Section 1: Task Definition, Evaluation Protocol, and Data

Intended Task: To explore the accuracy of numerical weather prediction (NWP) outcomes with different learning models to predict weather outcomes using user-friendly forecast data. First, to replicate the results of the paper BharatBench: Dataset for data-driven forecasting over India by Animesh Choudhury, Jagabandhu Panda, and Asmita Mukherjee, [1] which sought to validate the utility of its dataset for medium-range, data-driven forecasting. The inputs are from a novel dataset, BharatBench, with data recorded from 1990-2020, tracking various atmospheric conditions at surface pressure and at 13 vertical levels. The outputs are meteorological variables at future time steps, notably temperature and precipitation at various pressure levels. Performance metrics implemented include Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Anomaly Correction Coefficient (ACC); implemented models are the Persistence Forecast model, Climatological and Weekly Climatological models, Linear Regression model, Convolutional Neural Network (CNN) and Convolutional Long Short Term Memory (ConvLSTM) models. Data from years 1990-2017 were used for training; data from 2018 was used for validation; and data from years 2019-2020 were used for testing. Second, to observe the replicated results of the study and identify and test alternative evaluation models to achieve more efficient or more accurate forecasting.

Data Set: The dataset, named BharatBench, is curated using the IMDAA (Indian Meteorological Department Analysis and Assimilation) reanalysis dataset. It includes detailed meteorological information available from 1990 to 2020, clipped for 5°N to 40°N and 65°E to 100°E, at a resolution of 1.08°. Data was coordinated over latitude, longitude and time. The dataset includes the following surface variables (one vertical level): TMP_2m (temperature at 2m height); UGRD_10m (east-west wind component at 10m height); VGRD_10m (north-south wind component at 10m height); PRMSL_msl (mean sea level pressure); and APCP_sfc (total precipitation at sea level). Atmospheric variables, for 13 vertical pressure levels, include: TMP_prl (temperature); RH_prl (relative humidity); HGT_prl (geopotential height); UGRD_prl (east-west wind component); and VGRD_prl (north-south wind component). Constant variables include: Land_sfc (Land sea mask), MTERH_sfc (Terrain Height). Target variables considered were: H500 (geopotential height at 500hPa), T850 (temperature at 850 hPa), T2m (temperature at 2m), and TP6h (precipitation over 6 hours).

Metrics: The three evaluation metrics are as follows: Root Mean Square Error (RMSE): Measures the square root of the average of squared differences between predicted and actual values. It is sensitive to outliers and is preferred when outliers are rare. Mean Absolute Error (MAE): Measures the average of absolute differences between predicted and actual values, providing a straightforward interpretation of model accuracy. Anomaly Correlation Coefficient (ACC): Measures the spatial correlation between anomalies of forecasts and verifying values with climatological values. ACC values range from +1 (perfect correlation) to -1 (perfect anti-correlation), with values near +1 indicating high forecast accuracy. Evaluation matrices were captured using these metrics for each of the learning models implemented.

Section 2: Neural Network Machine Learning Model

The implementation uses TensorFlow's Keras API which provides an approachable, highly-productive interface for solving ML problems. Various layers such as Conv2D, MaxPooling2D,UpSampling2D, ConvLSTM2D,MaxPooling3D, UpSampling3D,Conv3D and Dropout are used. These layers are used as building blocks for CNNs and ConvLSTM models.

CNN: CNNs are a natural choice for spatial data due to their ability to effectively capture patterns and features regardless of their location or orientation within the data. The training of CNN is done with convolutional layers(Conv2D), max-pooling layers (MaxPooling2D), upsampling layers (UpSampling2D), and a dropout layer (Dropout).

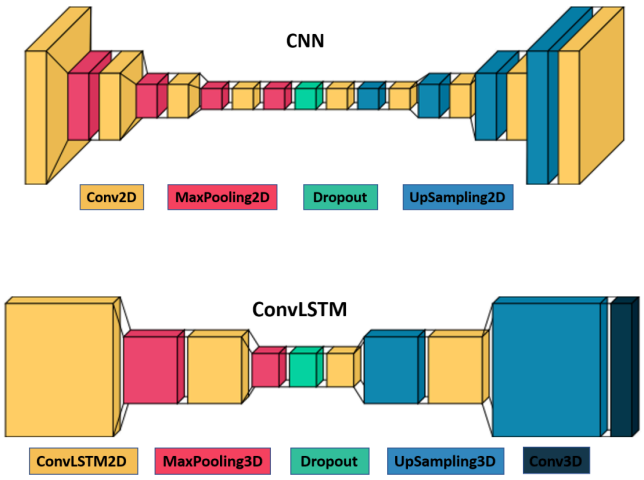
convLSTM: ConvLSTM is a type of recurrent neural network for spatio-temporal prediction that has convolutional structures in both the input-to-state and state-to-state transitions. It combines the spatial capabilities of CNNs with the temporal sequencing of LSTMs (Long Short Term Memory), making it suitable for analyzing sequences of spatial data over time.

