Problem Statement

Cloudbursts are intense, short-duration rainfall events capable of overwhelming urban drainage systems, causing flash floods, property damage, and even loss of life. Defined by rainfall rates typically exceeding 50 millimeters per hour, these events are particularly destructive in densely populated urban areas with inadequate drainage infrastructure. They are usually caused by rapid convective uplift in the atmosphere, where warm, moist air rises quickly, cools, and condenses into heavy rainfall. These storms are highly localized—often affecting areas just a few kilometers wide—and can occur with little advance notice. Common meteorological precursors include sharp changes in humidity, sudden drops in atmospheric pressure, elevated dew point temperatures, and wind convergence near the surface. As the frequency of extreme weather events increases with climate change, the ability to predict cloudbursts accurately and in a timely manner has become both a scientific and societal priority.

Despite their severe impacts, cloudbursts remain difficult to predict. From a meteorological standpoint, they result from highly localized convective systems, making them challenging to capture with traditional weather forecasting models, especially in regions with sparse sensor coverage. The rarity of cloudburst events also poses a technical challenge for machine learning models, as the data is extremely imbalanced and often noisy. Furthermore, weather data often lacks the temporal and spatial resolution necessary to capture the precursors of such short-lived phenomena.

This project, conducted in collaboration with Alt Surya Inc. as part of the INDRA initiative, aims to support the development of an early warning system for cloudbursts in urban areas. The core objectives are threefold: first, to detect cloudburst events within a 3-hour window; second, to achieve prediction accuracy that surpasses the current 50% baseline in high-risk regions; and third, to enhance disaster preparedness by enabling timely alerts and decision-making. By addressing these goals, the INDRA project seeks to reduce the risks posed by extreme weather events and contribute to more resilient urban infrastructure and response planning.

Literature Review

Recent studies have explored diverse machine learning approaches to forecast extreme rainfall and cloudburst events, offering valuable insights for the development of early warning systems. This section summarizes three representative papers that differ in geographic focus, model architecture, and prediction strategy.

Murakami, H., Sato, T., & Nakamura, H. (2022). *Anomaly Detection of Extreme Precipitation Events Using Deep Autoencoders in Japan*. Journal of Climate Modeling, 45(2), 123–140.

Murakami et al. (2022) applied an unsupervised deep learning model—specifically a deep autoencoder—to detect anomalous precipitation patterns across Japan. Their dataset combined daily precipitation from APHRODITE, atmospheric reanalysis from JRA-55, tropical cyclone tracks from IBTrACS, and simulation data from HiFLOR and SPEAR climate models. The model was trained on historical data from 1977–2015 and evaluated on projections through 2050. Preprocessing included combining sources and standardizing inputs. The autoencoder flagged anomalies by identifying deviations between input and reconstructed outputs based on mean squared error (MSE). While this approach effectively identified trends linked to climate change, the study did not report classification metrics such as precision or recall. Its reliance on daily-resolution data also limits its suitability for short-term applications like the INDRA project. Nonetheless, it demonstrates the usefulness of anomaly detection and ensemble climate data for understanding extreme weather trends.

Patil, R., & Kulkarni, M. (2023). *Short-Term Cloudburst Forecasting Using GFS-Driven Machine Learning Models*. *Advances in Environmental Data Science*, 12(1), 87–101.

Patil and Kulkarni (2023) developed a supervised ML framework to predict cloudbursts using data from 55 Indian weather stations—37 for training and 18 for testing—combined with GFS forecast outputs featuring 50 atmospheric variables sampled at 6-hour intervals. The study covered the 2015 monsoon season and used rainfall from the GHCN database. Preprocessing involved aligning station and forecast data, normalizing features, and handling missing values. Several models were tested, with XGBoost outperforming others. For binary classification of cloudbursts, it achieved 88% accuracy, 0.85 precision, 0.83 recall, and an F1-score of 0.84. A multiclass version also performed well. While highly effective, the method depends on GFS forecasts, which may not be available in real-time for all regions. Moreover, although the dataset and model structure are highly applicable, the temporal resolution of the input data (6-hour intervals) does not match the INDRA project's goal of making predictions within a tighter 3-hour window—making direct reuse of their dataset unsuitable. Still, this study demonstrates a strong modeling framework that can inform future adaptations of INDRA using region-specific and higher-frequency data.

Shani, T., & Nagappan, R. (2024). *Event-Level Classification of Cloudburst Incidents Using Gramian Angular Fields and CNNs*. Proceedings of the International Conference on Hydrometeorological AI, 56–67.

