Weather Data Intelligence

Comprehensive Data Engineering, In-Depth Analysis on Historical Data, and Forecasting on Real-Time Weather Data.

Workgroup 30
Nikhilbhai Laxmanbhai Nakum (0793359)
Yash Rameshbhai Gajera (0810804)
Pratik Sureshbhai Jayani (0810821)

Instructor Name: James Parker
Co-Instructor Name: Robert Sturgeon

Course Code:
AMOD-5610Y-A: Big Data Major Research Paper
Session: 2024FA

Date of Submission:
December 09, 2024

Submitted To:
Department of Applied Modeling and Quantitative Methods
Trent University
Peterborough, ON, Canada

Abstract

This project, titled *"Weather Data Intelligence: Comprehensive Data Engineering, In-Depth Analysis on Historical Data, and Forecasting on Real-Time Weather Data,"* presents an end-to-end cloud-based weather intelligence platform integrating data engineering, data analysis, and data science. Leveraging two distinct data sources—real-time weather data from the OpenWeather API and historical weather data from the National Centers for Environmental Information (NCEI) for New York City's JFK Airport (1990–2023)—we developed a robust data pipeline using Azure Data Factory for ingestion, Azure Data Lake Storage Gen2 (ADLS2) for scalable storage, and Azure Databricks for data processing and transformation.

The pipeline supports seamless ingestion of structured and unstructured data, enabling efficient cleaning, normalization, and transformation at scale. Exploratory data analysis (EDA), performed in R, and visualization dashboards in Tableau revealed critical weather patterns and seasonal trends. For forecasting, a Long Short-Term Memory (LSTM) model was trained using historical data, with performance compared against Seasonal Autoregressive Integrated Moving Average (SARIMA) and the Global Forecast System (GFS), a widely used standard for weather predictions. The LSTM model demonstrated promising results, offering enhanced predictive accuracy for temperature forecasting using key weather indicators.

This project represents a culmination of the skills and knowledge acquired during our master's program, including advanced concepts in data engineering, data analysis, and data science. The system provides a hands-on opportunity to gain insights into how cloud technologies and machine learning models function in real-world applications, enabling practical understanding and skill development.

Table of content

Introduction

The field of data science and engineering has rapidly evolved, with industries increasingly relying on data-driven insights and predictive models to make informed decisions. Recognizing this trend, our team embarked on an in-depth exploration of potential projects that would align with our academic goals and career aspirations. After extensive discussions, we identified a unique opportunity to develop a project that not only integrates all branches of the data industry—data engineering, data analysis, and data science—but also aims to achieve an industry-level mechanism for real-world applications.

Weather forecasting has now become an essential habit for individuals, helping them plan daily activities, whether it's commuting, traveling, or preparing for outdoor events. Reliable and accurate weather insights are critical not only for personal convenience but also for ensuring safety and efficiency in various sectors like transportation, agriculture, and emergency planning. With advancements in cloud computing and machine learning, there is an unprecedented opportunity to enhance forecasting accuracy and deliver meaningful insights in real-time.

This project, titled *"Weather Data Intelligence: Comprehensive Data Engineering, In-Depth Analysis on Historical Data, and Forecasting on Real-Time Weather Data,"* focuses on building a comprehensive, cloud-based weather intelligence platform. The project leverages two primary data sources: real-time weather data from the OpenWeather API and historical data from the National Centers for Environmental Information (NCEI) spanning over three decades for New York City's JFK Airport. A robust data engineering pipeline was developed using Azure services such as Data Factory, Data Lake Storage Gen2, and Databricks to ingest, clean, and process data efficiently. Exploratory data analysis (EDA) was conducted in R, and actionable insights were visualized using Tableau dashboards. For forecasting, machine learning models including Long Short-Term Memory (LSTM) and SARIMA were implemented and benchmarked against the Global Forecast System (GFS).

This project is the culmination of the knowledge and skills acquired during our master's program. It demonstrates how academic concepts can be applied to solve real-world challenges while providing hands-on experience in designing scalable, industry-grade solutions. By integrating all aspects of the data lifecycle, this project serves as a practical example of end-to-end weather intelligence system development.

Literature Review

Weather forecasting and data analysis have become pivotal in understanding atmospheric conditions and facilitating informed decision-making across various sectors. This review examines the research and advancements in data engineering, visualization, machine learning, and data science, emphasizing their applications in weather data processing and prediction. By analyzing these studies, insights are gained into the tools, techniques, and methodologies underpinning modern weather intelligence systems, providing a solid foundation for the development of this project.

2.1 Data Engineering and Weather Data Processing

Data engineering forms the backbone of weather data processing, offering the infrastructure necessary for efficient data ingestion, transformation, and storage. L'Esteve (2021) provides an in-depth exploration of the Azure Data Platform, emphasizing its scalability and flexibility in handling batch and real-time ingestion pipelines. Key components, such as Azure Data Factory, Databricks, and Data Lake Storage Gen2, are highlighted as critical tools for constructing modern ETL pipelines. These tools enable the seamless orchestration of workflows, a feature particularly relevant to managing the complexity of weather data integration. The practical examples presented by L'Esteve deliver actionable insights into optimizing performance, maintaining data integrity, and debugging pipelines, aligning with the goals of this project.

Similarly, Foshin et al. (2024) present a detailed guide on utilizing Azure Data Factory for designing ETL pipelines tailored to big data applications. They underscore the integration of analytical tools, such as Synapse Analytics, and the importance of workflow monitoring through the Azure portal, which are indispensable for ensuring reliability and scalability in weather data pipelines.

Beyond Azure's capabilities, Sinthong and Carey (2019) introduce AFrame, a scalable data analysis package built on Apache AsterixDB, which extends DataFrame operations for large-scale modern data analysis. With its focus on distributed execution and live data ingestion, AFrame emerges as a compelling open-source alternative for handling weather data at scale. Complementarily, Singu (2021) explores the integration of Azure and Databricks for building scalable and fault-tolerant data engineering pipelines. The study emphasizes the role of Databricks' Apache Spark capabilities in enhancing real-time data processing and analytics.

Together, these studies highlight the potential for combining Azure's robust infrastructure with Databricks' advanced analytics tools to address the challenges of large-scale weather data operations.

2.2 Visualization in Weather Data Analysis

Visualization is a crucial component of weather data analysis, transforming complex datasets into actionable insights through interactive and user-friendly dashboards. Joshi and Mahalle (2022) emphasize the significance of storytelling in data visualization, demonstrating how Tableau can be utilized to build compelling dashboards. Their work provides practical guidance on creating visualizations that effectively communicate data trends and relationships. Tableau's ability to integrate context and interactivity into visualizations makes it an ideal tool for weather data analysis, as demonstrated in this project.

Deshmukh and Kharade (2023) investigate the application of Power BI for meteorological data visualization, highlighting its effectiveness in representing complex weather data in accessible formats. While Power BI offers advanced visualization capabilities, its limited compatibility with macOS platforms led to the selection of Tableau for this project. The storytelling capabilities emphasized by Joshi and Mahalle further reinforce Tableau's suitability for designing dynamic and interactive dashboards tailored to weather data.

Visualization in this project serves dual purposes: providing historical insights through interactive filtering and enabling users to search for specific weather data by day and hour. David et al. (2014) discusses similar applications in energy simulation, where historical weather data is visualized to inform building energy efficiency models. This study underscores the importance of detailed and accessible visualizations in decision-making, aligning with this project's dashboard design objectives.

