# Predictive Analysis to Forecast Weather Variables with Graph Neural Networks

Nita Shah[1,2], Avni Pancholi[1,3], Akash Thakur[1,4]

Department of Applied Mathematical Science and Analytics,

Gujarat University, Ahmedabad-380009

Gujarat, India

Corresponding author: [2]Prof. Dr. Nita H Shah

ORCID ID: [3]0009-0007-4457-0637

ORCID ID: [4]0000-0001-7360-9130

## 1. Abstract

Weather forecasting is vital for diverse applications, from agriculture to disaster readiness. This study employs ERA5 data and Graph Neural Networks (GNN) to predict weather variables over five days. Preprocessing ERA5 data creates training, validation, and testing datasets. GNNs capture spatiotemporal relationships within weather data, surpassing persistence forecasting. The workflow includes data preprocessing, feature extraction, dataset partitioning, and model training. Evaluation metrics like RMSE, bias, and correlation assess GNN performance against persistence forecasting. GNNs exhibit superior accuracy and reliability across spatial and temporal scales. Findings emphasize GNNs' efficacy in capturing spatial dependencies and temporal dynamics in weather data. This research enhances weather prediction accuracy by integrating advanced machine learning, providing valuable insights for socio-economic activities.

***Keywords:*** Weather forecasting, ERA5 data, Graph Neural Networks, Machine Learning, Persistence forecasting, Performance metrics

## 2. Introduction

### 2.1 Background

Weather forecasting, an amalgamation of art and science, has profoundly evolved. From early civilizations' rudimentary observations to the sophisticated methods of the nineteenth century, forecasting has become indispensable in contemporary society. Initially reliant on basic atmospheric indicators like barometric pressure and cloud cover, modern forecasting harnesses advanced technology, particularly computer-based modeling, to integrate many atmospheric variables, thus enhancing precision and reliability (McGovern et al., 2017) [1].

### 2.2 Research Problem and Objectives

The significance of accurate weather forecasts spans myriad domains, from disaster preparedness to agricultural management and energy distribution. However, traditional forecasting methods, though effective to varying degrees, often encounter limitations in accurately capturing complex atmospheric dynamics. This study aims to address the limitations by employing Graph Neural Networks (GNN) to predict weather variables. Specifically, the research seeks to:

- Investigate the efficacy of GNNs in weather forecasting compared to traditional methods.
- Assess the performance of GNNs in capturing spatiotemporal dependencies inherent in weather data.
- Enhance the precision and reliability of weather predictions through advanced machine-learning techniques.

### 2.3 Hypothesis

Using Graph Neural Networks for weather forecasting is expected to produce more accurate and reliable predictions compared to traditional methods like persistence forecasting. Graph Neural Networks are anticipated to better capture the relationships between different weather variables over time and space, leading to improved forecasting outcomes.

## 3. Literature Review

**McGovern et al., 2017 [1]** demonstrate AI's potential to enhance weather forecasting accuracy, in particular, to predict severe storms and hurricanes, by integrating machine learning with observational data.

**Ganssle, G., 2018 [2]** demonstrate the architecture of GNN: Node Classification by Graph Convolutional Network, predicting nodes properties and its applications.

**Zhang et al., 2019 [3]** showcase the efficacy of Graph Neural Networks in learning complex data relationships, utilizing a two-layered architecture for node classification tasks.

**Rasp et al. (2020) [4]** introduce a benchmark dataset for data-driven weather forecasting, derived from the ERA5 archive, aiming to standardize evaluation metrics and accelerate research in this field.

**Remi Lam et al. (2022) [5]** present "GraphCast" a machine learning-based method for global medium-range weather forecasting, which outperforms operational deterministic systems, particularly in predicting severe weather events.

**Keisler, R., 2022 [6]** present a data-driven approach for forecasting global weather using graph neural networks. The underlying model is trained on reanalysis data from ERA5 or forecast data from GFS.

**Hannah Ritchie (2024) [7]** discusses the evolution of weather prediction capabilities and emphasizes the need to address global inequalities in weather forecasting resources, highlighting advancements in forecasting technologies and disparities in access across regions.

