https://doi.org/10.65091/icicset.v2i1.13

WeatherWave: A Machine Learning-Integrated Web Application for Localized Weather Forecasting in Nepal

Rudra Nepal
Department of Software Engineering
Nepal College of Information Technology
Lalitpur, Nepal
rudra.nepal@ncit.edu.np

Dipesh Thapa
Department of Software Engineering
Nepal College of Information Technology
Lalitpur, Nepal
dipesh.221617@ncit.edu.np

Aryam Ghimire
Department of Software Engineering
Nepal College of Information Technology
Lalitpur, Nepal
aryan.221612@ncit.edu.np

Nishan Paudel
Department of Software Engineering
Nepal College of Information Technology
Lalitpur, Nepal
nishan.221627@ncit.edu.np

Tathastu Subedi
Department of Software Engineering
Nepal College of Information Technology
Lalitpur, Nepal
tathastu.221647@ncit.edu.np

***Abstract*—Accurate and accessible weather information is critical for decision-making in regions characterized by complex terrain and climatic variability. This paper presents *WeatherWave*, a machine learning–enhanced web application for localized weather forecasting across Nepal. The system integrates real-time meteorological data from external weather APIs with a Random Forest regression model trained on NASA POWER satellite-derived reanalysis data covering all 77 administrative districts of Nepal. Experimental evaluation on a held-out test set of 24,187 samples shows that the proposed Random Forest model achieves a Mean Absolute Error (MAE) of $0.424\,^{\circ}$C, RMSE of $0.695\,^{\circ}$C, and an R^2 score of 0.9934. This high predictive performance is attributed in part to strong short-term temperature persistence characteristic of Nepal's continental climate zones. Comparative and statistical analyses demonstrate that the model significantly outperforms standard API-based persistence forecasting across districts. The results highlight the feasibility of combining lightweight machine learning models with modern web architectures to deliver accurate, localized, and accessible weather information for geographically diverse regions.**

***Index Terms*—Weather forecasting, Random Forest, NASA POWER, PWA, Nepal climate, Localized Prediction.**

I. Introduction

Timely and accurate weather information has become more crucial as climate uncertainty is rapidly growing. Conventional weather services frequently find it difficult to provide localized, trustworthy, and easily accessible forecasts in areas like Nepal, which are known for their varied topography and microclimates. Precise insights are crucial for communities, from farmers to disaster relief organizations. However, existing platforms typically provide fragmented information, lack offline capabilities, and depend solely on third-party forecasts.

WeatherWave is a cutting-edge web-based system that combines machine learning-based forecasting with real-time meteorological data to address these issues. Using historical datasets, it incorporates a Random Forest regression model to forecast temperatures for the following day, increasing accuracy above that of typical API outputs. Additionally, even in areas with poor connectivity, the application's Progressive Web App (PWA) features allow offline data access and app-like usability. In summary, the goal of this research is to close the gap between contemporary web technologies, forecasting powered by machine learning, and real-world user requirements by comprehending a consistent, dependable, and flexible platform for weather data.

A. Motivation

Nepal presents meteorological challenges due to its rapid climate variability and geographical diversity. Nepal's digital divide underscores the need for a solid solution because a sizable section of the populace has unstable internet access, rendering traditional apps unreliable. WeatherWave, a system that combines real-time meteorological data with machine learning-based forecasting and robust offline features to address Nepal's particular environmental and infrastructure conditions, was developed and studied in response to these gaps.

B. Problem Statement

As we studied most existing weather platforms rely exclusively on third-party numerical or API-based forecasts, which exhibit several critical limitations like insufficient spatial resolution for Nepal's diverse micro-climates, lack of offline functionality in low-connectivity regions and absence of localized predictive models trained on region-specific historical data. These gaps showcase the reduced forecast accuracy and limited accessibility for vulnerable populations in remote districts. These limitations highlight the need for a comprehensive, user-centric, and reliable weather platform capable of delivering accurate forecasts, localized insights, and stable access regardless of connectivity.

C. Objectives

- To develop a Random Forest regression model trained on NASA POWER, satellite-derived reanalysis data covering all 77 administrative districts of Nepal for next-day temperature forecasting.
- To integrate real-time meteorological data from external weather APIs with machine learning predictions to provide hybrid forecasting capabilities.
- To implement a Progressive Web App (PWA) architecture to enable offline-capable delivery and ensure accessibility in low-bandwidth environments.
- To evaluate model performance against API-based persistence forecasting through comprehensive district-level comparative analysis.