2.3 Machine Learning in Weather Forecasting

Machine learning (ML) has emerged as a transformative tool in weather forecasting, enabling the development of predictive models with enhanced accuracy and efficiency. Soumelidis et al. (2023) examine the role of ML in optimizing weather forecasts by reducing errors in numerical weather prediction (NWP) models. Their findings demonstrate the effectiveness of machine learning algorithms as post-processing tools for refining forecast outputs, a methodology directly relevant to this project's focus on temperature prediction.

Rasp and Thuerey (2021) explore the application of deep learning in medium-range weather forecasting, showcasing how ResNet models can predict weather parameters such as temperature and precipitation. Their research highlights the ability of data-driven approaches to achieve performance levels comparable to traditional physics-based models. By leveraging historical data for model training, deep learning techniques effectively capture complex atmospheric patterns, making them valuable for temperature and precipitation forecasting.

Parasyris et al. (2022) present a comparative analysis of SARIMA and LSTM models for meteorological forecasting, emphasizing the strengths of hybrid models that combine statistical methods with deep learning techniques. Such models leverage the seasonal patterns captured by SARIMA and the nonlinear relationships modeled by LSTM. Banerjee and Mukherjee (2022) validate this approach, showing that LSTM models outperform SARIMA for datasets with higher complexity and longer time horizons.

2.4 Data Science Methodologies and Weather Forecast

Data science methodologies enable the synthesis of insights from large datasets, offering a comprehensive view of weather patterns and trends. Fathi et al. (2022) provide a systematic review of big data analytics in weather forecasting, categorizing approaches into technique-based, technology-based, and hybrid methods. This framework facilitates the evaluation of scalability, accuracy, and efficiency, guiding the selection of appropriate data processing and analysis techniques for this project.

Lagasio et al. (2019) investigate the integration of numerical weather prediction models and Earth observation data to improve precipitation forecasting. Their research demonstrates the potential for high-resolution forecasts, enhancing situational awareness in extreme weather events. Similarly, Randriamampianina et al. (2019) emphasize the role of data assimilation in Arctic weather forecasting, illustrating the importance of observational data in improving initial conditions for NWP models.

Finally, Guoqiang and Ning (2022) propose a hybrid SARIMA-LSTM model for air temperature forecasting, which combines statistical and deep learning approaches to improve accuracy. By decomposing temperature series into trend, seasonal, and residual components, this model captures both linear and nonlinear patterns, offering a robust solution for time-series forecasting. This hybrid approach aligns with the project's goal of developing advanced predictive models that integrate multiple methodologies.

Dataset Description

This section describes the two primary data sources used in the project: real-time weather data from the OpenWeather API and historical weather data from the National Centers for Environmental Information (NCEI). Both datasets focus on New York City's JFK Airport station (Latitude: 40.6413° N, Longitude: 73.7781° W).

3.1 Historical Data Source: National Centers for Environmental Information (NCEI)

The NCEI provides an extensive repository of historical weather data, spanning decades. For this project, we utilized weather records from 1990 to 2023 for JFK Airport in New York City (Latitude: 40.6413° N, Longitude: 73.7781° W). This dataset was critical for training and validating the machine learning models. The data is available in CSV format and contains daily observations of key weather parameters.

Feature	Description	Unit	Value Range
temperature	Air temperature recorded at the station	°C	-18 to 44
dew_point_temperature	Temperature at which water vapor condenses into dew	°C	-30 to 29
station_level_pressure	Atmospheric pressure measured at station level	hPa	960 to 1046
sea_level_pressure	Atmospheric pressure reduced to sea level	hPa	960 to 1046
wind_direction	Direction of the wind	°	0 to 360
wind_speed	Speed of the wind	m/s	0 to 24
precipitation	Amount of precipitation	mm	0 to 74
relative_humidity	Relative humidity in the atmosphere	%	0 to 124
visibility	Horizontal visibility	km	0 to 73
altimeter	Altimeter pressure setting	hPa	960 to 1046
timestamp	Timestamp of the recorded observation	-	Datetime format
Weather_Description	Description of weather conditions	-	Various weather descriptions like rain, snow,cloudy etc...

Sky_Cover	Sky cover description	-	broken clouds, clear sky etc....

Data Access Link: NCEI Climate Data Online

3.2 Real-Time Data Source: OpenWeather API

The OpenWeather API provides real-time weather data, including temperature, humidity, wind speed, precipitation, and other atmospheric conditions. This data was collected specifically for JFK Airport in New York City (Latitude: 40.6413° N, Longitude: 73.7781° W) to monitor current weather and validate the accuracy of the forecasting models developed in this project.

Example of API Call:

```
https://api.openweathermap.org/data/2.5/onecall?lat=33.44&
lon=-94.04&appid={API key}
```

Example of API response:


```
{
  "dt": 1729814400,
  "main": {
    "temp": 289.19,"feels_like": 288.16,
    "pressure": 1016, "humidity": 50,
    "temp_min": 287.58, "temp_max": 290.4
  },
  "wind": {"speed": 6.17,"deg": 360
  },
  "clouds": {
    "all": 0
  },
  "weather": [
    {
      "id": 800,
      "main": "Clear", "description": "cloudy",
      "icon": "01n"}
  ]
}
```


Data Access Link: OpenWeather API Documentation

3.3 Exploratory Data Analysis

A detailed description of the EDA process, including the visualizations and insights derived, is provided in **Section 4.2.2**. This section elaborates on the methodologies used for EDA, including the specific plots and their interpretation.

Methodology

The architecture of this project is depicted in Figure 1**,** illustrating the seamless integration of Azure Data Factory for ETL processes, Databricks for data transformation, and Azure Data Lake for scalable storage. The pipeline further incorporates Tableau and R for data visualization and exploratory analysis, alongside Azure ML Studio for implementing advanced forecasting models like SARIMA and LSTM.

