \sigma(y - x)$

$\frac{dy}{dt} = x(\rho - z) - y$

$\frac{dz}{dt} = xy - \beta z$

I used the second equation to make the chaotic time series model. Then we try to predict the chaotic time series using those deep learning models.

7 Different Error Metrics:

I mainly used mse and mae i.e. mean squared error and mean absolute error to train the model. I also used other error metrics to make complete analysis of the models. I listed the error metrics here which I will be using in later part to compare the models.

7.1 Mean Square Error (MSE)-

Mean squared error is one of the most common error metrics used in time series analysis. The value is calculated as using this following formula:

MSE = $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

y_i = observation at i th step

$\hat{y}_i$ = prediction at i th step

7.2 Mean Absolute Error (MAE)-

This is another one of the most commonly used metrics to understand about accuracy of any regression model. The value is calculated using the following formula:

MAE = $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$

y_i = observation at i th step

$\hat{y}_i$ = prediction at i th step

7.3 Forecast and Predictand as Bivariate Random Variable.

As we noted in the forecast/predictand pair (F; P) form a bivariate random variable with a joint density function f_{FP}. The conditional density functions $f_{F|P=p}$ and $f_{P|F=f}$ tell us something about the performance of the forecast. I computed this values for my models also by generating the histograms and calculating the the number of data points in that range for the particular data points in the observation and prediction dataset.

7.4 Correlation skill score

The correlation between the forecast F and the verifying observation P is called the correlation skill score and is given by

$$\frac{\text{Cov}(F, P)}{\sqrt{\text{Var}(F)\text{Var}(P)}}$$

Clearly for a perfect forecast the value of this skill score will be 1.

7.5 Proportion of explained variance-

The proportion of explained variance is the percentage of actual variance that is explained by prediction. The value is calculated as using this following formula:

Proportion of explained variance = $1 - \frac{Var(F-P)}{Var(P)}$

F is prediction, P is observed

8 Results of individual models:

8.1 Result of ANN model-

The model was developed using Keras and Tensorflow. Data from the previous 3, 5, and 7 days was utilized to predict the value of the following day. The training process involved 10 epochs with a batch size of 1024. Various error metrics were computed to evaluate the model's performance, and subsequent analysis was conducted based on these metrics.

8.1.1 Error metrics-

In the given Table 1 I have put the value of error metrics of the three model i.e. the 3,5 and 7 days lookback model.
Here model where 3 previous values were used to predict the next value is denoted as Model 3, in the same way model where 5 and previous values were used to predict the next value is denoted as Model 5 and Model 7 respectively.

Model name and error metrics	**Model 3**	**Model 5**	**Model 7**
MSE	0.00772	0.00449	0.003561
MAE	0.06871	0.051048	0.045
Maximum Error	0.32455	0.25078	0.21940
Minimum error	-0.26129	-0.24801	-0.27051
Variance in prediction	0.00365	0.00759	0.00981
Co-relation skill score	0.90561	0.90622	0.90575
Proportion of explained variable	0.71441	0.64380	0.77799

Table 1: Error metrics of ANN models

8.1.2 Distribution of error-

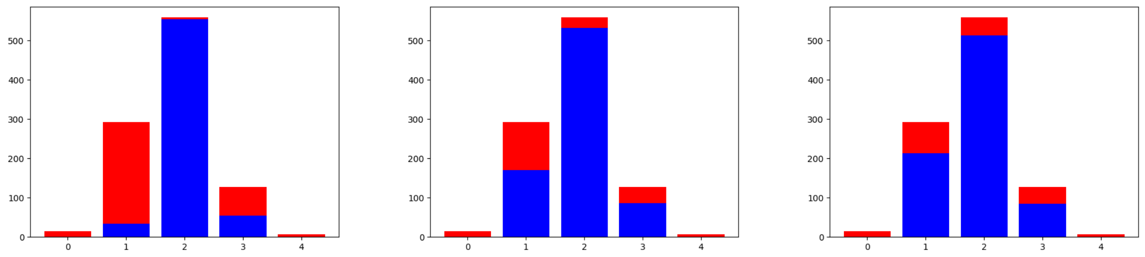
I tried to understand how the error are distributed in the dataset. To do this I divided the dataset into 5 parts. The parts are 0 to 0.2, 0.2 to 0.4, 0.4 to 0.6, 0.6 to 0.8 and 0.8 to 1 as the dataset is already normalized through Min Max Scaling. Then I got how many data point lie in these range and how many of them predicted in the corresponding. I got 5 probability values for each case.