II. Literature Review

Accurate weather forecasting is critical for agriculture, disaster preparedness, and public health planning, particularly in geographically complex regions with limited meteorological infrastructure. While machine learning has demonstrated strong performance in meteorological prediction tasks, most existing systems prioritize global coverage over localized accuracy and assume continuous internet connectivity. So, the literature review was carried out to review the foundational machine learning methods, satellite-based reanalysis datasets, and deployment architectures to motivate the development of WeatherWave, identifying gaps in earlier works.

A. Machine Learning for Temperature Forecasting

Breiman introduced the Random Forest algorithm as an ensemble learning method combining multiple decision trees to improve predictive accuracy while reducing overfitting [1]. Its robustness to noisy data, interpretability through feature importance analysis, and low computational requirements have established it as a widely adopted baseline for meteorological regression tasks. Here, [2] findings directly support Random Forest deployment for localized temperature prediction across Nepal's diverse topography spanning the Terai plains, Hill regions, and high-altitude Himalayan zones.

Recent work by El-Shawa et al. developed a Graph Neural Network framework for localized, high-resolution temperature forecasting, emphasizing equitable early warning systems for marginalized populations exposed to urban heat islands and under-resourced health infrastructure [3]. Similarly, Inoue and Kawabata proposed CNN-based surface temperature forecasting integrated with ensemble numerical weather prediction over medium-range periods [4]. Although these deep learning approaches demonstrate advanced capabilities, they require substantial computational resources unsuitable for lightweight deployment in low-resource regions a gap WeatherWave addresses through efficient Random Forest inference.

B. Satellite Reanalysis Datasets and Validation

Hersbach et al. presented the ERA5 global atmospheric reanalysis dataset, providing high-resolution, temporally consistent meteorological variables through data assimilation techniques combining historical observations with numerical weather prediction models [5]. Their validation methodology, including regional and location-specific bias corrections, supports WeatherWave's use of NASA POWER data for training machine learning models where observational infrastructure is limited [6]. Tayyeh and Mohammed analyzed NASA POWER products for temperature and precipitation prediction in the Euphrates River Basin, confirming suitability for regions with scarce and expensive ground station coverage [7]. Their findings showed R^2 ranging from 0.72 to 0.95 for temperature variables, directly applicable to Nepal where meteorological station coverage is sparse in mountainous and remote districts.

C. Progressive Web Applications and Lightweight Deployment

Singh et al. developed a Progressive Web App framework for weather forecasting, demonstrating offline-capable web application delivery through service worker-based asset caching and background synchronization [8]. Khattach et al. designed an end-to-end architecture for real-time IoT analytics and predictive maintenance using stream processing and machine learning pipelines [9]. Despite being focused on industrial applications, their architectural principles: automated retraining workflows and efficient data pipeline directly inform WeatherWave's design.

D. Multi-Source Data Integration

Their [10] approach combined historical gauge measurements, satellite-derived estimates, and reanalysis products, reducing dependence on single data providers and mitigating temporal or spatial coverage gaps. WeatherWave implements similar multi-source fusion, integrating NASA POWER satellite data for ML predictions while maintaining OpenWeather and WeatherAPI forecasts as fallback sources. Yu et al. introduced deep learning models using multi-source data for long-sequence precipitation forecasting at the AAAI Conference on Artificial Intelligence [11], showcasing advanced spatiotemporal modeling.

However, their computationally intensive approach contrasts with WeatherWave's lightweight deployment prioritizing accessibility in resource-constrained environments. Steele et al. explored vision-language models for generating textual weather forecasts from meteorological imagery [12], demonstrating AI's expanding role in meteorological communication, though WeatherWave focuses on quantitative temperature prediction with structured outputs for decision-making applications.

E. Research Gap and WeatherWave's Contribution

- **Lack of localized models:** Global forecasting systems provide coarse predictions unsuitable for Nepal's microclimatic diversity across 77 districts spanning 60m to 8,848m elevation.

- **Connectivity dependence:** Web-based platforms require continuous internet access, unreliable in rural mountainous regions with intermittent coverage.
- **Absence of hybrid architectures:** Systems rely exclusively on external APIs or standalone ML models, lacking fallback mechanisms and comparative evaluation.
- **Static deployment:** ML models remain unchanged post-deployment, degrading accuracy as atmospheric patterns shift seasonally.