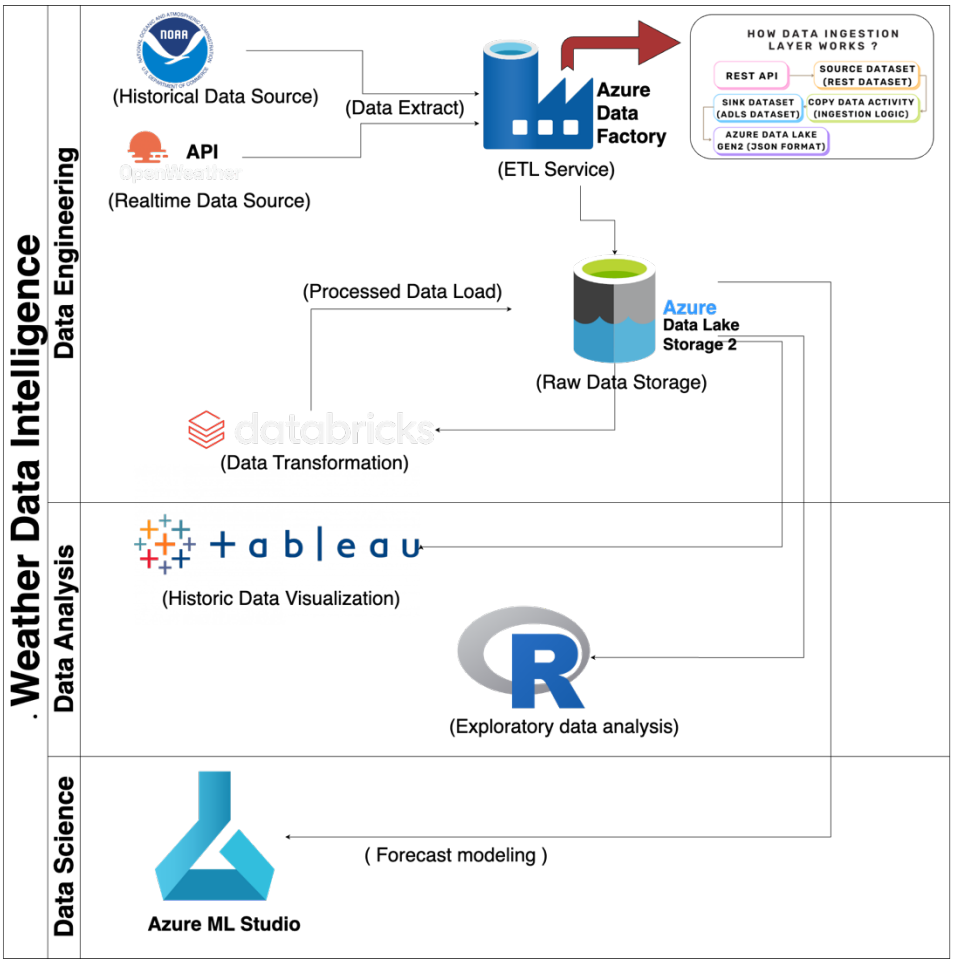


Figure 1: Project architecture and workflow of project

4.1 Data Engineering

The data engineering phase involved designing a scalable, cloud-based pipeline for ingesting, processing, and storing both real-time and historical weather data. Leveraging Azure services such as Azure Data Factory**,** Azure Data Lake Storage Gen2 (ADLS2)**,** and Azure Databricks**,** the pipeline efficiently handled large datasets while preparing them for downstream analysis and modeling.

4.1.1 Data Ingestion

Data ingestion was the initial step in the pipeline, where raw data from both real-time and historical sources was collected and stored in a scalable environment for processing.

1. **Real-Time Weather Data:**
    - Real-time data was sourced from the OpenWeather API, providing hourly updates on temperature, humidity, wind speed, and precipitation.
    - Data was collected specifically for JFK Airport in New York City (Latitude: 40.6413° N, Longitude: 73.7781° W). The API delivered data in JSON format.
    - Azure Data Factory was used to automate the API calls and store the raw data in the `raw-data` directory of Azure Data Lake Storage Gen2 (ADLS2).
2. **Historical Weather Data:**
    - Historical data was obtained from the National Centers for Environmental Information (NCEI), spanning 1990 to 2023.
    - Data was provided in CSV format and included attributes such as temperature, wind speed, precipitation, and pressure.
    - Azure Data Factory pipelines were configured to ingest the CSV files into the `raw-data` directory of ADLS2.
3. **Challenges and Solutions**
    - **Challenge:** Real-time ingestion required handling potential API downtime.
    - **Solution:** Configured retry mechanisms in Data Factory pipelines to ensure seamless ingestion even during transient failures.
    - **Challenge:** Variability in data formats between real-time (JSON) and historical (CSV) data.
    - **Solution:** Preprocessing pipelines were designed in Azure Databricks to standardize data formats during the transformation phase.

4.1.2 Data Processing

The raw weather data was processed to clean, transform, and normalize it for analysis and modeling. This involved handling missing values, deriving additional features, and ensuring data consistency. **Azure Databricks** and PySpark were utilized for distributed data processing, enabling efficient handling of large datasets.

Steps in Data Processing

1. Mounting the Azure Data Lake

   The raw weather data directory in Azure Data Lake Storage Gen2 (ADLS2) was securely mounted to Databricks using OAuth-based authentication. This setup ensured seamless access to data while maintaining security compliance.

2. Data Loading and Attribute Selection

   The raw data was loaded into a Spark DataFrame, and relevant attributes such as temperature, wind speed, and precipitation were selected for analysis.

3. Feature Engineering

   Temporal features like `date`, `hour`, `month`, and `year` were derived from the timestamp to support time-series analysis and seasonal trend identification.

4. Data Cleaning and Imputation

   Numerical columns (e.g., temperature, pressure) with missing values were cleaned using rolling averages, while categorical columns (e.g., weather descriptions) were imputed with the most frequent values within a defined time window.

5. Mapping Encoded Categorical Values

   Categorical values, such as weather condition codes, were mapped to human-readable descriptions to ensure clarity in downstream tasks.

6. Deduplication and Filtering

   Duplicate records were removed by retaining entries with the least missing data for each hourly interval, ensuring data quality and consistency.

This streamlined preprocessing pipeline enabled robust data preparation, forming a reliable foundation for both machine learning models and dashboard visualizations.

Challenges and Solutions

- **Challenge:** Handling missing data in large datasets.
  - **Solution:** Applied rolling averages and majority voting for imputation.
- **Challenge:** Normalizing timestamps from different sources.
  - **Solution:** Converted all timestamps to UTC during transformation.

4.1.3 Data Storage

The processed data was stored in the `processed-data` directory of ADLS2.

The **Parquet format** was chosen for its efficiency in querying and storage. The hierarchical storage structure included:

1. **Raw Data Layer:** Unprocessed files in JSON and CSV formats.
2. **Processed Data Layer:** Cleaned and transformed datasets, stored in Parquet format for analysis and modeling.

Hierarchy: ADLS2 > Container > Raw_Data | Transformed_Data | API_Fetched_Data.

Why ADLS2?

Azure Data Lake Storage Gen2 (ADLS2) was chosen for its scalability, enabling storage of vast datasets with high performance. Its seamless integration with Azure services like Data Factory and Databricks streamlined workflows. Additionally, the Parquet format ensured efficient data compression, reducing storage costs while enabling faster queries for large-scale analytics.

Merits and Challenges of Azure-Based Data Processing Framework

Azure's integration with Databricks provided a seamless and optimized environment for distributed data processing, enabling efficient handling of large datasets while leveraging Spark's powerful data transformation capabilities. Additionally, Azure Data Factory facilitated automated and scalable data ingestion workflows from both real-time and historical sources. Its ability to manage complex workflows with conditional triggers ensured a reliable and scalable data pipeline. Azure Data Lake Storage Gen2 further enhanced the process by offering secure, high-performance,

and cost-efficient storage for both raw and processed data. The hierarchical namespace feature of ADLS2 allowed efficient data organization and retrieval, making it an ideal storage solution. Moreover, Azure's native support for machine learning pipelines, particularly through Azure Machine Learning, simplified the deployment of forecasting models.

However, some challenges were encountered, particularly in the setup and management of Azure's authentication mechanisms. Implementing OAuth-based authentication for ADLS2 and managing service principals was time-consuming and susceptible to misconfigurations. Additionally, frequent manual updates to secrets and tokens added operational overhead. The pipeline's heavy reliance on Azure-specific tools also posed a limitation. Migrating the system to other cloud platforms, such as AWS or GCP, would necessitate extensive rework, including adapting APIs, rebuilding pipelines, and overhauling authentication logic, thereby restricting the system's flexibility and portability.

Why Azure Was Chosen

Azure was chosen for this project over AWS and GCP based on several key factors. Recommendations from industry experts specializing in cloud solutions emphasized Azure's suitability for building scalable data pipelines, making it a reliable choice. Additionally, the team's prior experience with Azure services facilitated efficient implementation and minimized the learning curve, enabling faster project execution. Azure's integrated ecosystem, including seamless interoperability between Data Factory, Databricks, and Azure Data Lake Storage Gen2 (ADLS2), further simplified the workflow and enhanced operational efficiency. While AWS and GCP are widely recognized, Azure offered competitive features, robust hybrid cloud support, and cost-effective pricing, making it the ideal platform for this project.

4.2 Data Analysis

The data analysis phase aimed to extract meaningful insights from historical weather data for New York City (JFK Airport). This phase involved Exploratory Data Analysis (EDA) conducted in R and the creation of a visually rich and interactive dashboard in Tableau. The goal was to understand key patterns, trends, and anomalies in the data while presenting them in a user-friendly format.