For Model 3 it was [0.0, 0.11643835616438356, 0.9910554561717353, 0.4251968503937008, 0.0].

For Model 5 it was [0.0, 0.5821917808219178, 0.9534883720930233, 0.6771653543307087, 0.0]

For Model 7 it was [0.0, 0.7294520547945206, 0.9177101967799642, 0.6614173228346457, 0.0].

Here in Figure 2 I have shown this through bar plot.



(a) ANN with 3 previous days (b) ANN with 5 previous days (c) ANN with 7 previous days

Figure 2: Comparison of ANN Models

8.1.3 Observation-

Observing the error metrics and the distribution of error it is clear that model is improving when we are increasing the number of previous days. Also we can see though the model predict better in the middle part but can not capture the outlier values.

8.2 Result of LSTM model-

The model was developed using Keras and Tensorflow. Data from the previous 3, 5, and 7 days was utilized to predict the value of the following day. Here also the training process involved 10 epochs with a batch size of 1024. Various error metrics were computed to evaluate the model's performance, and subsequent analysis was conducted based on these metrics.

8.2.1 Error metrics-

In the given Table 2 I have put the value of error metrics of the three model i.e. the 3,5 and 7 days lookback model.
Here model where 3 previous values were used to predict the next value is denoted as Model 3, in the same way model where 5 and previous values were used to predict the next value is denoted as Model 5 and Model 7 respectively.

Model name and error metrics	**Model 3**	**Model 5**	**Model 7**
MSE	0.00324	0.00343	0.003381
MAE	0.04274	0.04423	0.04433
Maximum Error	0.219	0.19543	0.21729
Minimum error	-0.24738	-0.24827	-0.28372
Variance in prediction	0.01425	0.01457	0.01443
Co-relation skill score	0.90416	0.89931	0.89957
Proportion of explained variable	0.81521	0.80599	0.80684

Table 2: Error metrics of ANN models

8.2.2 Distribution of error-

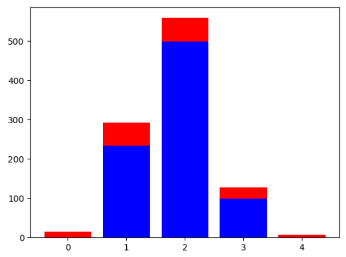
AS I did in the previous case in this case also I tried to understand how the error are distributed in the dataset. To do this again I divided the dataset into 5 parts. The parts are 0 to 0.2, 0.2 to 0.4, 0.4 to 0.6, 0.6 to 0.8 and 0.8 to 1 as the dataset is already normalized through Min Max Scaling. Then I got how many data point lie in these range and how many of them predicted in the corresponding. I got 5 probability values for each case.

For Model 3 it was [0.0, 0.8013698630136986, 0.8926654740608229, 0.7795275590551181, 0.0].

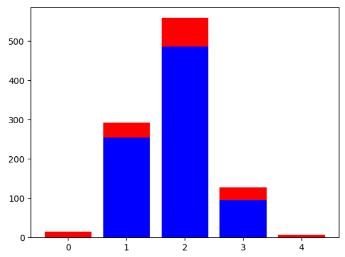
For Model 5 it was [0.06666666666666667, 0.8732876712328768, 0.8694096601073346, 0.7559055118110236, 0.0]].

For Model 7 it was [0.06666666666666667, 0.8732876712328768, 0.8640429338103757, 0.8031496062992126, 0.0].

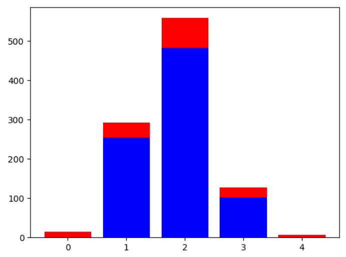
Here in Figure 3 I have shown this through bar plot.