## 4. Methodology

### 4.1 Data Set

The ERA5 data utilized in this study represents the fifth iteration of ECMWF reanalysis, spanning the last eight decades from 1940 onwards. Leveraging the advanced ECMWF Integrated Forecasting System (IFS) model, ERA5 offers enhanced spatial and temporal resolution compared to its predecessor, ERA-Interim reanalysis. With a spatial resolution of 31 km and 137 vertical levels, ERA5 provides detailed insights into atmospheric conditions. The assimilation system integrated into ERA5 incorporates diverse measurement forms to enhance data quality, while model ensembles offer estimates of uncertainty levels. Due to the vast size of the raw data (nearly 700 GB for a single vertical level), data re-gridding to lower resolutions (e.g., 5.625°) is performed using bilinear interpolation via the xesmf Python Package. Additionally, for 3D fields, 13 vertical levels are selected based on pressure in hecto-Pascals (hPa) instead of physical height: 50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, and 1000 hPa. A subset of the data spanning 17 years (2002-2018) is utilized for training (2002-2015), validation (2016-2016), and testing (2017-2018) purposes. The data is structured into four dimensions: time, levels, latitude, and longitude, with a training dataset dimension of (1400, 52, 32, 64), where 52 indicates vertical levels as there are 13 levels for each of the 4 parameters.

### 4.2 Graph Neural Network (GNN)

Graph Neural Network (GNN) is tailored for analyzing interconnected graph-structured data, treating nodes as entities and edges as connections between them. Through graph convolution operations, GNNs update node representations by aggregating information from neighboring nodes, effectively capturing complex relationships within graph data.

### 4.2.1 Basic Components of GNNs [2]:

- **Graph Representation:** A graph $G = (V, E)$ comprises nodes $V$ and edges $E$, with the adjacency matrix $A$ representing connectivity.
- **Node Features:** Each node in the graph is associated with a feature vector. If there are N nodes, then the feature matrix X is of size N×D, where D is the dimensionality of the node features.
- **Neural Network Layers:** GNNs consist of multiple layers, aggregating information from neighboring nodes and updating node representations to learn complex relationships within the graph data.

### 4.2.2 Architecture of Graph Neural Network (GNN)

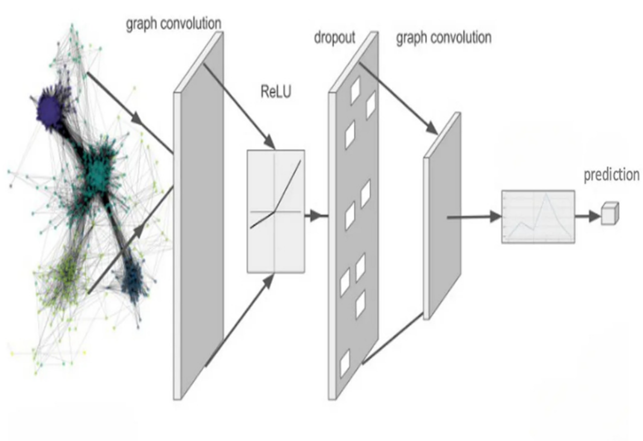

Figure 1: Architecture of GNN (Node Classification by Graph Convolutional Network) (Ganssle 2018, [2])

### 4.2.3 Graph Convolutional Networks (GCNs)

Graph Convolutional Network (GCN) layers are crucial in GNNs, allowing them to process graph-structured data by gathering insights from neighboring nodes and combining their features. This process helps GNNs create meaningful representations of nodes, essential for tasks like weather pattern prediction, node classification, and link prediction (Fig. 1).

Mathematically, the update rule for node representations in a graph convolutional layer can be expressed as:

$$h_v^{(l+1)} = \sigma\left(\sum_{u \in N(v)} \frac{1}{c_v} W^{(l)} h_u^{(l)}\right)$$

where $h_v^{(l)}$ represents the representation of node $v$ at layer $l$, $N(v)$ denotes the set of the neighbouring node of $v, W^{(l)}$ denotes the weight matrix at layer $l$, $\sigma$ represents the activation function, and $c_v$ is a normalization factor.

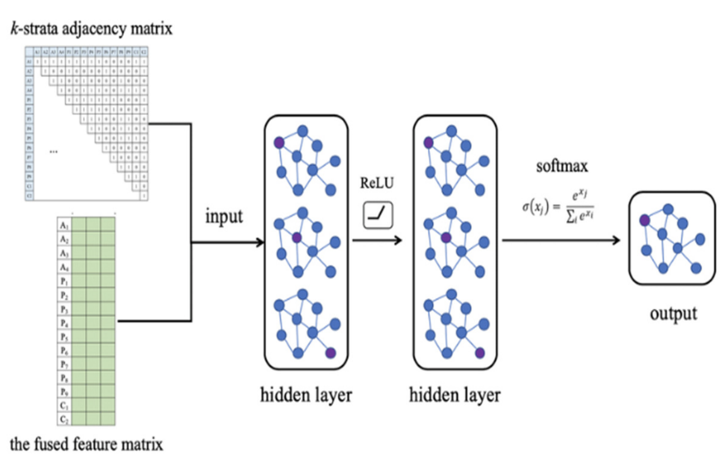

Figure 2: The adjacency matrix represents connections between nodes in a graph, while the feature matrix contains node attributes. These matrices serve as inputs to the neural network, which performs hidden layer computations with various activation functions to generate output. (Zhang 2019, [3])