4.2.1 interactive Dashboard

Detailed information described below, and access links are provided in the Appendix.

4.2.1 Part 1: Visualization with Customized Filtering:

To provide an overview of historical weather trends, the dashboard includes several visualizations that users can filter dynamically based on year, month, and day.

Features:

The interactive dashboard offers comprehensive weather insights with various dynamic visualizations. A line chart displays average, maximum, and minimum temperatures, showcasing yearly trends and seasonal fluctuations, with filters for specific time periods. Monthly precipitation patterns are presented through bar charts, highlighting wet and dry seasons, while wind speed correlations with weather conditions like thunderstorms and heavy snow are visualized in horizontal bar charts. A heatmap analyzes yearly average maximum heat indices, identifying extreme heatwaves. Additionally, a Gantt chart compares monthly temperature distributions over the years. Dynamic filters for year, month, and day enable users to explore specific timeframes, enhancing interactivity and usability.

Interactivity:

Users can apply multiple filters to adjust the visualizations dynamically.
For instance:

- Selecting a specific year (e.g., 2000) updates all graphs to show data only for that year.
- Narrowing down further by month (e.g., July) allows users to focus on data for a specific season or timeframe.

4.2.1 Part 2: Historical Data Search by Day and Hour

Purpose:

To provide a detailed view of weather features for specific days and hours, this component allows users to search for any historical data point within the dataset.

Features:

The interactive dashboard provides advanced weather exploration capabilities. Users can filter data by specific dates and hours to view detailed attributes, such as temperature, wind speed, precipitation, humidity, visibility, and weather descriptions. Customized summaries dynamically present key statistics for the

selected time, including maximum and minimum temperatures, average wind speed, precipitation, and overall weather conditions. An hourly breakdown table offers detailed insights into how weather attributes varied throughout a selected day.

Challenges and Solutions

1. **Challenge:** Handling a large volume of data for interactive filters.
   - **Solution:** Data was pre-aggregated for certain visualizations to improve dashboard performance without compromising interactivity.

2. **Challenge:** Representing complex weather relationships visually.
   - **Solution:** Chose visualizations like Gantt charts and heatmaps, which effectively showcase temporal and relational data.

   - **Solution:** Implemented intuitive filters for users to explore specific timeframes without requiring technical expertise.

4.2.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) serves as the cornerstone for understanding and preparing the weather dataset for further analysis and modeling. In this project, EDA was conducted to explore patterns, relationships, and anomalies in the weather parameters, providing essential insights into seasonal and temporal variations. Using the R programming language and the `ggplot2` library.

The first step involved examining the temperature distribution across months using a ridgeline plot (Figure 2). This visualization illustrated the temperature range and density for each month, revealing distinct seasonal trends. Winter months, such as December through February, showed narrower temperature distributions centered around lower values, typically ranging from -20°C to 5°C. Conversely, summer months, such as June through August, exhibited wider distributions, with temperatures often peaking near 40°C. Transitional months in spring and autumn presented intermediate patterns, signifying gradual seasonal shifts. The density curves also highlighted occasional extremes, including heatwaves and cold spells, emphasizing the variability in climatic conditions throughout the year.

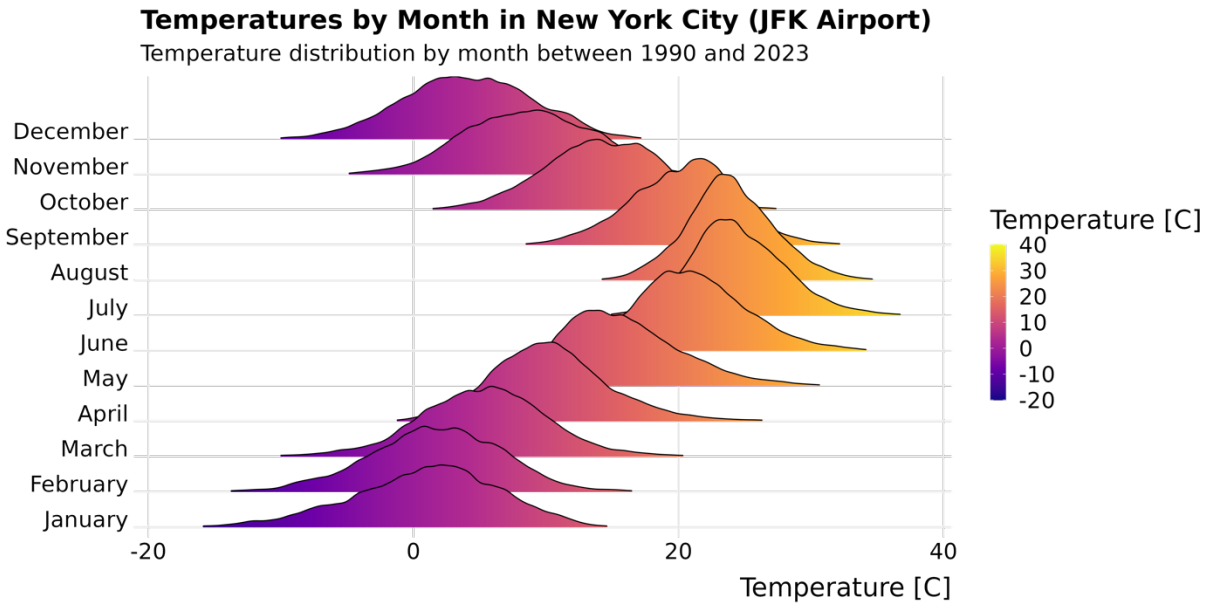


Figure 2: Ridgeline plot of monthly temperature distribution

The relationship between temperature and dew point was further explored using a scatter plot matrix (Figure 3), which grouped data by month and incorporated relative humidity as a color gradient. The scatter plots revealed a strong positive correlation between temperature and dew point across all months. During humid summer months, the data points clustered around higher values for both variables, while in drier winter months, the dew point remained lower even as temperatures varied. The gradient indicated higher relative humidity during summer and lower values during winter, providing a comprehensive view of seasonal atmospheric behavior.

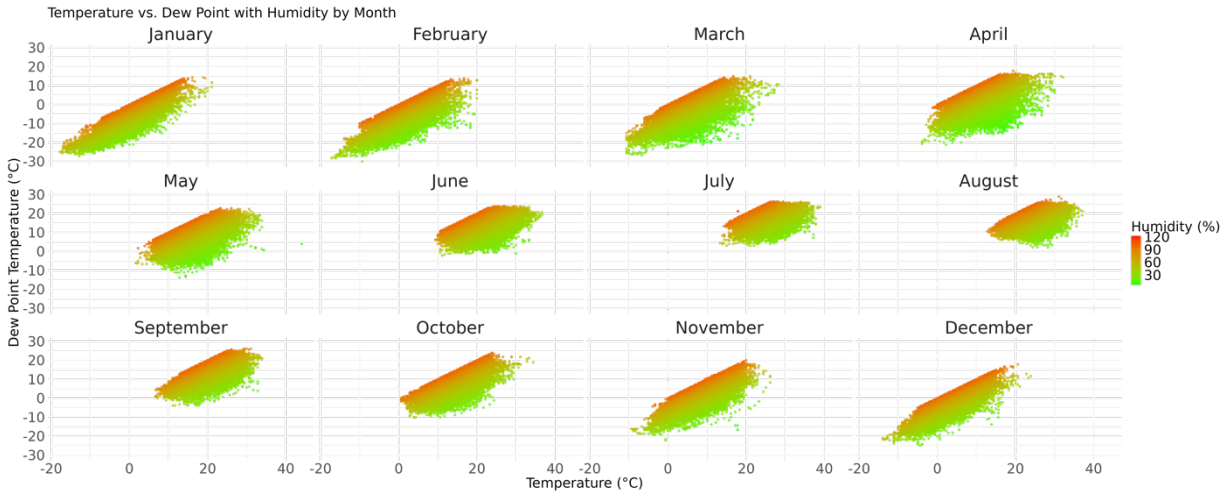


Figure 3: Small multiple scatter plot of temperature vs. dew Point

A circular bar plot (Figure 4) was used to represent wind speed by month, with dew point temperature encoded as a color gradient. This visualization emphasized the cyclical nature of weather patterns, making it easier to observe trends and seasonal peaks. Higher wind speeds were evident in spring and winter months, likely corresponding to seasonal storms and atmospheric disturbances. Dew point temperatures peaked during summer, corresponding to higher humidity levels.