(a) LSTM with 3 previous days



(b) LSTM with 5 previous days



(c) LSTM with 7 previous days

Figure 3: Comparison of LSTM Models

8.2.3 Observation-

Observing the error metrics and the distribution of error it is clear that model is not improving much when we are increasing the number of previous days. But also we can see though the model predict better in the middle part but can not capture the outlier values very well. Model 3 it could not predict anything but in Model 5 and Model 7 it predict a little bit better.

8.3 Result of N-BEATS model-

The model was developed using Keras N-BEATS and Tensorflow. Data from the previous 3, 5, and 7 days was utilized to predict the value of the following day. Here also the training process involved 10 epochs with a batch size of 1024. Here also various error metrics were computed to evaluate the model's performance, and subsequent analysis was conducted based on these metrics.

8.3.1 Error metrics-

In the given Table 3 I have put the value of error metrics of the three model i.e. the 3,5 and 7 days lookback model.
Here model where 3 previous values were used to predict the next value is denoted as Model 3, in the same way model where 5 and previous values were used to predict the next value is denoted as Model 5 and Model 7 respectively.

Model name and error metrics	Model 3	Model 5	Model 7
MSE	0.0029	0.00294	0.00295
MAE	0.03895	0.03941	0.03957
Maximum Error	0.20242	0.21982	0.23013
Minimum error	-0.24111	-0.22856	-0.22203
Variance in prediction	0.01709	0.01725	0.01724
Co-relation skill score	0.91673	0.91655	0.91671
Proportion of explained variable	0.83216	0.831	0.83138

Table 3: Error metrics of ANN models

8.3.2 Distribution of error-

AS I did in the previous case in this case also I tried to understand how the error are distributed in the dataset. To do this again I divided the dataset into 5 parts. The parts are 0 to 0.2, 0.2 to 0.4, 0.4 to 0.6, 0.6 to 0.8 and 0.8 to 1 as the dataset is already normalized through Min Max Scaling. Then I got how many data point lie in these range and how many of them predicted in the corresponding. I got 5 probability values for each case.

For Model 3 it was [0.3333333333333333, 0.8184931506849316, 0.889087656529517, 0.8267716535433071, 0.2857142857142857].

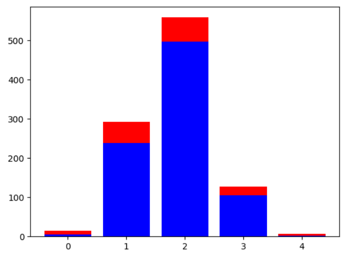
For Model 5 it was [0.3333333333333333, 0.8184931506849316, 0.889087656529517, 0.8031496062992126, 0.42857142857142855].

For Model 7 it was [0.4666666666666667, 0.839041095890411, 0.889087656529517, 0.8188976377952756, 0.2857142857142857].

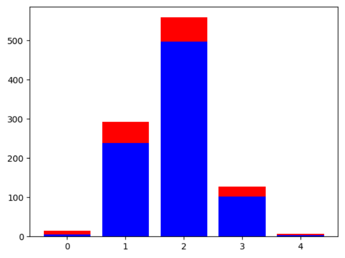
Here in Figure 4 I have shown this through bar plot.

8.3.3 Observation-

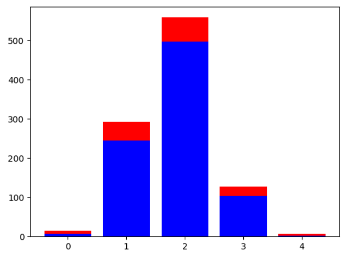
Observing the error metrics and the distribution of error it is clear that model is not improving much when we are increasing the number of previous days. They also mentioned in the paper that the model performs well when it is 3 and that is also depicted here. Also we can see though the model predict better in the middle part and fairly well in the outside outliers then the other models.