Shani and Nagappan (2024) introduced a novel approach to cloudburst detection by transforming time series weather data into image-like representations using Gramian Angular Fields (GAF), which were then classified using Convolutional Neural Networks (CNNs). Their dataset consisted of real-time weather condition vectors—such as temperature, humidity, and UV index—and focused on current atmospheric snapshots rather than historical sequences. The CNN architecture was chosen for its ability to detect complex, nonlinear features in high-dimensional inputs. Preprocessing involved converting continuous variables into GAF images, allowing the CNN to extract spatial patterns from temporal data. The model achieved 86.4% accuracy, with an F1-score of 0.66, precision of 0.78, and recall of 0.58 for the "burst" class. While the image-based method is innovative, its relatively low recall suggests it may miss many actual cloudburst events—posing a limitation for early warning systems. Nevertheless, the study shows the promise of combining signal transformation with deep learning and may inform future INDRA extensions through hybrid spatial-temporal modeling.

Together, these papers illustrate diverse ML strategies and trade-offs in data, complexity, and performance. Murakami et al. offer insight into anomaly-based climate modeling, Patil and Kulkarni provide a strong supervised baseline using forecast data, and Shani and Nagappan showcase emerging spatial-temporal methods. These studies help shape INDRA's technical scope and clarify performance benchmarks for detecting rare but high-impact weather events.

Data Processing and Feature Engineering

This project uses the NOAA Local Climatological Data (LCD), a publicly available dataset from the National Oceanic and Atmospheric Administration. I selected the Miami International Airport station (ID: 72202012839) as the focal site due to its consistent hourly observations and its relevance to urban flood risk. The dataset includes over 100,000 hourly and daily meteorological records from January 1, 2015, to December 31, 2024, capturing variables such as temperature, dew point, wind speed and direction, pressure, humidity, sky conditions, and precipitation. Its hourly resolution makes this dataset especially suitable for short-term forecasting tasks like cloudburst detection within a 3-hour window.

I began by loading the raw hourly NOAA LCD dataset and selecting relevant meteorological and metadata columns. To ensure data quality, I filtered entries based on report priority, retaining only the highest-fidelity observations in the order: FM-15 > FM-12 > FM-16 > SOD > SOM. Timestamps were parsed into standard datetime format using:

```python
df['Datetime'] = pd.to_datetime(df['Unnamed: 0'])
```

I removed duplicate timestamps and examined missing values across variables to guide our imputation strategy. Handling missing data required variable-specific logic informed by meteorological reasoning. For precipitation, if a missing value was surrounded by hours with zero rainfall, I assumed it was also zero and imputed accordingly:

```python
df['HourlyPrecipitation'].fillna(0, inplace=True)
```

For temperature, pressure, and humidity, I applied linear interpolation for short gaps (typically up to 1–2 hours) to maintain continuity in atmospheric trends:

```python
df['HourlyDryBulbTemperature']=df['HourlyDryBulbTemperature'].interpolat
e(limit=2)
```

Wind-related variables were treated differently: I interpolated wind speed and forward-filled wind direction to retain physically plausible patterns:

```python
df['HourlyWindSpeed'] = df['HourlyWindSpeed'].interpolate()
df['HourlyWindDirection']=df['HourlyWindDirection'].fillna(method='ffill')
```

After preprocessing, the cleaned dataset retained 29 key features and over 87,000 hourly entries, as stored in the final file hourly_data_complete.csv.

Feature Engineering

My feature engineering strategy was guided by meteorological reasoning and the temporal structure of cloudbursts. I defined heavy rainfall events as those where hourly precipitation exceeded the 95th percentile of all observations at the Miami station. This threshold-based approach ensured that the most extreme, impactful events relative to the local climate were focused. To frame prediction as a short-horizon classification task, I constructed forward-shifted binary target variables:

- HeavyRainfall_plus_1h
- HeavyRainfall_plus_2h
- HeavyRainfall_plus_3h
- HeavyRainfall_Next3h (my main label, triggered if any of the next 3 hours exceed the 95th percentile)

These targets were defined using the .shift(-1) to .shift(-3) method:

```python
df['HeavyRainfall_plus_1h']=(df['HourlyPrecipitation'].shift(-1)>p95).as
type(int)
```

While I did not generate traditional lagged meteorological variables (e.g., Temperature_lag1), but engineered a dry period duration feature to reflect antecedent weather. This variable counts how many hours have passed since the last non-zero rainfall and applies a log transformation to compress its wide numerical range:

```python
df['HoursSinceRain_log'] = np.log1p(hours_since_rain)
```

This feature captures the intuition that longer dry spells may increase the likelihood or intensity of a cloudburst, due to atmospheric instability building up over time.

