QMSSGR5055 - PRACTICUM IN DATA ANALYSIS

Final Report

Name: Weixuan (Ariel) Shao

UNI: ws2652

1. Problem Statement

Cloudbursts are intense, short-duration rainfall events—typically over 100 mm within an hour—driven by convective systems, rapid cloud buildup, high humidity, and sudden pressure drops. Their localized nature and brief lead time make them hard to detect, yet their impacts are severe, often triggering flash floods, landslides, and infrastructure damage, especially in vulnerable or mountainous regions. Early prediction is critical but faces technical hurdles such as sparse weather station coverage, difficulty modeling storm-scale dynamics, and extreme class imbalance. The INDRA project aimed to address these challenges by building a machine learning system capable of detecting cloudbursts within a 3-hour window, improving on baseline accuracy, and supporting disaster preparedness through interpretable models and actionable insights.

2. Literature Review

Murakami et al. (2022) used an unsupervised deep learning autoencoder to detect anomalous precipitation events in Japan based on APHRODITE rainfall data, JRA-55 reanalysis, and HiFLOR/SPEAR climate simulations. The model flagged anomalies using reconstruction error and revealed a rise in extreme precipitation linked to climate change. While useful for trend analysis and defining baseline variability, the method lacks precision/recall evaluation and is not suited for short-term cloudburst prediction. In contrast, Kulkarni and Patil (2023) applied supervised learning models—XGBoost and Neural Nets—using GHCN rainfall and

GFS forecast data in India. Their binary XGBoost model achieved strong results (F1 = 0.84, Precision = 0.85, Recall = 0.83), demonstrating the effectiveness of integrating weather forecasts for near-term prediction.

Shani and Nagappan (2024) proposed a CNN-based method using Gramian Angular Fields to transform multivariate weather time series into image-like inputs. Their model achieved 86.4% accuracy and F1 = 0.66 for burst detection, though recall (0.58) remained modest. These three studies offer valuable guidance for the INDRA project: Murakami et al. highlight long-term anomaly detection, Kulkarni and Patil show the power of forecast-driven ML for short-term prediction, and Shani and Nagappan present an innovative architecture that can be adapted for event-level detection despite recall limitations.

3. Data Processing and Feature Engineering

We used 10 years of hourly weather data via NOAA's Local Climatological Data (LCD), which provides high-frequency weather station observations across the U.S., well-suited for short-term cloudburst risk modeling. The dataset includes variables such as temperature, dew point, humidity, wind speed and direction, pressure, sky condition, and hourly precipitation—key indicators for capturing meteorological precursors to extreme rainfall. Data cleaning involved removing duplicates, constructing a complete hourly timeline, and handling missing values using domain-informed strategies: zero-filling, linear interpolation, and forward-filling.

Our feature engineering was grounded in meteorological logic and the temporal structure of cloudbursts. We included lagged features (df[f'{col}_lag1'] = df[col].shift(1)) and rolling statistics (df['precip_3h_sum'] = df['HourlyPrecipitation'].rolling(3).sum()) to capture recent

momentum and cumulative effects. We also derived variables like dew point depression (df['dew_point_dep'] = df['Temp'] - df['DewPoint']), vapor pressure, and wind vector components (df['wind_u'] = df['WindSpeed'] * np.sin(np.radians(df['WindDir']))) to represent atmospheric instability. Time-based features and interaction terms further supported the model in detecting patterns tied to daily cycles or compound effects. Two key challenges were the extreme class imbalance—which we mitigated using class_weight='balanced', SMOTE, and by excluding no-rain weeks from training—and variable-specific missingness, which we addressed using meteorologically sound filling methods to preserve physical meaning and model reliability.

4. Model Implementation and Evaluation

My best-performing model for cloudburst prediction was a Random Forest classifier trained on engineered features. This model effectively handled the class imbalance challenge by applying class_weight='balanced', which boosted the sensitivity to rare events without sacrificing performance on the majority class. Key hyperparameters included n_estimators=100, max_depth=15, and min_samples_split=5, which were tuned based on validation performance. The model input included features such as rolling precipitation sums, pressure and humidity changes, interaction terms, and cyclic encodings. A simplified code snippet of the model setup is:

```
# Train Random Forest with tuned hyperparameters and class balancing
rf_model = RandomForestClassifier(n_estimators=100, max_depth=15,
min_samples_split=5, class_weight='balanced', random_state=42)
rf_model.fit(X_train, y_train)
```

The best Random Forest model achieved a precision of 0.26, recall of 0.40, and F1-score of 0.32 for the cloudburst class, with a ROC AUC of 0.88. While these metrics reflect modest success, they also highlight the limitations imposed by extreme class imbalance, and the model still misses a majority of them. The results underscore a core challenge in rare event prediction: high overall accuracy can mask poor detection of the most critical cases. Compared to logistic regression and the autoencoder, the Random Forest offered better rare-event sensitivity and interpretability, but current performance remains far from ideal. These findings point to the need for enhanced resampling techniques, improved feature selection, and the integration of external predictors to better capture dynamics.