Convolutional Layers apply a convolution operation to the input, passing the result to the next layer. This operation involves sliding a filter or kernel over the input data and computing the dot product of the filter and local regions of the input to produce a feature map.

Upsampling layers increase the spatial dimensions (height and width) of the input volume. This can be done through methods like nearest neighbor or bilinear upsampling, where the input data is effectively "stretched" to a larger size by interpolating new values between existing ones.

Max pooling layers reduce the spatial dimensions (height and width) of the input volume. This is achieved by sliding a window over the input and taking the maximum value in each window as the output.

Dropout layers randomly set a fraction of input units to 0 at each update during training time, which helps to prevent overfitting. This means that each update to a layer during training is performed with a different "view" of the configured layer.



Section 3: Experiment Design

Numerical methods that simulate physics on gridded meshes are computationally expensive. Machine Learning (ML) has been employed in this domain and significant progress has occurred in the past decade, leading to ML applications that are now competitive with traditional numerical methods.

Hypothesis:

Incorporating spherical convolution layers and probabilistic forecasting methods will improve the accuracy and reliability of the weather prediction model. The spherical convolution layers will enhance the model's ability to capture vertical atmospheric dynamics, while probabilistic forecasting will provide a better understanding of forecast uncertainty and improve the prediction of extreme weather events.

Dependent Variables

Root Mean Square Error (RMSE)
Mean Absolute Error (MAE)
Anomaly Correlation Coefficient (ACC)
Learning Curves
Convergence Intervals
Geopotential height at 500 hPa (H500), temperature at 850 hPa (T850), temperature at 2m (T2m), six hourly accumulated precipitation (TP6h).

Independent Variables

Hyper Parameters	Learning Parameters
Kernel size	Learning rate
Number of filters	Batch size
Activation functions	Number of epochs

Control Variables:

1. Dataset: Use the same dataset as BharatBench
2. Training and Validation Process: Keep the training and validation process consistent.
3. Lead_days : The number of days ahead the forecast is for.

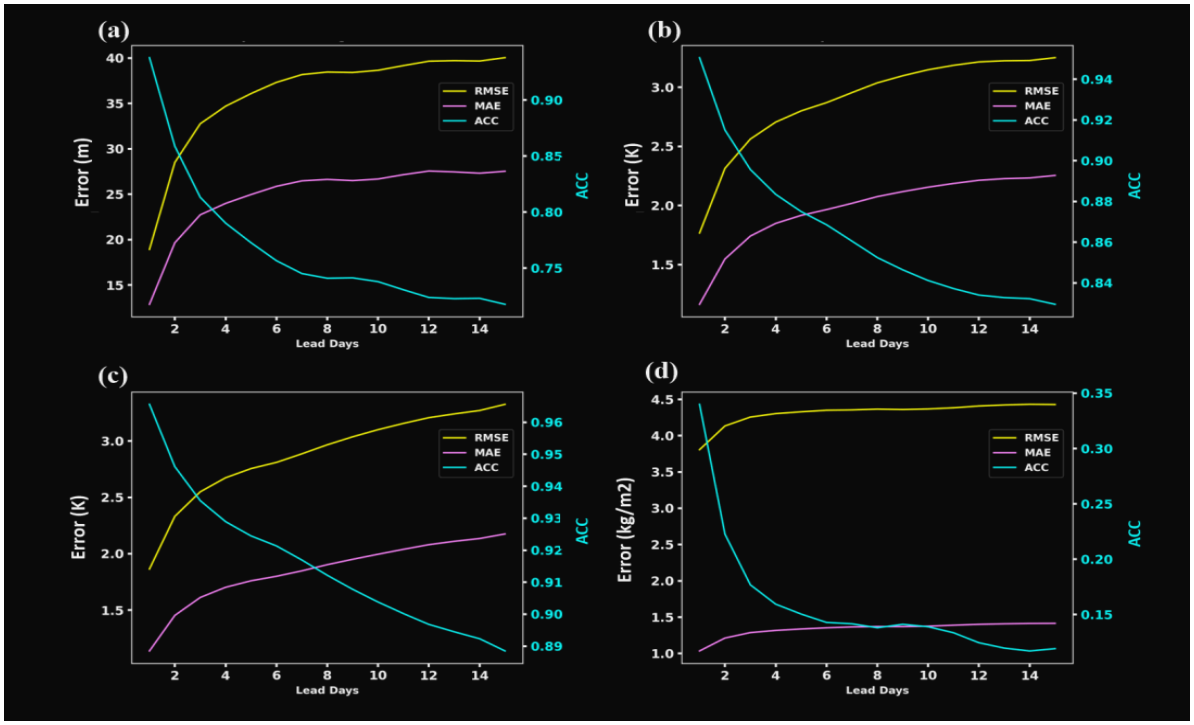
Methodology:

a. Set Up the Spherical Convolution Layer. Try with an existing spherical convolution layer or develop a custom one.
b. Incorporate the spherical convolution layer into your neural network model.
c. Normalize the data and split it into training, validation, and test sets.
d. Train the model using the prepared dataset.
e. Evaluate the model on the test dataset and interpret the results
f. Small random changes are added to the initial observational data used by the model with techniques like Ensemble Kalman Filter (EnKF) or Ensemble Variational Methods (EnVar)
g. Model parameters are varied within realistic bounds to generate ensemble members.