(a) N-BEATS with 3 previous days



(b) N-BEATS with 5 previous days



(c) N-BEATS with 7 previous days

Figure 4: Comparison of N-BEATS Models

8.4 Result of the Average model-

The is a type of ensemble model. The model is developed by averaging the prediction of the three deep learning time series which I talked earlier. I averaged them and used the average as the new prediction. Here I given the analysis which I did using this model. Here I used the prediction of the deep learning models which taken three previous data for prediction.

8.4.1 Error metrics-

In the given Table 4 I have put the value of error metrics of the average model.

Model name and error metrics	**Average model**
MSE	0.00302
MAE	0.04146
Maximum Error	0.22559
Minimum error	-0.2279
Variance in prediction	0.00999
Co-relation skill score	0.92467
Proportion of explained variable	0.81063

Table 4: Error metrics of Average model

8.4.2 Distribution of error-

AS I did in the previous cases in this case also I tried to understand how the error are distributed in the dataset. To do this again I divided the dataset into 5 parts. The parts are 0 to 0.2, 0.2 to 0.4, 0.4 to 0.6, 0.6 to 0.8 and 0.8 to 1 as the dataset is already normalized through Min Max Scaling. Then I got how many data point lie in these range and how many of them predicted in the corresponding. I got 5 probability values for this case also.

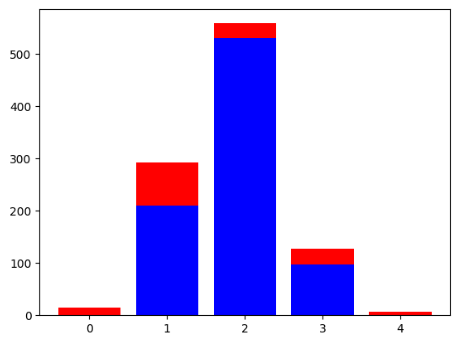


Figure 5: Average model

The values I got for the average model is [0.0, 0.7191780821917808, 0.9499105545617174, 0.7716535433070866, 0.0].

Here in Figure 5 I have shown this through bar plot of the Average model.

8.4.3 Observation-

Observing the error metrics and the distribution of error it is clear that model is performing well but this is also true that the N-BEATS model is alone it. So this model is not a very useful ensemble model.

8.5 Result of the Meta model-

The is a type of ensemble model. The model is developed by using a meta model and the prediction of the three deep learning time series which I talked earlier. I took the there prediction and put it as an input in a simple ANN model and use a validation set of thousand points for training and then used the test data of the same formate for testing. Here I given the analysis which I did using this model. Here I used the prediction of the deep learning models which taken three previous data for prediction.

8.5.1 Error metrics-

In the given Table 5 I have put the value of error metrics of the meta model.

8.5.2 Distribution of error-

AS I did in the previous cases in this case also I tried to understand how the error are distributed in the dataset. To do this again I divided the dataset into 5 parts. The parts are 0 to 0.2, 0.2 to 0.4, 0.4 to 0.6, 0.6 to 0.8 and 0.8 to 1 as the dataset is already normalized through Min Max Scaling. Then I got how many data point lie in these range and how many of them predicted in the corresponding. I got 5 probability values for this case also.

Model name and error metrics	Meta model
MSE	0.00263
MAE	0.03751
Maximum Error	0.20127
Minimum error	-0.21656
Variance in prediction	0.01571
Co-relation skill score	0.92261
Proportion of explained variable	0.84797

Table 5: Error metrics of Meta model

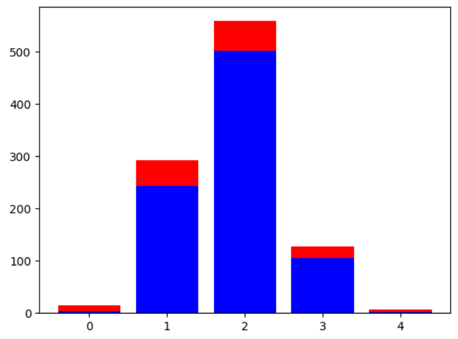


Figure 6: META model

The values I got for the average model is [0.2, 0.8356164383561644, 0.8998211091234347, 0.8346456692913385, 0.2857142857142857]].

Here in Figure 6 I have shown this through bar plot of the Meta model.