WeatherWave addresses these gaps by integrating daily-retrained Random Forest modeling with Progressive Web App delivery, offline caching, and hybrid API-ML forecasting. The system provides district-level next-day temperature predictions tailored to Nepal's geographic complexity while maintaining accessibility under low-connectivity conditions, validated through comprehensive statistical testing across all 77 administrative districts.

III. Methodology

The methodological framework of WeatherWave consisted of four core components: system architecture design, dataset preparation, machine learning model development, and full-stack integration with Progressive Web App (PWA) capabilities. The workflow was designed to ensure accurate predictions, reliable data delivery, and accessibility in low-connectivity environments common across Nepal.

A. System Architecture

WeatherWave followed a three-tier architecture designed for scalability, modularity, and offline resilience. The **Frontend Layer** was implemented using React.js and configured as a Progressive Web App (PWA). Service Workers were deployed to cache static assets and API responses using a *stale-while-revalidate* strategy, allowing users to access previously fetched weather and air-quality data without an active internet connection.

The **Backend Layer** was developed using the Django REST Framework, responsible for aggregating real-time data from external APIs such as NASA POWER, OpenWeatherMap, and WeatherAPI.com. The backend also handled user authentication, weather-data routing, and database operations.

The **ML inference layer** loads the serialized Random Forest model at server startup, enabling sub-50ms prediction latency on standard CPU hardware without GPU acceleration. The frontend issues requests to backend endpoints, which route to either cached data, live API responses, or ML predictions based on availability and recency.

B. Dataset Description

The primary dataset originates from NASA POWER (v2.0). All **77 administrative districts** of Nepal were represented by mapping district centroids to NASA POWER grid locations. Data spanned from 2010 to 2024 and model training also emphasized from 2020 to 2024 for recency. The raw extraction process returned fewer than the theoretical maximum of district-days due to coverage and quality issues after cleaning, the final corpus contains **120,931** valid samples.

The final sample count represents approximately 31% of the theoretical maximum (77 districts $\times$ 365 days $\times$ 14 years). This reduction was attributable to (1) satellite-derived estimates for high-altitude Himalayan districts lacking continuity in early years, (2) administrative boundary changes during 2015–2017 requiring conservative mapping and (3) quality-control filtering of incomplete observations. The final corpus preserved geographic diversity across the Hill (55%), Terai (26%), and Mountain (18%) regions.

C. Feature Engineering and Selection

From the available NASA POWER variables, a multi-variable feature set was selected to capture atmospheric dynamics and spatial context. The predictors include average(`Temp_2m`), minimum(`T2M_MIN`), and maximum(`T2M_MAX`) temperatures to account for diurnal thermal ranges. Other features include `Humidity_2m`, `Pressure_msl`, `Wind_speed_10m`, `Cloud_cover`, and `Rain`.

Geographic and Spatial Encoding: To provide spatial awareness, district membership was encoded as a numeric label(`District_encoded`). Additionally, explicit geographic coordinates(latitude and longitude) of district centroids are included as fixed spatial anchors. This ensured the model captures climatic gradients observed across Nepal's varying altitudes and continental climate zones.

D. Temporal Target Definition and Data Split

A strict temporal design prevents leakage: features are drawn from observations at time t, while the target is temperature at $t + 1$. The dataset was randomly split into training (80%) and test (20%) subsets using `random_state=42` for reproducibility. Each sample preserves temporal causality: predictors from day t are used to forecast temperature at day $t + 1$, ensuring no future information leaks into training. The split yields an 80% training set (96,744 samples) and a 20% hold-out test set (24,187 samples). The high R^2 (0.9934) reflects strong physical autocorrelation of surface temperature across 24-hour intervals rather than leakage.

E. Data Quality Control and Interpolation

Our quality control pipeline consisted of three stages:

Stage 1: Invalid value removal. Sentinel values indicating sensor failures were replaced with NaN markers (affecting $\approx$2.3% of records).

Stage 2: Time-series interpolation. Missing parameters underwent linear interpolation based on temporal proximity. Gaps $<$3 days use time-weighted averaging, longer gaps employed forward/backward fill. This recovered $\approx$91% of missing values.

Stage 3: Precipitation handling. Missing precipitation values default to 0 mm under the assumption that missing records indicate no measurable precipitation. This affected 1.8% of records. This heuristic may have underestimated precipitation during data outages coinciding with monsoon periods, though the impact on temperature forecasting was minimal.