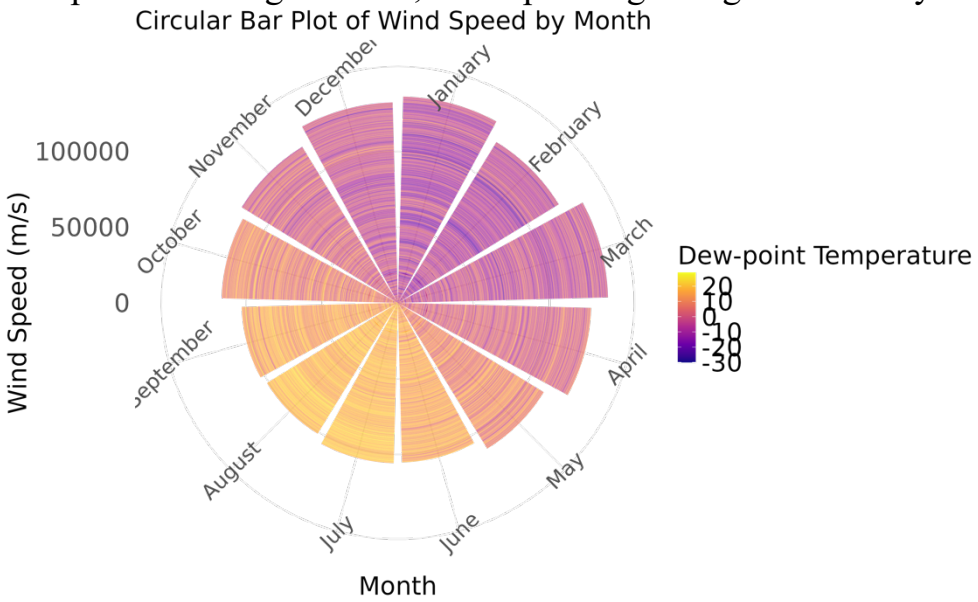


Figure 4: Circular bar plot of wind speed by month.

The interaction between temperature, relative humidity, wind speed, and pressure was visualized using a bubble plot (Figure 5). In this scatter plot, relative humidity was placed on the x-axis and temperature on the y-axis, with wind speed indicated by a color gradient and station-level pressure represented by bubble sizes. The plot highlighted an inverse relationship between temperature and relative humidity, with higher temperatures generally corresponding to lower humidity. Wind speed values, represented by the color gradient, showed no direct correlation with temperature or humidity but were distributed evenly across the plot. Larger bubbles, indicating higher pressure, clustered around lower temperatures and higher humidity levels, consistent with atmospheric dynamics during colder months.

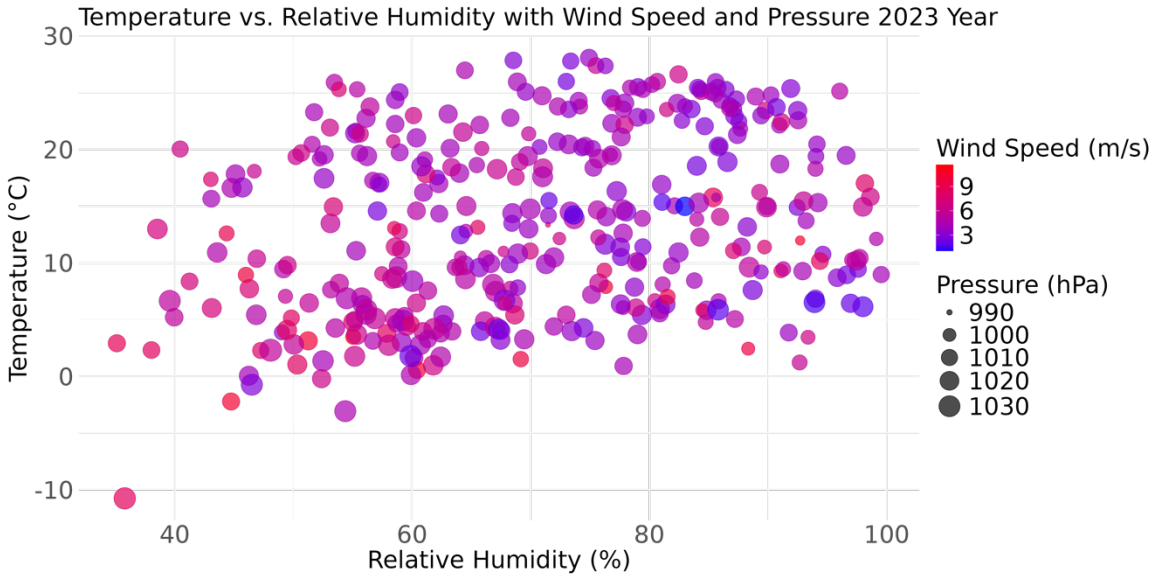


Figure 5: Bubble plot of temperature vs. relative humidity

Finally, a combined bar and line plot (Figure 6) was developed to examine monthly average precipitation and relative humidity. The bar plot represented average precipitation levels, while the line plot overlaid average relative humidity across months. This visualization revealed that summer months experienced the highest precipitation, consistent with monsoonal patterns and wetter weather conditions. Winter months, in contrast, exhibited significantly lower precipitation. Relative humidity remained consistently high throughout the year, with slight increases during summer, further confirming the interplay between rainfall and atmospheric moisture.

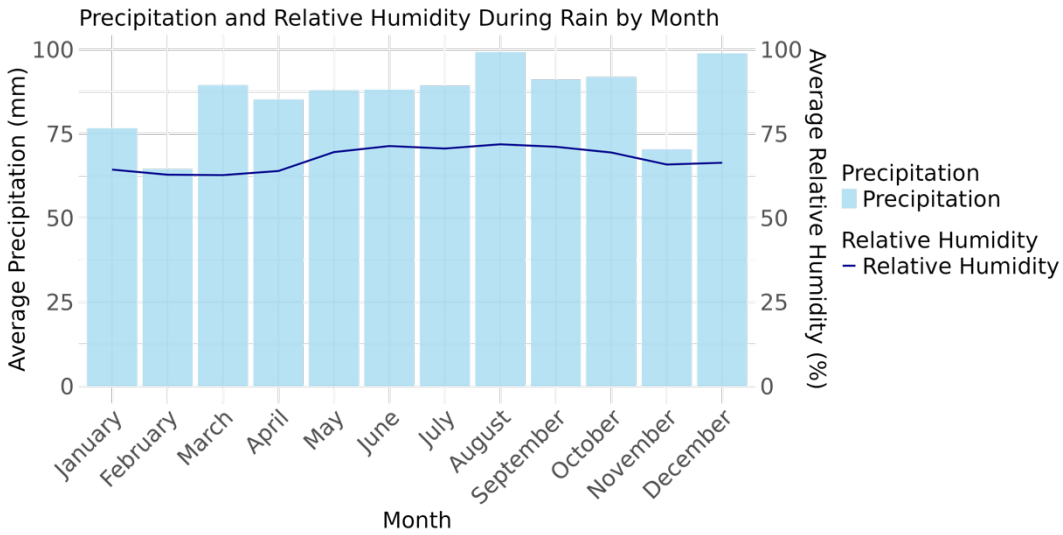


Figure 6: Bar and Line plot of precipitation and relative humidity

Challenges Encountered

The EDA process was not without challenges. One major issue was the presence of missing values in critical columns such as temperature, precipitation, and wind speed. These gaps required careful handling to avoid skewing the analysis. Additionally, the dataset included multiple units and formats, necessitating significant preprocessing to achieve consistency.