8.5.3 Observation-

Observing the error metrics and the distribution of error it is clear that model is performing well then the N-BEATS model. So we can consider it as a good model as it provide better prediction then any one of the individual model.

8.6 Result of the confidence model-

The is also a type of ensemble model. The model is developed by using a weighted average technique described in 5.2. Here I given the analysis which I did using this model. Here I used the prediction of the deep learning models which taken three previous data for prediction.

8.6.1 Error metrics-

In the given Table 6 I have put the value of error metrics of the confidence model.

Model name and error metrics	**Confidence model**
MSE	0.02659
MAE	0.1287
Maximum Error	0.51554
Minimum error	-0.57647
Variance in prediction	0.01014
Co-relation skill score	0.012279
Proportion of explained variable	-0.38633

Table 6: Error metrics of Confidence model

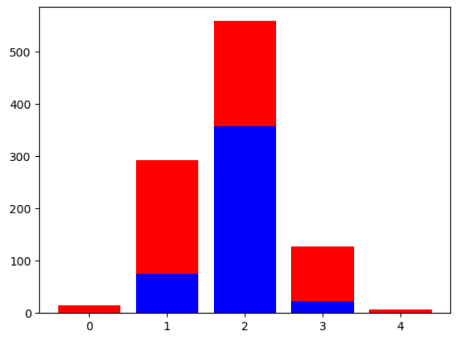


Figure 7: Confidence model

8.6.2 Distribution of error-

AS I did in the previous cases in this case also I tried to understand how the error are distributed in the dataset. To do this again I divided the dataset into 5 parts. The parts are 0 to 0.2, 0.2 to 0.4, 0.4 to 0.6, 0.6 to 0.8 and 0.8 to 1 as the dataset is already normalized through Min Max Scaling. Then I got how many data point lie in these range and how many of them predicted in the corresponding. I got 5 probability values for this case also.

The values I got for the average model is [0.0, 0.2568493150684932, 0.6404293381037567, 0.18110236220472442, 0.0].

Here in Figure 7 I have shown this through bar plot of the Confidence model.

8.6.3 Observation-

Observing the error metrics and the distribution of error it is clear that model is not at all performing well though the way of doing the averaging sounds logical. So we would not consider it as a good ensemble model.

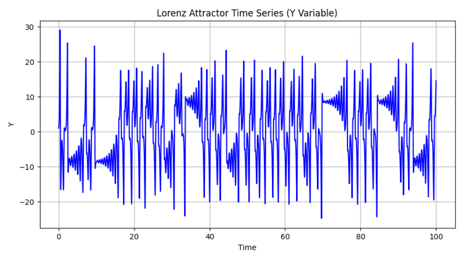


Figure 8: Lorenz Attractor Time Series (Y Variable)

8.7 Result of the Chaotic time series model-

As I discussed in 6 part that I have developed a chaotic time series model using Lorenz system. The time series I got is shown in the Figure 8.

8.7.1 Error metrics-

In the given Table 7 I have put the value of error metrics of the the results I got from this three models. Here the values are not normalized.

Model name and error metrics	**ANN**	**LSTM**	**N-BEATS**
MSE	1.17392	0.72912	0.441452
MAE	0.71992	0.54816	0.42322
Maximum Error	4.34663	4.6102	3.94549
Minimum error	-4.38929	-2.56676	-1.93329
Variance in prediction	74.70262	75.163	73.88834