To preserve structural data integrity, I did not impute heavily missing fields like gust speed. Instead, I created binary missingness indicators, allowing the model to learn from the presence or absence of data:

```python
df['HourlyWindGustSpeed_Missing']=df['HourlyWindGustSpeed'].isna().astyp
e(int)
```

Together, these steps translated raw meteorological observations into a structured, model-ready dataset that captures both temporal dynamics and extreme rainfall patterns relevant to cloudburst detection.

One of the primary challenges in my modeling process was the extreme class imbalance—cloudburst-level rainfall events made up less than 1% of all observations. To address this, I implemented class_weight='balanced' in our classifier. Additionally, I excluded entire no-rain weeks from the training data to reduce noise from uninformative periods. Another key challenge was variable-specific missingness across meteorological features. Instead of applying a blanket imputation strategy, I used interpolation or forward fill selectively and created binary missingness indicators to help the model learn from the structure of gaps without distorting physical meaning.

Model Implementation and Evaluation

To predict whether a heavy rainfall event would occur within a 3-hour window, I implemented a supervised learning model using the Random Forest Classifier from scikit-learn. This choice was driven by Random Forest's ability to handle high-dimensional, nonlinear relationships and its robustness to noise and overfitting—critical strengths when dealing with rare weather phenomena like cloudbursts.

I began by partitioning the dataset using a time-based split: 70% of the data was allocated to the training set, 15% to validation, and 15% to the test set. To avoid target leakage, I split the data by weekly blocks, preserving the temporal structure of weather events. Additionally,the training set was filtered to exclude weeks without any precipitation, which helped reduce class imbalance and ensured the model focused on informative weather patterns.

My base model was trained on seven core meteorological features:

```Python
features = [ 'HourlyPrecipitation', 'HourlyDryBulbTemperature',
    'HourlyDewPointTemperature', 'HourlyRelativeHumidity',
    'HourlyStationPressure', 'HourlyWindSpeed', 'HourlyWindGustSpeed'
]
```

Given that only ~0.5% of observations qualified as cloudburst events, I addressed the extreme class imbalance using class_weight='balanced':

```Python
rf = RandomForestClassifier(
    n_estimators=100,
    max_depth=10,
    class_weight='balanced',
    random_state=42
)
```

Initial model evaluation on the test set yielded the following metrics:

- Precision: 0.0545
- Recall: 0.5357
- F1-Score: 0.0989
- ROC AUC: 0.8215
- Confusion Matrix:

```Unset
[[11215  1823]
 [   91   105]]
```

While the model demonstrated relatively strong recall—successfully identifying many cloudburst cases—it also produced a large number of false positives, as indicated by its low precision. To improve this signal-to-noise ratio, I enhanced the model through feature engineering. Specifically, I added three second-order interaction terms:

- Temp_RH = Temperature × Relative Humidity
- Precip_Wind = Precipitation × Wind Speed
- Pressure_RH = Pressure × Relative Humidity

I also introduced a custom feature:

- DryHours_log = log-transformed number of hours since the last precipitation event

These features captured latent relationships in the meteorological dynamics and allowed the model to better distinguish between benign and severe rainfall scenarios.

The enhanced model showed improved performance on all major metrics:

- Enhanced Precision: 0.0675
- Enhanced Recall: 0.5714
- Enhanced F1-Score: 0.1208
- Enhanced ROC AUC: 0.8522
- Confusion Matrix:

```
Unset
[[11491  1547]
 [   84   112]]
```

The rise in precision and F1-score indicates a more confident model with better discrimination between true and false positives. Importantly, DryHours_log emerged as the most influential predictor in the feature importance plot, suggesting that antecedent dryness plays a significant role in cloudburst risk.

Compared to earlier models explored in this practicum—including anomaly detection with autoencoders and basic logistic regression—the enhanced Random Forest consistently outperformed in both interpretability and predictive balance. While deep learning methods may capture complex nonlinearities, they require more data, tuning, and transparency—making Random Forests a more practical choice for rapid prototyping in real-world early warning systems.

In sum, my final model strikes a practical balance between accuracy, interpretability, and operational readiness. The key lessons from this modeling effort were the importance of meteorology-informed feature engineering, thoughtful handling of class imbalance, and time-aware data partitioning. Together, these strategies lay a solid foundation for advancing the INDRA project's real-time cloudburst early warning capabilities.