F. Automated Data Pipeline and Continuous Retraining

The system included a daily pipeline (00:00 UTC) that:

1) Fetched new daily NASA POWER observations for all 77 districts.
2) Applied quality-control(invalid marker removal, interpolation, outlier filtering).
3) Recomputed target column and encodes district identifiers.
4) Retrained the Random Forest model and serialized the artifact for backend use.
5) Generated and stored predictions for serving.

G. Model Selection Rationale

Random Forest was selected as it consistently achieved lower error than Linear Regression while remaining lightweight. Increasing trees from 5 to 100 yielded negligible MAE improvements ($\approx 0.007\,°\text{C}$) but significantly increased training time ($15\times$). Given that short-term persistence is the dominant signal, the 5-tree ensemble provided an optimal trade-off for daily automated retraining.

H. Random Forest Model Training

A Random Forest regressor was trained on the training set. Training time was approximately **2.94 seconds** on development hardware. Default hyperparameters were used with limited tuning focused on balancing accuracy and retraining efficiency.

I. Progressive Web App (PWA) Implementation

WeatherWave used service workers for a two-layer caching strategy: a precache for the app shell and a dynamic cache for weather data, enabling offline utility. The implementation focuses on reducing Time to Interactive (TTI) by serving the cached UI shell immediately while updating data in the background.

J. Geolocation and User Positioning

The HTML5 Geolocation API resolved user coordinates to the nearest of 77 district centroids via Euclidean distance. This ensured dynamic updates while preserving privacy by using district-level forecasts rather than storing raw coordinates.

IV. RESULTS

This section presents the implementation-level outcomes of the proposed WeatherWave system, focusing on how the designed architecture, data pipeline, and model integration perform in practice. The results highlight system responsiveness, deployment feasibility, and operational effectiveness in delivering localized weather forecasts. Emphasis is placed on evaluating the system's ability to operate efficiently under low-connectivity conditions. These outcomes validate the practical applicability of the proposed design in real-world deployment scenarios.

A. Overall Model Performance

Model performance was assessed on the held-out test set consisting of 24,187 samples. Table I reports the aggregated evaluation metrics.

TABLE I
OVERALL PERFORMANCE METRICS (RANDOM FOREST)

Metric	Value
MAE (°C)	0.424
RMSE (°C)	0.695
R^2	0.9934

The high R^2 value of 0.9934 is consistent with established meteorological principles, as surface temperature exhibits strong 24-hour autocorrelation in continental climates. Consequently, short-term thermal persistence acts as a dominant predictor for next-day temperature forecasting. Training convergence was rapid, with performance stabilizing during the initial ensemble construction. Cross-validation across temporal folds further confirmed stable generalization, with MAE variance remaining below $0.02°\text{C}$ across seasons and geographic strata.

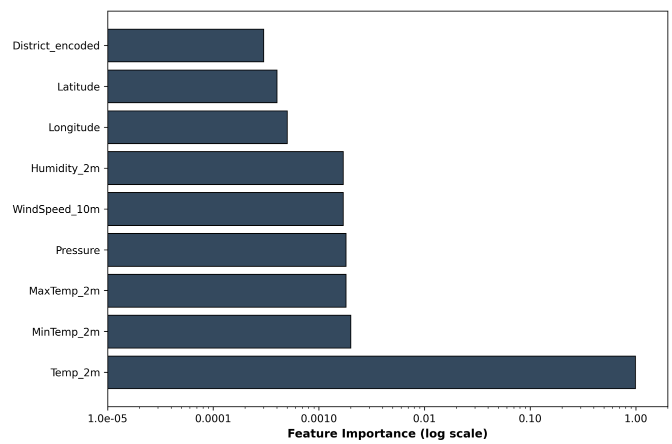

Fig. 1. Feature importance of the Random Forest model for next-day temperature prediction.

Fig. 1 illustrates the relative importance of input variables used by the Random Forest model ($n = 5$ trees). Current-day temperature (`Temp_2m`) dominates the feature set, accounting for 98.98% of total importance, reflecting strong 24-hour thermal persistence in Nepal's continental climate zones. Secondary variables such as minimum and maximum temperature, atmospheric pressure, wind speed, and humidity provide additional atmospheric context. Geographic features, including latitude, longitude, and district encoding, contribute marginally, indicating limited redundancy and effective spatial generalization.