Another challenge was computational efficiency. The dataset spanned over 30 years of weather records, comprising millions of rows. This required efficient data handling techniques and software capable of processing large datasets without performance bottlenecks.

Why R for EDA

R was chosen as the primary tool for EDA due to its robust statistical and graphical capabilities. The `ggplot2` library in R provided unparalleled flexibility for creating sophisticated and highly customizable visualizations, such as the ridgeline plot and scatter plot matrix. R's functional programming paradigm also facilitated efficient data wrangling and transformation using packages like `dplyr` and `lubridate`.

Compared to Python's Matplotlib and Seaborn, R's visualization ecosystem often requires less boilerplate code for generating complex plots, saving time and effort. Additionally, the vibrant R community and extensive documentation enabled quicker troubleshooting and adoption of advanced plotting techniques.

4.3 Data Science (Modeling)

The project employed two forecasting models: **SARIMA** (Seasonal Autoregressive Integrated Moving Average) and **LSTM** (Long Short-Term Memory). These methods were selected to compare the effectiveness of statistical models versus neural networks for weather prediction.

The **SARIMA** model, an extension of ARIMA, Before training, the dataset's stationarity was validated using the Augmented Dickey-Fuller (ADF) test, which confirmed the time series as stationary with a p-value close to zero. Given its univariate nature, SARIMA relied solely on temperature data. The hyperparameters were optimized using the **auto_arima** function, which selected an ARIMA order of (4, 0, 1) and a seasonal order of (0, 0, 0, 24), ensuring the model captured daily patterns. SARIMA was trained on daily aggregated temperature data spanning from 2018 to 2022, while testing was conducted on 2023 data. Using Maximum Likelihood Estimation (MLE), SARIMA iteratively refined its coefficients to minimize residuals, but its univariate nature limited its ability to incorporate other variables like wind speed and humidity.

In contrast, the **LSTM** model, a type of Recurrent Neural Network (RNN), was selected for its ability to handle sequential dependencies and capture non-linear relationships in multivariate datasets. The model was trained on multiple features, including temperature, wind speed, and humidity, to learn interdependencies between variables. Before training, features were scaled using **MinMaxScaler** to ensure faster convergence, and the dataset was split into training (80%) and testing (20%) sets while maintaining temporal order. The architecture consisted of a single LSTM layer with 50 units to capture temporal dependencies, a dropout layer with 20% dropout to prevent overfitting, and a dense layer for final predictions. Compiled with the Adam optimizer and a Mean Squared Error (MSE) loss function, the model was trained for 20 epochs with a batch size of 32, leveraging early stopping to avoid overfitting. Validation on unseen 2023 data demonstrated its robust generalizability.

The **choice of LSTM** was driven by its superior accuracy and multivariate capabilities, as it effectively modeled complex interdependencies between variables. Its architecture is well-suited to capturing non-linear patterns, which proved advantageous over SARIMA's linear assumptions. Furthermore, LSTM's performance metrics, including lower Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE), highlighted its forecasting strength.

However, both models faced unique challenges. SARIMA's univariate limitation restricted its ability to utilize additional variables like wind speed and humidity, and

it struggled with abrupt changes and irregular trends due to poor non-linearity handling. On the other hand, LSTM required significantly more computational resources and longer training times, with its performance heavily reliant on careful feature scaling and parameter tuning.

Results

Two forecasting models, SARIMA and LSTM, were implemented and evaluated on historical weather data. The performance of both models was assessed using standard metrics, such as RMSE and MAE, and their results were compared against each other and benchmarked against the established GFS model.

5. 1 SARIMA Model Performance

The SARIMA model served as the baseline, focusing on univariate temperature data. The stationarity of the data was confirmed using the Augmented Dickey-Fuller test, and hyperparameters were optimized using `auto_arima`.

- **Key Results**:
  - RMSE: **3.37°C**
  - MAE: **2.77°C**
  - Confidence intervals were wide, reflecting the SARIMA model's inability to capture abrupt changes in weather patterns effectively.
  - As seen in Figure 7, the predicted values remained close to the mean and failed to respond to high variability in the actual data, particularly during sudden temperature drops or spikes.

This plot (Figure 7) highlights SARIMA's limitations in capturing non-linear patterns, as it predicts smoother temperature trends with wider confidence intervals.

Takeaway: The SARIMA model works well for seasonal, stationary data but struggles to adapt to non-linear relationships and multivariate dependencies. It is not suitable for real-world weather forecasting, where multiple variables and abrupt weather changes play a critical role.

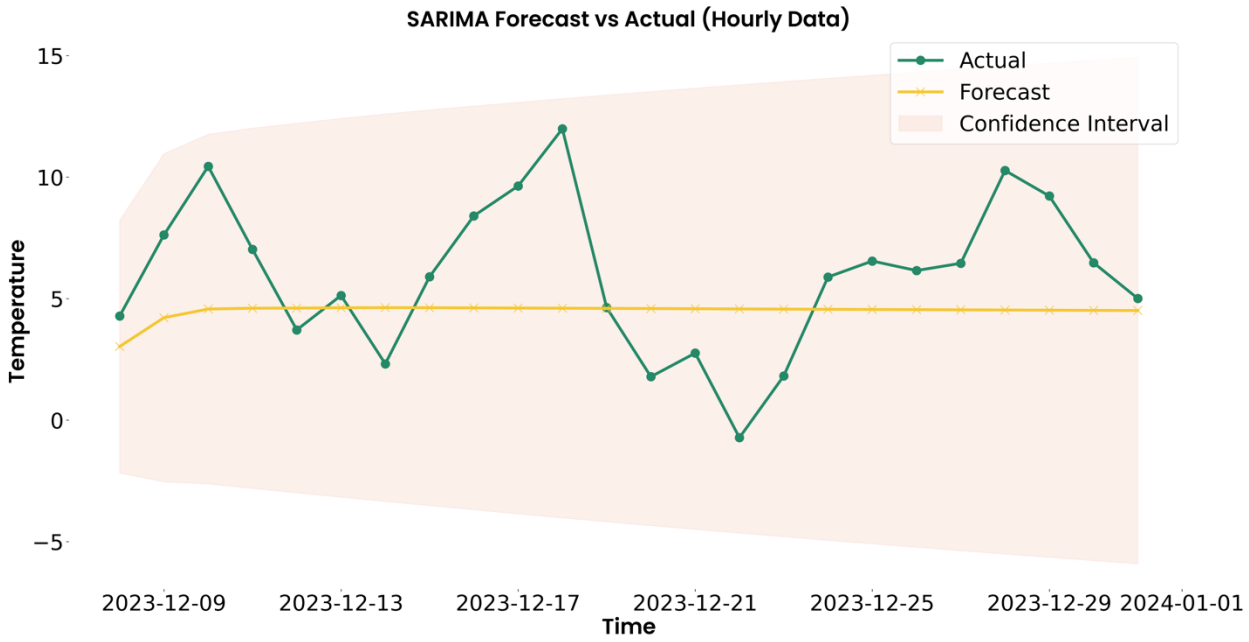


Figure 7: SARIMA performance

5. 2 LSTM Model Performance

The LSTM model was designed to handle multivariate input, including temperature, wind speed, and humidity, enabling it to learn temporal and non-linear dependencies. The model was trained on a limited set of features due to constraints in live sensor data availability.