### 4.2.4 GCN Layer Components:

- **Input Feature Matrix:** Represents structured data in graph-based machine learning models, with each row corresponding to a node and each column representing specific node features (e.g., temperature, humidity, wind speed) in weather prediction scenarios.
- **Weight Matrix:** Encapsulates parameters learned by the GCN layer during training, dictating how information from neighbouring nodes is aggregated to update node representations. These weights are optimized through backpropagation to learn relevant features for the task.
- **Message Passing:** Nodes update their representations by aggregating information from neighbours based on the learned weight matrix. Aggregated messages are transformed using an activation function to produce updated node representations, allowing nodes to exchange information and capture local and global structures within the graph.
- **Activation Function:** Applied element-wise to aggregated messages (e.g., ReLU, sigmoid), introducing non-linearity into node representations to capture complex relationships within the data and model intricate patterns and dependencies.

### 4.2.5 Dropout Layer

A dropout layer randomly deactivates a proportion of nodes or connections during training to prevent overreliance on specific features or nodes, promoting more robust learning, reducing overfitting, and encouraging the network to learn generalized representations.

#### 4.2.6 Multi-Layer Architectures

GNNs typically comprise multiple GCN layers stacked on top of each other to facilitate hierarchical feature learning. Each layer refines node representations by incorporating information from a broader neighborhood in the graph, allowing GNNs to capture increasingly complex patterns and relationships, leading to improved performance on various tasks.

### 4.3 Implementation

**4.3.1 Design:** Fig. 3 illustrates the process of training graph Neural network.

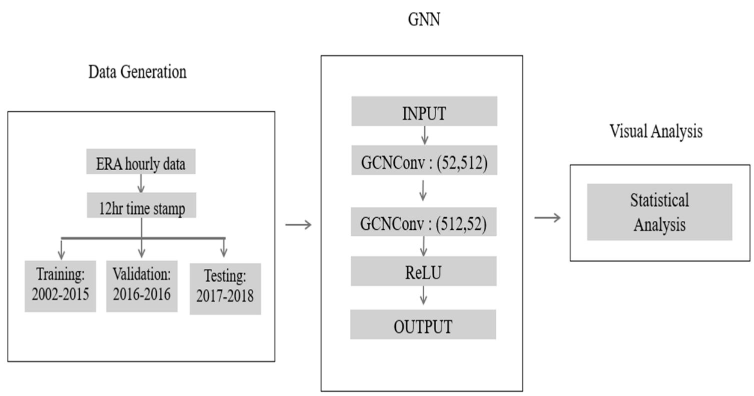


Figure 3: Flowchart illustrating the process of training a Graph Neural Network (GNN) model on input data with 52 dimensions, utilizing a hidden layer size of 512 and ReLU activation function. The model is trained for 250 epochs, followed by conducting various statistical analyses to evaluate its performance.

The input variables consist of temperature, humidity, u-component, and v-component of wind, each measured at 13 levels (50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, 1000 hPa). Two graph convolution layers are employed for training the model. The first layer, GCNConv, has 52 input channels (due to 13 levels for each variable) and a hidden dimension of 512. The subsequent layer is another graph convolution layer with 52 output channels. Rectified Linear Unit (ReLU) serves as the activation function. The training utilized the Adam optimizer and employed a mean square error loss function. The model was trained for 250 epochs.

### 4.3.2 Evaluation

The evaluation process is structured into three distinct categories: training, validation, and testing as shown in Table 1. This dataset is utilized to train the forecasting model. It encompasses historical weather data over a time span of 12 hours at a 5.625° resolution. The model learns from this dataset to capture patterns and relationships within the weather data.

| | |
|---|---|
| Training Dataset | January 1st, 2002 – December 31st, 2015 |
| Validation Dataset | January 1st, 2016 – December 31st, 2016 |
| Testing Dataset | January 1st, 2017 – December 31st, 2018 |

Table 1: Structure of dataset

The initial data is a five-dimensional array (1400, 13, 32, 64, 4) representing time stamps, levels, latitude points, longitude points, and parameters (temperature, humidity, u-component, v-component of wind). To apply graph-based techniques, it's reshaped into (1400, 2048, 52), where 2048 nodes are created based on latitude and longitude points, with 52 features per node derived from 4 parameters and 13 levels each. This data is structured into a graph where each node corresponds to a geographic point. A GNN model with a GCNConv layer (hidden dimension of 512) is trained over 250 epochs. Predicted values are generated in (1400, 2048,

52) format and reshaped for visualization. Root Mean Square Error (RMSE) is calculated to select the most accurate model for forecasting weather conditions.