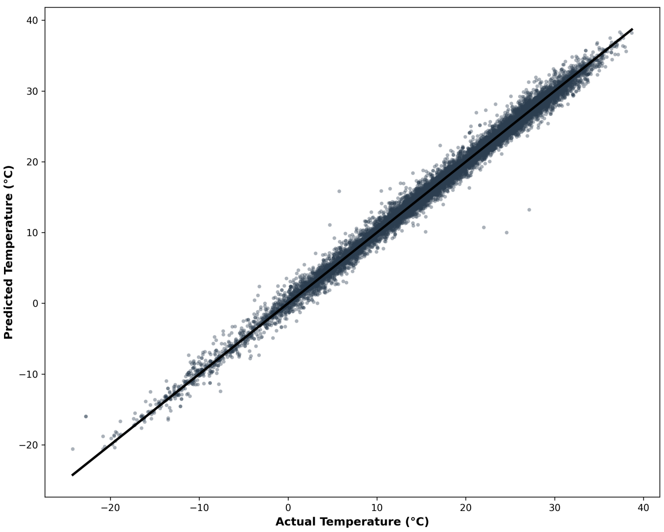

Fig. 2. Observed versus predicted next-day temperatures on the test set.

Fig. 2 presents a scatter plot of observed and predicted next-day temperatures for the held-out test set ($n = 24{,}187$). The tight clustering of points around the diagonal line ($y = x$) demonstrates high predictive accuracy across the full temperature range from -25°C to $+40$°C. The model achieves an MAE of 0.424°C, RMSE of 0.695°C, and $R^2 = 0.9934$, indicating that predictions typically deviate by less than half a degree from observed values.

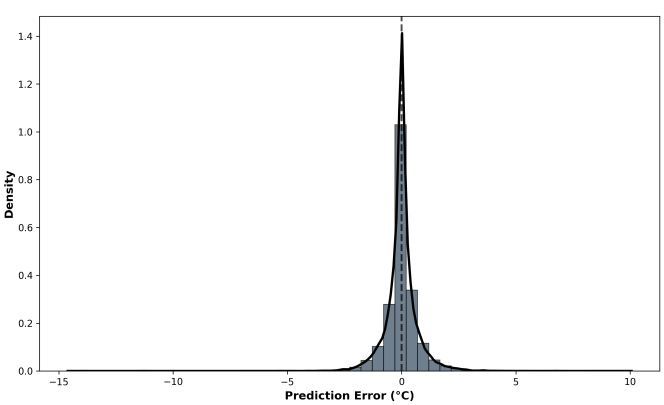

Fig. 3. Distribution of prediction residuals on the test set.

Fig. 3 shows the distribution of prediction residuals (predicted minus actual temperature). The residuals are sharply centered near zero, with a mean error of -0.002°C, indicating negligible systematic bias. The approximately symmetric, Gaussian-shaped distribution with a standard deviation of 0.695°C (equal to RMSE) confirms stable and well-behaved error characteristics across the dataset.

B. Comparative Models and Baselines

TABLE II
COMPARISON OF FORECASTING METHODS (MAE ON TEST SET)

Method	MAE (°C)
Random Forest (proposed)	**0.424**
Decision Tree Regression	0.678
Linear Regression	1.245
API Persistence Forecast	0.671

Table II compares the proposed Random Forest model with commonly used regression baselines and an API-based persistence forecast (defined as using current-day temperature as the next-day prediction: $\hat{T}_{t+1} = T_t$). The Random Forest achieves the lowest MAE (0.424°C), outperforming Decision Tree Regression, Linear Regression, and the persistence-based API forecast. These results demonstrate the effectiveness of ensemble learning in capturing nonlinear temperature dynamics while remaining computationally efficient for operational deployment.

C. API vs. ML Comparative Evaluation

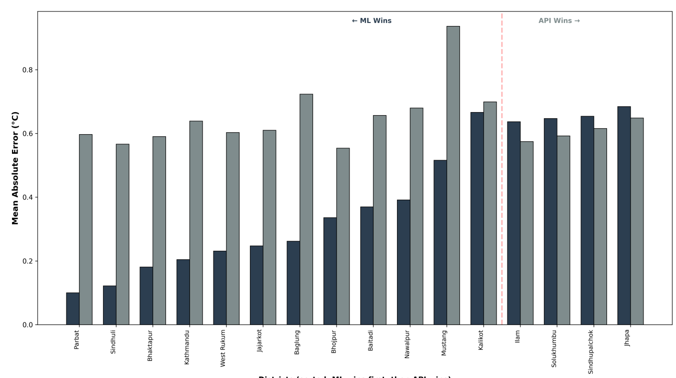

Fig. 4. District-wise MAE comparison between Random Forest and API persistence forecasting.