Table 7: Error metrics of different models

8.7.2 Distribution of error-

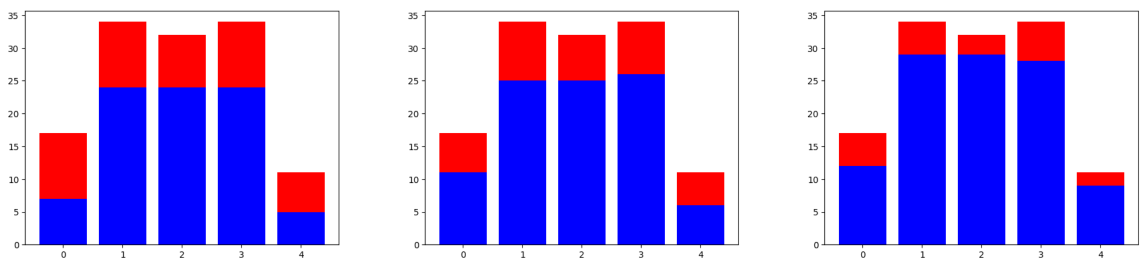
AS I did in the previous cases in this case also I tried to understand how the error are distributed in the dataset. To do this again I divided the dataset into 5 parts. The parts are 0 to 0.2, 0.2 to 0.4, 0.4 to 0.6, 0.6 to 0.8 and 0.8 to 1 as the dataset is already normalized through Min Max Scaling. Then I got how many data point lie in these range and how many of them predicted in the corresponding. I got 5 probability values for this case also.

The values I got for the ANN is [0.4117647058823529, 0.7058823529411765, 0.75, 0.7058823529411765, 0.45454545454545453].

The values I got for the LSTM is [0.6470588235294118, 0.7352941176470589, 0.78125, 0.7647058823529411, 0.5454545454545454].

The values I got for the N-BEATS is [0.7058823529411765, 0.8529411764705882, 0.90625, 0.8235294117647058, 0.8181818181818182].

Here in Figure 9 I have shown this through bar plot of the Confidence model.



(a) ANN Chaotic model (b) LSTM Chaotic model (c) N-BEATS Chaotic model

Figure 9: Comparison of different Models

8.7.3 Observation-

Observing the error metrics and the distribution of error it is clear that N-BEATS is performing better then LSTM which is performing better then ANN which is quite expected as the model complexity increased it try to capture more variations of the data in the model.

9 Comparison of results of different models:

Here I will try to compare between all the model which I have discused earlier. We have developed some deep neural model for our time series forecasting. Also we have some simple model like linear regression or exponential smoothing model for reference. After going through all of them it is clear that overall N-BEATS model perform better then other models. But after doing detailed analysis of each part it is also clear that there are some parts of the dataset where other model can perform better then the N-BEATS model. So there are places where the model can perform better. Also it would be interesting to see how the more modern and recent models will work and can the outperform all the existing model from all cases.

Also from our study of different ensemble models as we saw we have found some models which is better then any individual model. Yes this is also true as I saw that some of the ensemble techniques were performing really bad. So we can try more complex ensemble technique only using this prediction and error metrics which may give more better prediction then what we got in this project through these models.

10 Conclusion:

This was a project focused on time series prediction using deep neural network models. The model I discussed were developed through the time trying to solve the problem of the existing model. It outperform the existing models in some aspect but can not completely outperform it from all the aspects if we analyse it from different perspective through different type of error analysis. The N-BEATS is the most recent among all of them and performed quite well in our dataset along with in the chaotic time series model.

The project can be extended in different ways. We can include all the existing deep neural language models and all type of error metrics to come up with more concrete conclusion which I discussed earlier.

11 References

1. Boris N. Oreshkin Element, Dmitri Carpov Element, Nicolas Chapados Element, Yoshua Bengio Mila yoshua.(2020). N-BEATS: NEURAL BASIS EXPANSION ANALYSIS FOR INTERPRETABLE TIME SERIES FORECASTING. Published as a conference paper at ICLR 2020

2. Keras End to End Solution usingNBeats
`https://www.kaggle.com/code/andy8744/beginner-keras-end-to-end-solution-usingnbeats`

3. KerasBeats: An Easy Way to Use N-Beats in Keras, Jonathan Bechtel
`https://medium.com/@jonathanbechtel/kerasbeats-an-easy-way-to-use-n-beats-in-keras-395b24c5cc28`

4. NBEATS `https://nixtlaverse.nixtla.io/neuralforecast/models.nbeats.html`

5. Storch, H. von, and Zwiers, F. W. (2003). Statistical Analysis in Climate Research. Cambridge University Press.

6.Hyndman, R.J., and Athanasopoulos, G. (2021) Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3

7.Murphy Allen H, Brown Barbar G and Chen Yin-Sheng, 1989, Diagnostic Verification of Temperature Forecasts, American Meteorological Society