**Root Mean Squared Error (RMSE) was used to evaluate the performance of the model.**

**Experimentation Results with Different Hidden Layer Sizes shown in Table 2:**

| **Model No.** | **Layers (Hidden dim)** | **Loss in training** | **Error with ML Model** |
|---|---|---|---|
| 1 | 128 | 0.7703 | 0.8807 |
| 2 | 256 | 0.7659 | 0.8748 |
| **3** | **512** | **0.7602** | **0.8713** |
| 4 | 1024 | 0.7593 | 0.8738 |

As the error of model 3 was the lowest, this model was taken for training. Extensive experimentation reveals that employing a hidden layer size of 512 and training over 250 epochs yield optimal results, ensuring superior accuracy in weather forecasting.

## 5. Results and Discussion

### Root Mean Square Error (RMSE)

RMSE is a key metric for evaluating predictive models, including weather forecasting. It calculates the average magnitude of errors between predicted and actual values using the formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(P_i - 0_i)^2}{n}}$$

where, $P_i$ is Predicted values and $O_i$ is Ground Truth.

**Interpretation:**

**Lower RMSE** values indicate better model performance, as predictions align more closely with observed values.

**Higher RMSE** values signify poorer performance, indicating greater discrepancies between predicted and actual weather conditions.

**Result 5.1**

**Pressure Level Analysis**

Figure 4 illustrates RMSE against pressure levels for temperature and humidity predictions (subplots (a) and (b)). The Graph Neural Network (GNN) model consistently outperforms the persistence model, displaying lower RMSE values across pressure levels. Subplots (c) and (d) depict U_wind (zonal wind) and V_wind (meridional wind) predictions, with the GNN model demonstrating superior performance, especially at higher pressure levels. These findings underscore the GNN model's effectiveness in weather prediction, offering better accuracy compared to the persistence model. This suggests its potential to enhance forecast accuracy and decision-making across various sectors relying on weather predictions.

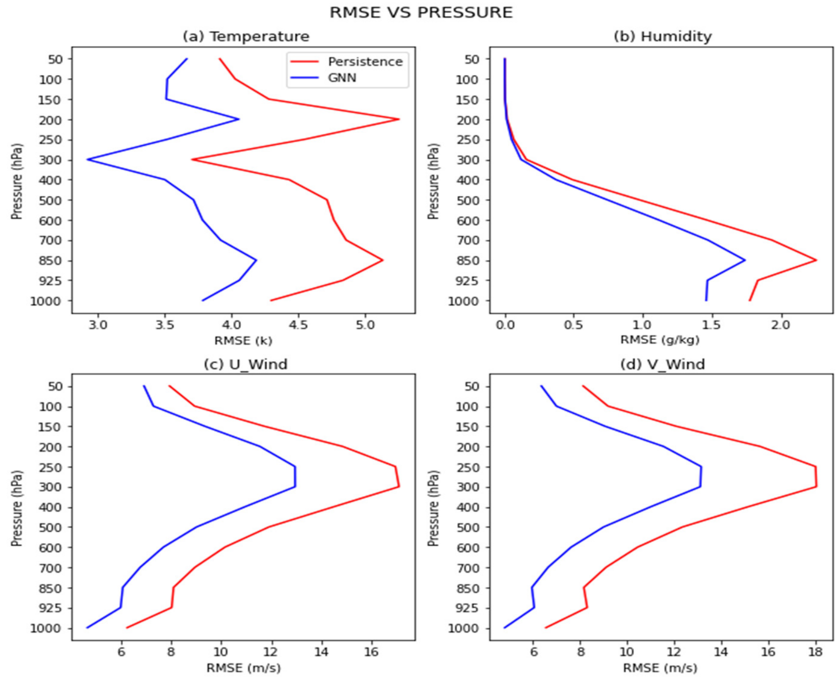


Figure 4: Line Plot of RMSE vs Pressure

**Result 5.2**

**Temporal Analysis**

Figure 5 presents a temporal analysis of Root Mean Square Error (RMSE) values plotted against time for key weather variables. Covering the period from January 1, 2017, to December 31, 2018, at 850 hPa pressure level, each subplot compares RMSE values from the persistence model (red line) with those from the Graph Neural Network (GNN) model (blue line).

Subplots (a) and (b) for temperature and humidity respectively, consistently show lower RMSE values for the GNN model, indicating its superior predictive performance over the persistence model throughout the period. Similarly, subplots (c) for U_wind and (d) for V_wind also demonstrate the GNN model's superiority, particularly in capturing wind dynamics accurately. Overall, the figure highlights the GNN model's robustness in capturing complex weather patterns and dynamics, offering valuable insights for improving forecast accuracy across various sectors reliant on weather predictions.