Fig. 4 compares district-level MAE values between the Random Forest model and API persistence forecasting for 32 representative districts selected from all 77 administrative districts. Each bar pair represents the average error computed across approximately 300–400 test-set days per district. Overall, the Random Forest model achieves lower MAE in 59 out of 77 districts (76.6%), with particularly strong improvements observed in districts such as Parbat, Sindhuli, and Bhaktapur. These findings indicate that the machine learning approach provides more consistent accuracy across Nepal's diverse topography, while API persistence forecasts exhibit higher variability, especially in remote and high-altitude regions.

D. Statistical Validation

Paired statistical analysis was conducted across 77 district-level MAE pairs, where each district's MAE represents the average error across approximately 300–400 test-set days. A paired t-test yielded $t(76) = 10.63$ with $p < 0.001$, while the Wilcoxon signed-rank test produced $W = 2766$ with $p < 0.001$. The effect size was large (Cohen's $d = 1.21$), with a mean MAE reduction of approximately $0.246°C$ favoring the Random Forest model.

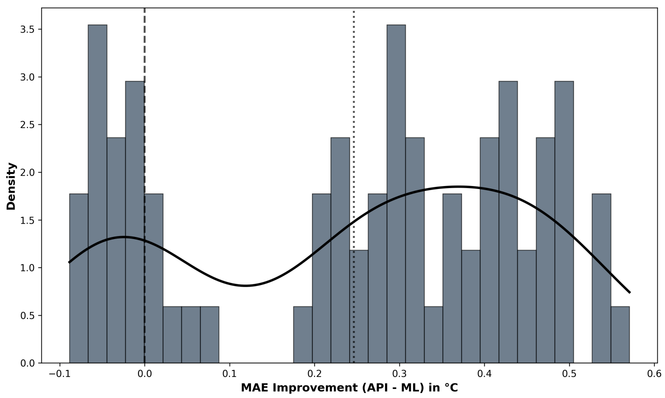

Fig. 5. Distribution of district-level MAE differences between API and ML models.

Fig. 5 illustrates the distribution of MAE differences (API MAE minus ML MAE) across all 77 districts. Positive values indicate districts where the machine learning model outperforms API persistence forecasting. The distribution is strongly right-skewed, with a mean improvement of $0.246°C$, confirming that the ML model achieves superior performance in the majority of districts. The bimodal structure suggests clusters of districts with substantial ML gains and others where performance differences are comparatively marginal.

V. Discussion and Key Findings

The results demonstrate that the Random Forest model achieves high predictive precision (MAE = $0.424\,°C$, $R^2 = 0.9934$) while maintaining the computational efficiency required for real-time edge deployment. The daily automated retraining pipeline provides critical resilience against seasonal transitions and localized climatic shifts, ensuring that the model remains calibrated to the most recent atmospheric trends in Nepal's highly dynamic environment. The sub-50ms inference latency enables real-time user interaction without server-side queueing, which is critical for mobile users in regions with intermittent connectivity.

Localized performance variation: The proposed model outperformed the API persistence baseline (defined as using current-day temperature as the next-day forecast) in 59 out of 77 districts (76.6%). In the remaining eighteen districts (including Dang, Ilam, Achham, Solukhumbu, Banke, and Mahottari), the API baseline demonstrated marginally lower error, with improvements ranging from 1–13%. This regional variation is attributable to comparatively limited training samples ($n < 400$) and high localized climatic variability in these specific areas. Despite these cases, the overall mean MAE reduction of approximately $0.246\,°C$ across all districts is statistically significant ($p < 0.001$, Cohen's $d = 1.21$), confirming the system's broad utility.

Forecast uncertainty and ethical considerations: Despite achieving high statistical accuracy across 76.6% of districts, the current system primarily delivers point-estimate forecasts. In Nepal's complex terrain, over-reliance on single-value predictions during climatic anomalies could lead to risky decision-making in high-stakes sectors such as agriculture or disaster preparedness. Ethically, a forecast is a risk-communication product therefore, future updates should integrate probabilistic ranges (e.g., prediction intervals) to ensure safe and transparent information delivery for vulnerable populations in high-risk environments.