5. Ethical Considerations

A false negative—failing to detect a real cloudburst—can lead to unpreparedness, property damage, or even loss of life, especially in vulnerable communities. On the other hand, frequent false positives may lead to unnecessary panic, resource misallocation, and long-term public distrust in early warning systems. Beyond these risks, ML models introduce other concerns such as overfitting to historical patterns that may no longer reflect changing climate dynamics, or blind reliance on model outputs without understanding limitations or uncertainty. Although our dataset posed minimal privacy concerns since it relied on public weather station data, future integration with higher-resolution or user-generated data may introduce privacy and security risks. To address these challenges, we recommend using conservative thresholds favoring recall, transparent communication about model confidence and limitations, and regular validation with updated data and expert oversight to ensure responsible and adaptive deployment.

6. AI Assistants Use and Process Reflection

Throughout the project, I used AI tools like ChatGPT to support my workflow—from debugging code and designing feature engineering pipelines to summarizing meteorological research papers and clarifying complex domain concepts. These tools were especially helpful when I encountered unfamiliar weather variables or needed to quickly translate theoretical ideas into working code. I verified and refined AI-generated content by testing it in Google Colab, adjusting logic based on performance metrics like recall and F1-score, and applying my domain knowledge to remove physically unrealistic suggestions. I also corrected visualizations when chart outputs appeared misleading. However, AI tools had notable limitations. They often mishandled missing values without considering the context of specific weather variables, sometimes suggesting physically unrealistic approaches like filling wind direction with zeros or interpolating precipitation. They also struggled with rare event detection, often prioritizing overall accuracy rather than recall, and overlooked important strategies such as threshold tuning or resampling to address class imbalance. These limitations highlighted the importance of applying domain knowledge and critically validating AI-generated outputs throughout the modeling process.

What worked well for me in tackling this novel technical domain was combining strong coding fundamentals in machine learning with rapid acquisition of meteorological knowledge. On the technical side, I was comfortable implementing models like Random Forest and Autoencoders, tuning hyperparameters, and engineering time-based features such as lagged variables and rolling statistics. At the same time, I used tools to efficiently digest domain literature—summarizing research papers, extracting relevant predictors, and applying them directly in modeling. A key challenge I faced was dealing with the rarity of cloudburst events, which made most models biased toward the majority class. I overcame this by experimenting with class weighting, filtering the dataset to exclude dry periods, and

optimizing for recall rather than accuracy. Through this process, I developed several transferable skills—designing domain-aware features and validating models using both technical metrics and real-world logic. I also learned to use AI tools critically, integrating them into my workflow while verifying and refining their outputs to ensure accuracy and relevance. Additionally, I became more confident tackling unfamiliar domains by learning to quickly extract key insights from technical literature and apply them to modeling.

7. Conclusion and Future Directions

This project deepened my understanding of how data science can be applied to real-world challenges like extreme weather prediction. Through modeling 10 years of NOAA data, I built and evaluated machine learning models, with the Random Forest classifier achieving the best balance of recall and interpretability for rare cloudburst events. Key limitations include reliance on single-station inputs, domain-specific missing data challenges, and difficulties generalizing to Southeast Asia due to climate pattern differences, coverage gaps, and feature shifts. Future work should incorporate higher-resolution, multi-source datasets, consult climate experts for advanced feature engineering, and explore threshold tuning, ensemble methods, or user-facing alert systems to improve operational readiness for INDRA deployment.

References

Murakami, H., Delworth, T. L., Cooke, W. F., Kapnick, S. B., & Hsu, P.-C. (2022). Increasing frequency of anomalous precipitation events in Japan detected by a deep learning autoencoder. *Earth's Future, 10*(4), e2021EF002481. https://doi.org/10.1029/2021EF002481

Patil, A. A., & Kulkarni, K. (2023). A hybrid machine learning–numerical weather prediction approach for rainfall prediction. In *Proceedings of the 2023 IEEE India Geoscience and Remote Sensing Symposium (InGARSS)* (pp. 1–4). IEEE. https://doi.org/10.1109/InGARSS59135.2023.10490397

Shani, H., & Nagappan, R. (2024). Cloud burst prediction system using machine learning. In *2024 OPCON International Conference on Smart Computing for Innovation and Advancement in Industry 4.0* (pp. 1–7). IEEE. https://ieeexplore.ieee.org/document/10687554