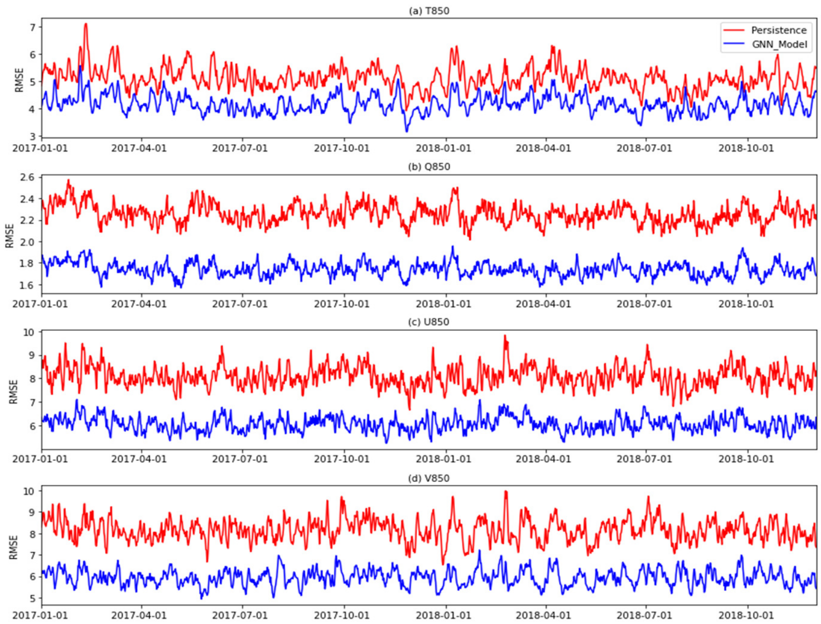


Figure 5: Line Plot of Time vs RMSE

**Result 5.3**

**Synoptic Analysis**

**5.3.1 Spatial Plots of all variables at 850 hPa**

Figure 6 offers a spatial comparison of prediction errors for four atmospheric variables at 850 hPa, across eight subplots in a 4x2 grid. Each subplot contrasts the persistence model with the Graph Neural Network (GNN) model for a specific variable.

The X-axis represents longitude (0-360 degrees), and the Y-axis depicts latitude (-90 to 90 degrees). colour intensity reflects prediction errors, with higher intensity indicating greater variability.

Consistently, the GNN model's graphs exhibit lower color intensity than the persistence model, suggesting superior predictive accuracy across all variables. This pattern underscores the GNN model's enhanced performance in predicting atmospheric conditions at 850 hPa.

Overall, these spatial analyses offer valuable insights into the predictive capabilities of the GNN approach, aiding in the evaluation and refinement of weather forecasting models.

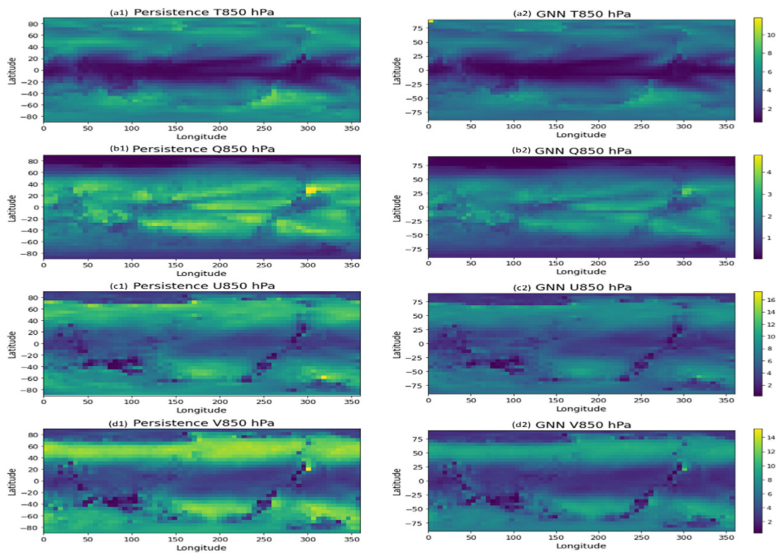


Figure 6: Spatial Plots of all variables at 850 hPa

### 5.3.2 Spatial Plots of all variables at 200 hPa

Figure 7 offers a spatial comparison of prediction errors for four atmospheric variables at 200 hPa, across eight subplots in a 4x2 grid. Each subplot contrasts the persistence model with the Graph Neural Network (GNN) model for a specific variable.

The X-axis represents longitude (0-360 degrees), and the Y-axis depicts latitude (-90 to 90 degrees). Colour intensity reflects prediction errors, with higher intensity indicating greater variability.

Consistently, the GNN model's graphs exhibit lower color intensity than the persistence model, suggesting superior predictive accuracy across all variables. This pattern underscores the GNN model's enhanced performance in predicting atmospheric conditions at 200 hPa.