Key Results:

- RMSE: **0.83°C**
- MAPE: **~6.3%**

Figure 8 clearly demonstrates that LSTM predictions closely follow the actual temperature trends, even capturing fluctuations, which SARIMA failed to achieve.

The LSTM plot (Figure 8) shows significant improvement, capturing real-world variability in temperature trends with high accuracy.

Takeaway: LSTM's ability to model temporal dependencies and non-linear interactions across multiple variables significantly improved forecasting accuracy. This makes it a highly promising approach for short- to medium-range weather prediction.

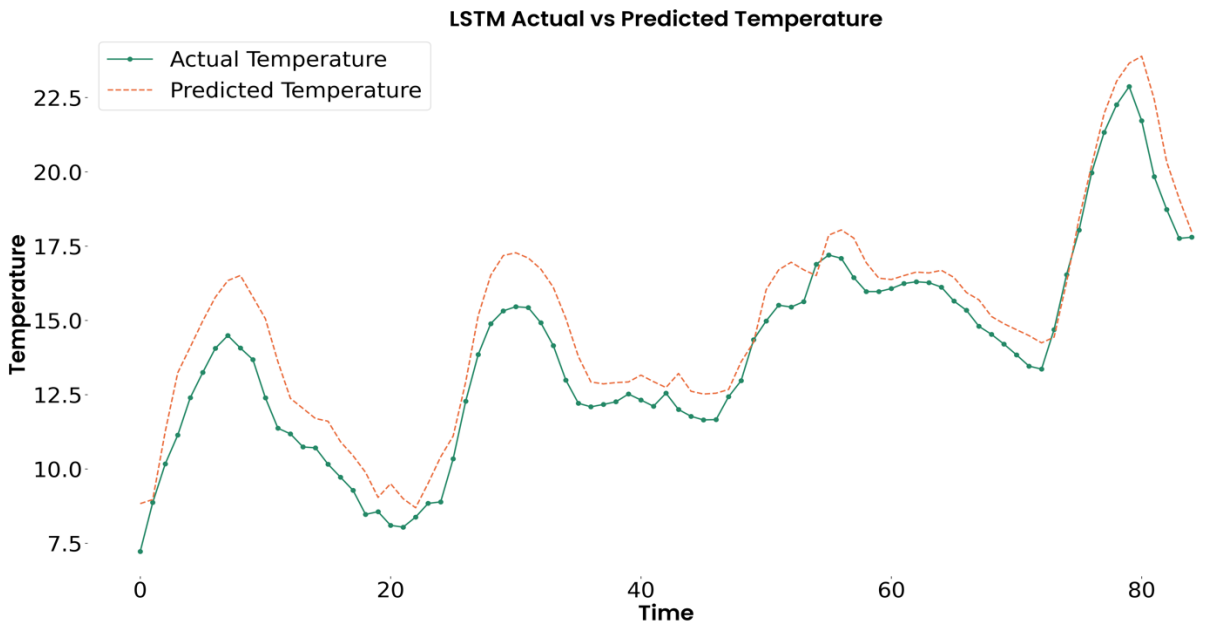


Figure 8: LSTM model performance.

5. 3 Comparison with GFS Model

The LSTM model's performance was benchmarked against the Global Forecast System (GFS), a renowned weather prediction model. While the GFS benefits from live sensor data and atmospheric simulations, the LSTM model achieved comparable accuracy on a smaller scale, showcasing its potential to replicate GFS-level performance when trained on high-resolution multivariate data. However, achieving GFS efficiency posed significant challenges. Unlike GFS, the LSTM model lacked access to real-time sensor data, such as atmospheric pressure, precipitation, and satellite observations, limiting its predictive scope. Additionally, GFS employs advanced numerical simulations and global data assimilation, demanding substantial computational power and sophisticated real-time data pipelines, which were beyond the scope of the LSTM model.

Future Potential: To achieve GFS-like efficiency, the following steps are essential:

- Integration of live, high-resolution sensor data.
- Use of ensemble modeling approaches combining physical models (like GFS) with data-driven models (like LSTM).
- Implementation of distributed deep learning frameworks for faster training and inference.

Discussion and Conclusion

This project demonstrated the seamless integration of data engineering, data analysis, and data science to create a robust and scalable framework for weather forecasting. Using Azure's cloud-based tools and advanced machine learning models, the project tackled the challenges associated with handling large-scale historical weather datasets while achieving meaningful insights and predictions. Each component of the project—data engineering, analysis, and predictive modeling—played a critical role in delivering a comprehensive solution.

The data engineering phase was instrumental in building a scalable and efficient pipeline to preprocess, clean, and store decades of weather data. Using tools like Azure Data Factory and Data Lake Storage, the project ensured the availability of high-quality data for downstream tasks. Challenges such as inconsistent formats and missing data were addressed through robust preprocessing steps, resulting in a well-structured repository of weather attributes, including temperature, humidity, wind speed, and precipitation. However, the reliance on static historical datasets rather than live sensor data limited the ability to capture real-time anomalies, emphasizing the need for future integration of real-time sources.

Exploratory Data Analysis (EDA) revealed critical insights into weather patterns, including seasonal temperature fluctuations, correlations between humidity and precipitation, and temporal trends. Visualizations created using R's ggplot2 library provided an intuitive understanding of the data, uncovering non-linear relationships that justified the use of deep learning models. Despite these insights, EDA faced challenges in visualizing multivariate dependencies over decades, requiring efficient computational and visualization strategies. Nevertheless, it laid a strong foundation for feature selection and model development.

The data science component highlighted the comparative strengths and weaknesses of predictive models. SARIMA, a statistical approach, performed well in capturing linear and seasonal trends but struggled with abrupt changes and complex relationships due to its univariate nature. Its simplicity made it computationally efficient but less accurate, with an RMSE of 3.37°C. On the other hand, the LSTM model excelled in learning non-linear dependencies and handling multivariate data. Training the LSTM on limited features due to resource constraints, it still achieved an RMSE of 0.83°C, closely approximating the accuracy of global weather forecasting models like GFS. However, LSTM required longer training times and higher computational resources, indicating room for optimization in future implementations.

This project demonstrated that accurate weather forecasting requires a blend of advanced modeling techniques and high-quality, real-time data. The LSTM model's performance showed its potential to mimic the accuracy of sophisticated forecasting systems despite limited features and resources. The findings underscore the importance of data quality, multivariate inputs, and the role of deep learning in short- to medium-range weather forecasting. Future efforts should focus on integrating real-time sensor data, exploring hybrid models that combine physical and statistical approaches, and utilizing distributed frameworks to optimize computational efficiency.

In conclusion, the project successfully bridged data engineering, analysis, and science to deliver a practical weather forecasting solution. The results not only advance the understanding of weather patterns but also provide a scalable and cloud-based framework suitable for industrial applications. By leveraging cloud platforms like Azure, the solution demonstrates its adaptability to real-world use cases, enabling integration with live data streams and scalability for handling large-scale weather datasets. This work underscores the potential of deploying machine learning-driven forecasting systems in operational environments.