Section 4: Experimental Results and Discussion

With the BharatBench, the linear regression model predicted four selected variables (H500, T850, T2m, TP6h) three and five days ahead. The model was notably successful in predicting three days in advance, and its five-day predictions were comparable to climatological forecasts. CNN and ConvLSTM models were used for H500 and T850, with CNN showing better performance after extensive training. However, both models struggled to surpass linear regression for T850 forecasts.

		3 days		5 days	
Metric	**Model**	**H500**	**T850**	**H500**	**T850**
RMSE	**Linear**	26.61557	2.005987	29.58229	2.202205
	CNN	26.27218	2.053407	28.80511	2.256589
	ConvLSTM	26.87115	2.210813	29.5925	2.381422
MAE	**Linear**	18.74379	1.396874	20.6882	1.539523
	CNN	18.42743	1.433121	20.10753	1.577852
	ConvLSTM	18.80561	1.535963	20.74711	1.657516
ACC	**Linear**	0.868517	0.933918	0.833596	0.919551
	CNN	0.872457	0.930596	0.843826	0.915599
	ConvLSTM	0.866052	0.919435	0.83462	0.905175



(a) H500, (b) T850, (c) T2m, and (d) TP6h at different lead days

RMSE, MAE, and ACC values after the proposed modifications will be compared to those in the table above. Additionally, the relationship between errors and lead days for all predicted variables will be graphed after the modifications.

Section 5: References

Choudhury, A., Panda, J., & Mukherjee, A. (2024). BharatBench: Dataset for data-driven weather forecasting over India. *Department of Earth and Atmospheric Sciences, National Institute of Technology, Rourkela, India*. [arXiv:2405.07534]

Original dataset: https://www.kaggle.com/datasets/maslab/bharatbench

Source code: https://github.com/MASLABnitrkl/BharatBench.

What are Convolutional Neural Networks? | IBM-
https://www.ibm.com/topics/convolutional-neural-networks,

ConvLSTM Explained | Papers With Code
https://paperswithcode.com/method/convlstm

Taco S. Cohen, Mario Geiger, Jonas Koehler, & Max Welling. (2018). Spherical CNNs. [https://arxiv.org/pdf/1801.10130], https://github.com/jonkhler/s2cnn?utm_source=catalyzex.com

Section 6: Viability Test

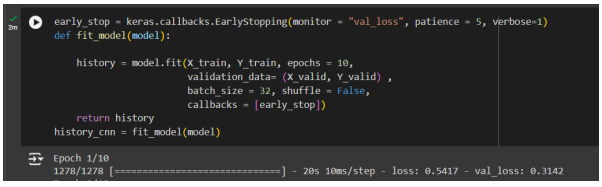
<table>
<tr><td>CNN</td><td>convLSTM</td></tr>
<tr><td>Number of training samples

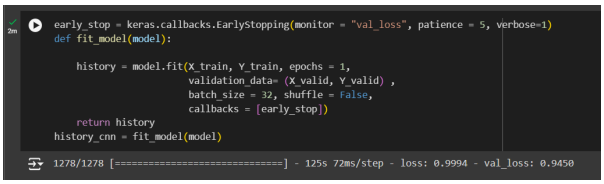
Total params: 181057 (707.25 KB)
Trainable params: 181057 (707.25 KB)
Non-trainable params: 0 (0.00 Byte)</td><td>Number of training samples

Total params: 1544225 (5.89 MB)
Trainable params: 1544225 (5.89 MB)
Non-trainable params: 0 (0.00 Byte)</td></tr>
<tr><td>Training (x and y), validation (x and y) and test data (x and y)

(40888, 32, 32, 1)
(40888, 32, 32, 1)
(1440, 32, 32, 1)
(1440, 32, 32, 1)
(2904, 32, 32, 1)
(2904, 32, 32, 1)</td><td>Training (x and y), validation (x and y) and test data (x and y)

(40888, 1, 32, 32, 1)
(40888, 1, 32, 32, 1)
(1440, 1, 32, 32, 1)
(1440, 1, 32, 32, 1)
(2904, 1, 32, 32, 1)
(2904, 1, 32, 32, 1)</td></tr>
<tr><td>Training over 1 epoch

  </td><td>Training over 1 epoch

  </td></tr>
<tr><td>Evaluation metrics

91/91 [==========] - 0s 3ms/step
RMSE: 2.5778944
MAE 1.809538
ACC 0.8881967</td><td>Evaluation metrics

91/91 [==========] - 3s 14ms/step
RMSE: 5.8957796
MAE 4.033869
ACC -1.3858196e-06</td></tr>
</table>