Overall, these spatial analyses offer valuable insights into the predictive capabilities of the GNN approach, aiding in the evaluation and refinement of weather forecasting models.

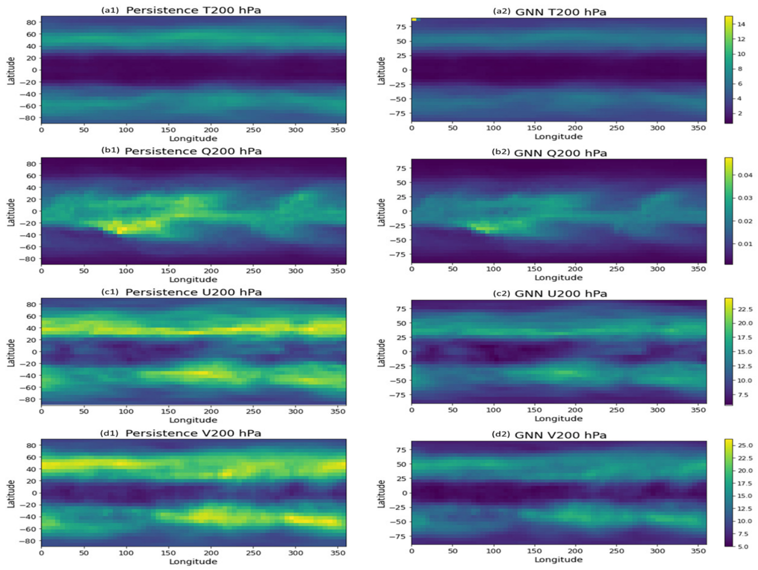


Figure 7: Spatial Plots of all variables at 200 hPa

**Result 5.4**

**Density Plot**

A density graph is a graphical representation used to visualize the distribution of data points in a two-dimensional space. In the context of weather prediction or any predictive modeling task, a density graph can provide insights into the relationship between the predicted values, observed values, and the features used for prediction.

**Spread and Variability:** The width of the curve at any point represents the density or probability of observing values around that point. A wider curve indicates higher variability or spread in the data, while a narrower curve suggests lower variability.

**Correlation:** The density curve can provide information about the correlation between the two variables. In regions where the density is higher and the curve is denser, it suggests a stronger relationship or correlation between the variables. Conversely, in regions where the density is lower and the curve is sparser, it indicates a weaker or no correlation between the variables.

**Result 5.4.1**

**Density Plot of temperature at 850 hPa**

In the density plot, a yellow oval shape indicates a higher concentration of accurate predictions, while the transition to green, blue, and dark blue signals decreasing accuracy. Figure 8 (A) displays the persistence model's predictions for temperature at 850 hPa, showing a spread-out blue color, suggesting more deviations from true values and lower accuracy. In contrast, Figure 8 (B) illustrates the GNN model's predictions, with a more concentrated yellow color, indicating closer predictions to true values with less variability. Performance metrics reveal that Figure 8 (B) has a lower RMSE (4.19 K) and higher correlation (0.96), indicating superior performance compared to Figure 8 (A), which has a higher RMSE (5.13 K) and lower correlation (0.95). These metrics quantify the accuracy and reliability of each model's predictions, with the GNN model demonstrating better performance in predicting temperature values at 850 hPa.

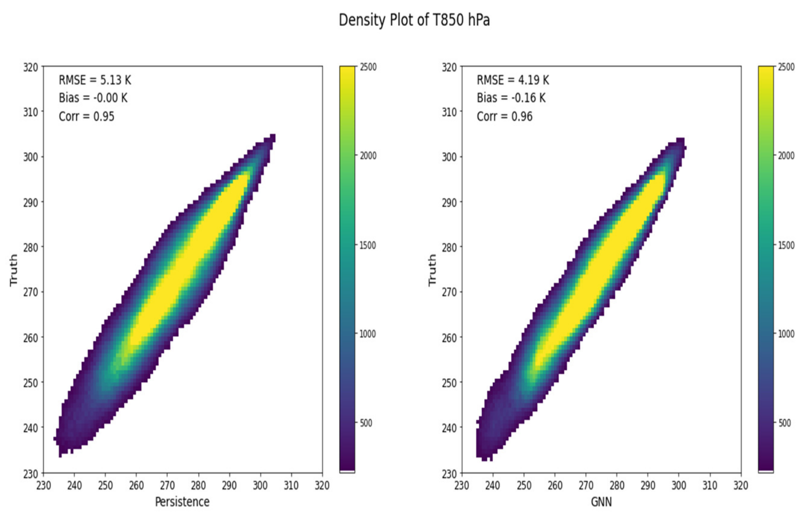


Figure 8: Density Plot of temperature at 850 hPa

**Result 5.4.2**

**Density Plot of humidity at 850 hPa**

Figures 9 (A) and (B) show humidity at 850 hPa, with color intensity denoting data point density. Blue indicates lower density, while yellow indicates higher density. In Figure 9 (A), representing persistence vs. truth, a spread-out blue color suggests less concentrated humidity predictions, while a central yellow point indicates higher density and possibly more accurate predictions. In Figure 9 (B) for the GNN model vs. truth, the yellow color is more spread, indicating higher data point density and more concentrated predictions around true values compared to persistence. The blue color spread is less, indicating higher density and better concentration around true values than in (A).