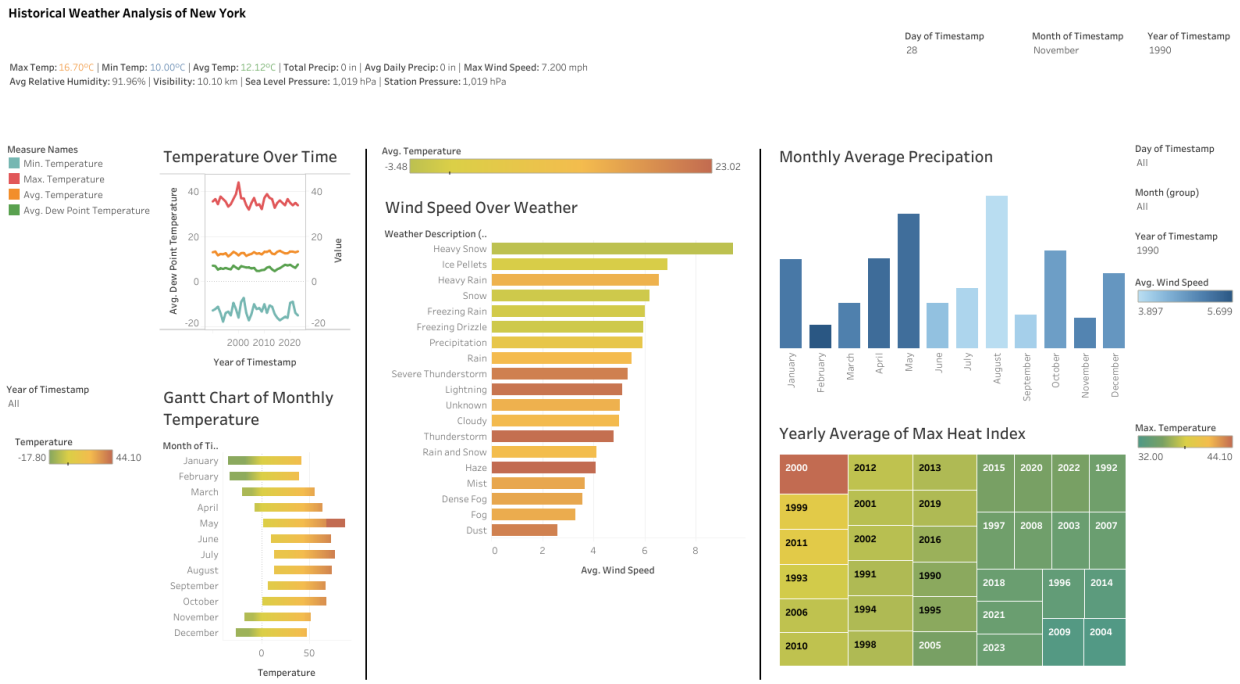
Bibliography

1. R. C. L'Esteve, *The Definitive Guide to Azure Data Engineering: Modern ELT, DevOps, and Analytics on the Azure Cloud Platform*, 1st ed. Berkeley, CA: Apress, 2021. DOI: 10.1007/978-1-4842-7182-7.
2. P. Sinthong and M. J. Carey, "AFrame: Extending DataFrames for Large-Scale Modern Data Analysis," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 359–371. DOI: 10.1109/BigData47090.2019.9006303.
3. X. Lu, D. Shankar, H. Shi, and D. K. Panda, "Spark-uDAPL: Cost-Saving Big Data Analytics on Microsoft Azure Cloud with RDMA Networks," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 321–326. DOI: 10.1109/BigData.2018.8622615.
4. D. Foshin, T. Chernyshova, D. Anoshin, and X. Hertzenberg, *Azure Data Factory Cookbook: A Data Engineer's Guide to Building and Managing ETL and ELT Pipelines with Data Integration*, 2nd ed. Birmingham, UK: Packt Publishing Ltd., 2024.
5. S. Kumar Singu, "Designing Scalable Data Engineering Pipelines Using Azure and Databricks," *ESP Journal of Engineering & Technology Advancements*, vol. 1, no. 2, pp. 176–187, 2021.
6. P. M. Joshi and P. N. Mahalle, *Data Storytelling and Visualization with Tableau: A Hands-on Approach*. CRC Press, 2022. DOI: 10.1201/9781003307747.
7. P. Deshmukh and J. Kharade, "Analyzing Meteorological Weather Data and Visualization Using Power BI," *Indian Journal of Economics and Development*, vol. 21, no. 1, pp. 21–29, 2023.
8. M. David, L. Adelard, F. Garde, and H. Boyer, "Weather Data Analysis Based on Typical Weather Sequences: Application: Energy Building Simulation," in *Proceedings of the IBPSA 2005*, 2014.
9. D. Soumelidis, G. Karoutsos, N. Skepastianos, and N. Tzonichakis, "Optimization of Weather Forecast Data Using Machine Learning Algorithms," *Environmental Sciences Proceedings*, vol. 26, no. 1, pp. 49, 2023. DOI: 10.3390/environsciproc2023026049.
10. S. Rasp and N. Thuerey, "Data-Driven Medium-Range Weather Prediction With a ResNet Pretrained on Climate Simulations: A New Model for WeatherBench," *Journal of Advances in Modeling Earth Systems*, vol. 13, no. 2, 2021. DOI: 10.1029/2020MS002405.
11. A. Parasyris, G. Alexandrakis, G. V. Kozyrakis, K. Spanoudaki, and N. A. Kampanis, "Predicting Meteorological Variables on Local Level with

SARIMA, LSTM and Hybrid Techniques," *Atmosphere*, vol. 13, no. 6, pp. 878, 2022. DOI: 10.3390/atmos13060878.
12. S. Banerjee and S. Mukherjee, "A Comparative Study of Seasonal-ARIMA and RNN (LSTM) on Time Series Temperature Data Forecasting," *Lecture Notes in Networks and Systems*, vol. 317, 2022. DOI: 10.1007/978-981-16-5640-8_25.
13. M. Fathi, M. H. Kashani, and S. M. Jameii, "Big Data Analytics in Weather Forecasting: A Systematic Review," *Archives of Computational Methods in Engineering*, vol. 29, pp. 1247–1275, 2022. DOI: 10.1007/s11831-021-09616-4.
14. M. Lagasio, A. Parodi, L. Pulvirenti, et al., "A Synergistic Use of a High-Resolution Numerical Weather Prediction Model and High-Resolution Earth Observation Products to Improve Precipitation Forecast," *Remote Sensing*, vol. 11, no. 20, 2019. DOI: 10.3390/rs11202387.
15. R. Randriamampianina, H. Schyberg, and M. Mile, "Observing System Experiments with an Arctic Mesoscale Numerical Weather Prediction Model," *Remote Sensing*, vol. 11, no. 8, 2019. DOI: 10.3390/rs11080981.
16. G. Li and N. Yang, "A Hybrid SARIMA-LSTM Model for Air Temperature Forecasting," *Advanced Theory and Simulations*, vol. 6, no. 2, pp. 2200502-2200502, 2022. DOI: 10.1002/adts.202200502.
17. R. B. Alley, K. A. Emanuel, and F. Zhang, "Advances in Weather Prediction," *Science*, vol. 365, no. 6425, pp. 342–344, 2019.
18. D. Bilitza, D. Altadill, V. Truhlik, et al., "International Reference Ionosphere 2016: From Ionospheric Climate to Real-time Weather Predictions," *Space Weather*, vol. 15, no. 2, pp. 418–429, 2017. DOI: 10.1002/2016SW001593.
19. S. Salloum, R. Dautov, X. Chen, P. X. Peng, and J. Z. Huang, "Big data analytics on Apache Spark," *International Journal of Data Science and Analytics*, vol. 1, pp. 145–164, 2016.

Appendix

The dashboard provides an interactive exploration of historical weather data, showcasing key metrics like temperature, precipitation, wind speed, and heat index trends. It includes filtering options for specific time frames and weather conditions, offering actionable insights for weather analysis and prediction.



[Dashboard view]