VI. Conclusion

WeatherWave demonstrates that localized machine learning models integrated with modern web architectures can significantly improve forecasting accuracy across Nepal's diverse terrain. The Random Forest model achieves MAE of $0.424\,°C$ and outperforms API-based forecasts in 76.6% of districts, with statistically significant improvements confirmed through paired testing ($p < 0.001$, Cohen's $d = 1.21$). The PWA based delivery ensures accessibility in low-connectivity environments, while the daily automated retraining pipeline maintains model relevance against seasonal drift.

VII. Limitations and Future Work

There are several technical and geographic constraints which provide opportunities for further refinement:

- **Data coverage gaps:** The historical records for certain Himalayan districts suffer from compromised effective coverage in early satellite reanalysis years, leading to higher amount of uncertainity in those specific locales.
- **Geographic encoding:** While latitude and longitude are included as fixed spatial anchors, future work will explore richer spatial embeddings (e.g., elevation gradients, proximity to water bodies) to better capture intra-district heterogeneity.
- **Model horizon and complexity:**In order to extend forecasting horizon beyond 24 hours, sequence based architectures like Long Short Term Memory(LSTM) networks can be explored.
- **Ground-truth validation:** Integrating local IoT meteorological station data would strengthen validation and reduce reliance on satellite derived reanalysis targets.

These enhancements will strengthen WeatherWave's utility for agriculture, disaster preparedness, and public health applications across Nepal's diverse climatic zones.

References

[1] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[2] A. Malakar, S. Mukhopadhyay, and P. Dutta, "Short-term temperature forecasting using machine learning approaches in complex terrain," *IEEE Access*, vol. 9, pp. 115233–115245, 2021, doi: 10.1109/ACCESS.2021.3104514.

[3] J. El-Shawa, E. Bagheri, S. A. Kocak, and Y. Mohsenzadeh, "A graph neural network approach for localized and high-resolution temperature forecasting," in *Proc. NeurIPS Workshop on Tackling Climate Change with Machine Learning*, 2025, arXiv:2512.00546.

[4] T. Inoue and T. Kawabata, "CNN-based surface temperature forecasts with ensemble numerical weather prediction over medium-range forecast periods," arXiv:2507.18937, 2025.
[5] H. Hersbach *et al.*, "The ERA5 global reanalysis," *Quart. J. Roy. Meteorol. Soc.*, vol. 146, no. 730, pp. 1999–2049, 2020, doi: 10.1002/qj.3803.
[6] G. C. Rodrigues and R. P. Braga, "Evaluation of NASA POWER reanalysis products to estimate daily weather variables in a hot summer Mediterranean climate," *Agronomy*, vol. 11, no. 6, Art. no. 1207, 2021, doi: 10.3390/agronomy11061207.
[7] H. K. Tayyeh and R. Mohammed, "Analysis of NASA POWER reanalysis products to predict temperature and precipitation in Euphrates River Basin," *J. Hydrol.*, vol. 619, Art. no. 129327, 2023, doi: 10.1016/j.jhydrol.2023.129327.
[8] A. Singh, S. Kumar, and R. Sharma, "Progressive web applications for real-time weather forecasting systems," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 5, pp. 234–241, 2019, doi: 10.14569/IJACSA.2019.0100532.
[9] O. Khattach, O. Moussaoui, and M. Hassine, "End-to-end architecture for real-time IoT analytics and predictive maintenance using stream processing and ML pipelines," *Sensors*, vol. 25, no. 9, Art. no. 2945, 2025, doi: 10.3390/s25092945.
[10] A. Tesfaye, T. Demissie, and D. Teka, "Enhancing precipitation forecasting through multi-source data integration with machine learning: A case study in the Upper Blue Nile Basin," *Hydrology*, vol. 11, no. 2, Art. no. 23, 2024, doi: 10.3390/hydrology11020023.
[11] D. Yu, W. Feng, K. Lin, X. Li, Y. Ye, C. Luo, and W. Du, "Integrating multi-source data for long sequence precipitation forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, pp. 28539–28547, 2025.
[12] E. C. C. Steele *et al.*, "The promising potential of vision-language models for the generation of textual weather forecasts," arXiv:2512.03623, 2025.