Performance metrics for Figure 9 (A) the RMSE (Root Mean Square Error) is 2.26 g/kg, the bias is 0.00 g/kg, and the correlation is 0.85. Figure 9 (B), the RMSE is 1.74 g/kg, the bias is 0.00 g/kg, and the correlation is 0.91. In this case, the lower RMSE and higher correlation values for figure (B) indicate better performance of the GNN model compared to the persistence model in predicting humidity values.

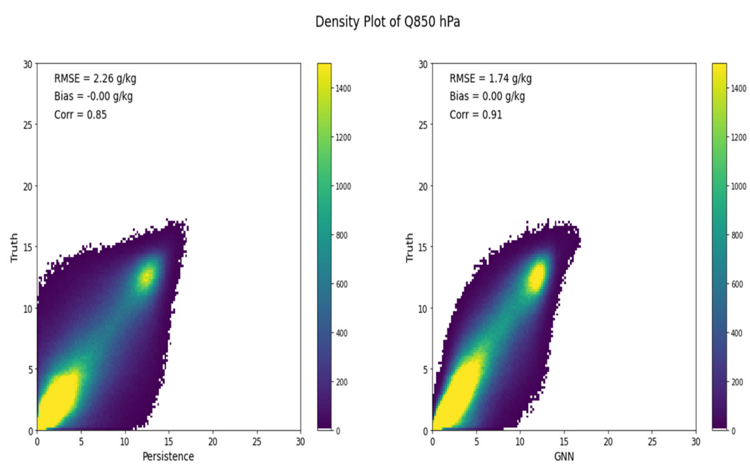


Figure 9: Density Plot of humidity at 850 hPa

**Result 5.4.3**

**Density Plot of u wind at 850 hPa**

Figures 10 (A) and (B) depict predicted u wind values compared to true values for the persistence model and the Graph Neural Network (GNN) model, respectively. In Figure 10 (A), the persistence model's predictions are less concentrated and more scattered, with a central concentration of more accurate predictions. Figure 10 (B) shows the GNN model's predictions to be more concentrated around true values compared to persistence, with reduced blue and yellow colors indicating higher overall data point density and more accurate predictions.

The performance metrics further support these observations:

Figure 10 (A): Has an RMSE of 8.11 m/s, bias of 0.00 m/s, and correlation of 0.52, suggesting relatively low accuracy and correlation compared to the true observed values. Figure 10 (B): Shows improved performance with an RMSE of 6.08 m/s, bias of 0.21 m/s, and correlation of 0.68, indicating better accuracy and reliability of predictions compared to the persistence model.

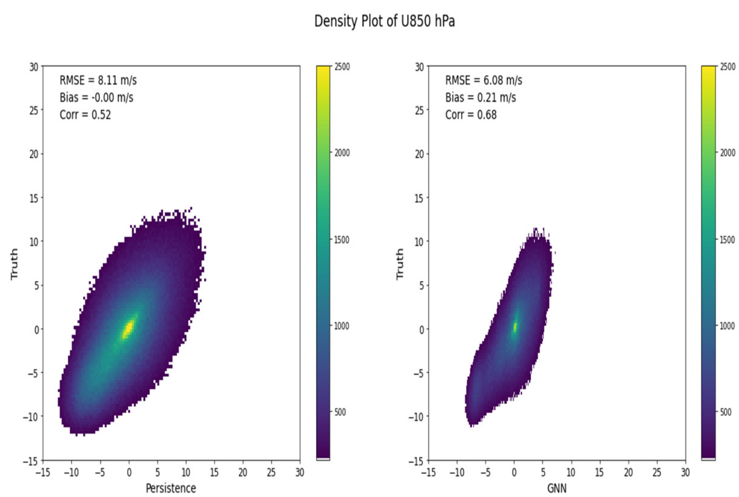


Figure 10: Density Plot of Zonal Wind at 850 hPa

**Result 5.4.4**

**Density Plot of V_Wind at 850 hPa**

In Figure 11 (A), representing persistence vs. truth of V wind, blue color indicates scattered data points with lower density, while yellow signifies a central concentration of more accurate predictions.