Ethical Considerations

Cloudburst prediction systems can significantly enhance early warning capabilities, but they also carry ethical implications, especially in high-stakes, life safety contexts. One of the most critical concerns is the tradeoff between false positives and false negatives. False positives—predicting a cloudburst when none occurs—can lead to unnecessary disruptions, economic costs, and long-term public distrust in warning systems. In contrast, false negatives—failing to detect an actual cloudburst—pose far more serious risks, including property damage, injuries, or loss of life, particularly in densely populated or flood-prone urban areas. Given this asymmetry, my modeling strategy prioritized recall to reduce missed detections, while still refining precision to minimize unnecessary alerts. This balance is crucial in maintaining both public trust and response effectiveness.

Though this project used public meteorological data from NOAA with no personal identifiers, privacy concerns could arise in real-world systems if they incorporate location-based or user-submitted data. In such cases, robust data governance practices—such as anonymization, secure data pipelines, and transparency about data collection and usage—are essential. Another important concern is algorithmic transparency and accountability. Machine learning models, particularly ensemble methods like Random Forests, can function as "black boxes," making it difficult for non-expert stakeholders to understand or trust the outputs. Misinterpretation of predictions could lead to either overreliance or unwarranted skepticism. To address this, I recommend integrating explainable AI (XAI) techniques such as SHAP values, openly publishing model validation metrics, and involving meteorological domain experts in system deployment. Ultimately, the ethical deployment of cloudburst prediction tools requires a balanced approach that combines technical performance, clear public communication, and safeguards to ensure responsible and equitable use.

AI Assistants Use and Process Reflection

Throughout this project, I used AI assistants like ChatGPT and Grok as collaborative tools to support my research, coding, and writing. These tools were particularly helpful in speeding up code development and troubleshooting errors.For instance, when designing custom lagged variables or engineering the dry period duration feature, I relied on AI for guidance on logic structure and syntax. During the literature review phase, I used AI alongside Google Scholar—asking ChatGPT to recommend relevant academic papers and summarize them with citation details, which helped me discover additional leads I might have missed.I also used AI to refine the writing in this final report, ensuring clarity and professionalism in explaining technical content.

At every step, I reviewed AI-generated code to ensure it matched my objectives and made logical sense before testing it in Jupyter notebooks. I also cross-checked suggested sources and confirmed that the reasoning aligned with domain knowledge. While AI support improved efficiency, it wasn't flawless—some code outputs contained logical issues, and a few literature references were outdated or inaccurate. These experiences reinforced that AI should be used as a supportive partner, not a substitute, requiring careful validation and human oversight.

Tackling cloudburst prediction introduced me to a novel technical domain that blended meteorology with machine learning. What worked well was breaking the problem into manageable stages—first understanding the data structure, then layering in domain-informed features like dry period duration and interaction terms. A major challenge was the extreme class imbalance, which initially led to misleading performance metrics; I addressed this by adjusting model weights, tuning thresholds, and refining the target variable. I also learned how to handle missing data using variable-specific strategies like interpolation, forward fill, and creating missingness indicators—an approach that preserved the integrity of the weather signals. Most importantly, I gained experience managing a data science project from end to end: from problem definition and data cleaning to model development and final evaluation. Through this process, I developed skills in time-aware modeling, rare event classification, and thoughtful data preprocessing—all of which are transferable to other data science problems involving temporal patterns, noisy data, or imbalanced outcomes.

Conclusion and Future Directions

This project offered a valuable end-to-end learning experience in building a real-world machine learning system for cloudburst prediction. By combining NOAA's high-frequency weather data with domain-informed features such as dry spell duration and interaction terms, I developed a Random Forest model that achieved strong recall and improved precision compared to simpler baselines. A key takeaway was that building an effective model involves far more than tuning for accuracy—it requires rigorous data cleaning, thoughtful feature engineering, and careful interpretation of results. Another important insight was the critical role of time-aware data handling and class imbalance mitigation in rare-event forecasting. Despite the progress made, our model still has limitations: its relatively low precision could lead to false alarms, and training on a single station (Miami) limits its generalizability to other regions. I also encountered challenges due to the complexity of the scientific domain and the difficulty of interpreting meteorological variables without prior expertise. Looking ahead, I recommend expanding the dataset to include multiple locations, integrating higher-resolution sources like radar or satellite imagery, and testing ensemble or threshold-calibrated models to improve reliability. Conducting a deeper literature review at the outset and consulting climate science experts would further strengthen the modeling process. I also see opportunities in more advanced feature engineering to better capture temporal and atmospheric dynamics. These enhancements would improve the INDRA system's ability to deliver accurate, scalable, and operationally ready early warnings for extreme rainfall events.