Figure 11 (B), depicting the GNN model vs. truth of V wind, shows reduced blue and yellow colors, indicating less concentrated and lower density predictions compared to persistence.

Additionally, let's consider the performance metrics for each figure:

For Figure 11 (A), the RMSE (Root Mean Square Error) is 8.16 m/s, the bias is 0.00 m/s, and the correlation is 0.17. For Figure 5.4.4 (B), the RMSE is 5.95 m/s, the bias is 0.02 m/s, and the correlation is 0.34. These performance metrics provide quantitative insights into the accuracy and reliability of the predictions made by each model. In this case, the lower RMSE and higher correlation values for Figure 11 (B) indicate better performance of the GNN model compared to the persistence model in predicting V wind values.

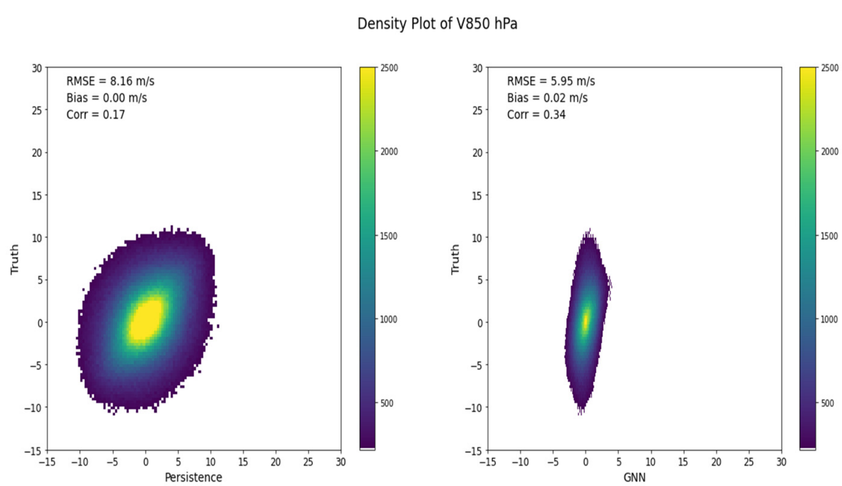


Figure 11: Density Plot of Meridional Wind at 850 hPa

## 6. Conclusion

The comprehensive analysis conducted in this paper, utilizing various visualization techniques, demonstrates the superior performance of the Graph Neural Network (GNN) model compared to the persistence model in weather prediction.

Firstly, the line plot analysis revealed that the GNN model consistently outperformed the persistence model across multiple atmospheric variables, including temperature, humidity, zonal wind, and meridional wind. Spatial plots provided insights into the distribution and variability of prediction errors across different atmospheric variables, with the GNN model demonstrating reduced variability and more accurate predictions compared to the persistence model. Additionally, density plots illustrated the concentration of data points and highlighted areas of improved prediction accuracy with the GNN model.

The analysis of performance metrics further corroborated these findings, with the GNN model consistently achieving lower Root Mean Square Error (RMSE), higher correlation values, and better skill scores compared to the persistence model.

In summary, the combination of line plots, spatial plots, and density plots indicates that the GNN model performs better than the persistence model in weather prediction tasks. These results underscore the enhanced accuracy and reliability offered by the GNN approach, making it a valuable tool for improving forecast accuracy and aiding decision-making in various sectors reliant on weather predictions.

## Reference

1. McGovern, A., Elmore, K. L., Gagne, D. J., Haupt, S. E., Karstens, C. D., Lagerquist, R., & Williams, J. K. (2017). Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bulletin of the American Meteorological Society*, *98*(10), 2073-2090.
2. Ganssle, G., 2018. Node classification by Graph Convolutional Network. https://www.experoinc.com/insights/blog/node-classification-by-graph-convolutional-network

3. Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning-based recommender system: A survey and new perspectives. ACM computing surveys (CSUR), 52(1), 1-38.
4. Rasp, S., Dueben, P. D., Scher, S., Weyn, J. A., Mouatadid, S., & Thuerey, N. (2020). WeatherBench: a benchmark data set for data-driven weather forecasting. Journal of Advances in Modeling Earth Systems, 12(11), e2020MS002203.
5. Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., ... & Battaglia, P. (2022). GraphCast: Learning skillful medium-range global weather forecasting. arXiv preprint arXiv:2212.12794.
6. Keisler, R. 2022. Forecasting Global Weather with Graph Neural Networks. arXiv:2202.07575. Retrieved from https://arxiv.org/abs/2202.07575
7. Ritchie, H., & Roser, M. (2024). Weather forecasts have become much more accurate; we now need to make them available to everyone. Our World in Data. https://ourworldindata.org/weather-forecasts