iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VII **Month of publication:** July 2026

DOI: https://doi.org/10.22214/ijraset.2026.84352

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

Weather-Fused CNN-LSTM and GRU Ensemble Learning for Explainable Air Quality Index Forecasting Using the CPCB Methodology

M. Ankitha[1], Pravitha R. Prasad[2]
[1]*Student, Master of Computer Applications (MCA), Department of Computer Applications, Andhra University, Visakhapatnam, Andhra Pradesh, India*
[2]*Project Guide, Department of Computer Applications, Andhra University, Visakhapatnam, Andhra Pradesh, India*

Abstract: ***Air pollution is now one of the biggest environmental and public-health problems facing Indian cities. The Central Pollution Control Board (CPCB) has repeatedly reported that most monitored cities exceed the National Ambient Air Quality Standards for particulate matter, which makes reliable Air Quality Index (AQI) forecasting genuinely useful for public health advisories, traffic control, and simply deciding whether it is safe to step outside. An earlier version of this system predicted AQI from historical pollutant readings alone. It had no weather context, could not respond in real time, and gave no indication of which factor was actually driving a given forecast. This paper builds on that system by fusing raw pollutant concentrations (PM2.5, PM10, NO2, SO2, CO, O3) with weather variables (temperature, humidity, wind speed, atmospheric pressure) and cyclical time-of-day and day-of-week features. We also implement the official CPCB 2014 sub-index methodology ourselves, breakpoint table and max-operator rule included, so that the AQI values used for training and testing are computed rather than assumed from an existing label column. Forecasting is done with a two-branch ensemble: a CNN-LSTM hybrid that first extracts short, local patterns with 1-D convolutions and then models longer sequences with stacked LSTM layers, paired with a separately trained GRU network that is lighter and acts as a second opinion. The two outputs are simply averaged, which cuts down variance compared to relying on one model alone. On top of this, a permutation-importance step tells us which features the ensemble actually leans on when it makes a prediction. Testing on an hourly dataset with a proper chronological (non-shuffled) train/validation/test split, the ensemble reached a Mean Absolute Error of 16.89 AQI points, an RMSE of 23.19, an R^2 of 0.799, a MAPE of 13.19%, and, arguably the number that matters most in practice, a CPCB category accuracy of 82.06% — meaning that more than four out of five hourly forecasts land in the correct health-advisory bracket. Taken together, these results show that a fairly modest set of changes — weather fusion, a two-model ensemble, and a simple explainability layer — noticeably improves both the accuracy and the usefulness of a pollutant-only baseline.***

Keywords: Air Quality Index, CPCB, CNN-LSTM, GRU, Ensemble Learning, Time-Series Forecasting, Explainable AI, Weather Fusion, Permutation Importance, Sliding Window.

I. INTRODUCTION

The Air Quality Index (AQI) is meant to make a messy pile of pollutant readings understandable to an ordinary person — one number, one category, and a rough idea of whether it is safe to go outside. In India, the Central Pollution Control Board (CPCB) defines exactly how that number is calculated, through a standardised sub-index and breakpoint methodology. The AQI on its own only tells you what conditions look like right now, though. Anything useful for planning — issuing an advisory, restricting traffic, telling a factory to cut back for a few hours — needs a forecast, not just a snapshot, and ideally one that arrives before the exceedance happens rather than after. An earlier version of this project forecast AQI directly from historical pollutant concentrations, and nothing else. In practice this created three separate problems. First, it had no access to weather data at all, despite the fact that wind speed and humidity are well-known drivers of how pollutants disperse. Second, it could not tell anyone why it predicted what it predicted, which makes a forecast hard to trust or act on. Third, because it relied on a single recurrent model, it reacted poorly to sudden pollution spikes — the kind of thing an ensemble tends to handle more gracefully. This paper tries to fix all three problems at once, without over-engineering the solution. We fuse pollutant concentrations with weather observations and cyclical time features at the input stage; we build a two-branch CNN-LSTM/GRU ensemble, so the model gets both convolutional pattern extraction and two different flavours of recurrent memory; and we add a permutation-importance step

that at least gives some indication of which feature is driving a given prediction. We also went back and implemented the CPCB 2014 sub-index rules ourselves rather than trusting a pre-existing label column, mainly so that every number reported later in this paper is something we can actually verify against CPCB's own worked examples.

A. *Problem Definition*

Put simply, the baseline system had no weather context, no real-time response, a forecasting horizon that was stuck at daily granularity, poor handling of sudden spikes, and no way to explain itself. That last point matters more than it might seem — a number with no reasoning behind it is not much use to someone who actually has to decide whether to issue an advisory.

B. *Objectives*

1) Fuse pollutant concentrations with weather observations and cyclical time-of-day/day-of-week features at the input layer, instead of treating pollutants as the only signal worth modelling.
2) Implement the CPCB 2014 sub-index methodology from scratch, so the ground-truth AQI used for training and evaluation can actually be checked against CPCB's own examples.
3) Build and train a two-branch CNN-LSTM and GRU ensemble for one-hour-ahead AQI forecasting, combining the two branches through simple averaging.
4) Add feature-level explainability through permutation importance, so that a forecast comes with at least some indication of what drove it.
5) Evaluate everything on a properly chronological test split using MAE, RMSE, R^2, MAPE, and CPCB category accuracy — not just the raw regression numbers.
6) Wrap the whole pipeline in a Stream lit dashboard so it can actually be used, not just benchmarked offline.

II. LITERATURE REVIEW

Statistical time-series models such as ARIMA and its seasonal variant SARIMA [9] were the natural starting point for AQI and pollutant forecasting. They are cheap to compute and easy to interpret, and they do fine on near-stationary, roughly linear data. Air pollution, unfortunately, is rarely that well-behaved — it spikes suddenly with weather changes and depends on long-range meteorological effects that a linear model just cannot represent.

Classical machine-learning regressors — Linear Regression, Decision Trees, Random Forests [10], Support Vector Regression — do somewhat better. Comparative studies on real pollutant data have reported ensemble-tree methods like Extra Trees reaching an MAE around 11.9 and R^2 around 0.87, clearly ahead of linear baselines. The catch is that these models still need someone to hand-engineer lag features to fake temporal awareness; they don't learn sequence structure on their own.

Recurrent networks changed that. LSTM [6] in particular was a real step forward, since its gated memory cells let it retain information across long sequences without the vanishing-gradient problems that plague plain RNNs. Work comparing ARMA, SARIMA, RNN, LSTM, and GRU on Beijing air-quality data found the recurrent models consistently ahead of the statistical ones on this kind of volatile, non-linear series. GRU [7] itself is essentially a simplified LSTM — fewer gates, less compute — and tends to land close to LSTM in accuracy, which is why it shows up so often as the lighter half of a hybrid architecture rather than a full replacement for it.

A lot of recent work pairs 1-D convolutions with recurrent layers, since CNNs are good at picking out short, local patterns (a spike over a couple of hours, say) while recurrent layers are better suited to the longer story. A CNN-LSTM model built for Beijing AQI forecasting beat ARMA, SARIMA, RNN, LSTM, and GRU baselines individually [1], which more or less established CNN-LSTM as a solid default architecture for this problem. A variant called CNN-ILSTM, which tweaks the internal LSTM gating to train faster and more stably, reached an MAE around 8.41 and R^2 around 0.96 on multi-year hourly data from Shijiazhuang, China, beating eight other baseline models [2]. Other groups have pushed further still: a CNN-LSTM-KAN architecture adds Kolmogorov-Arnold Network attention layers for multi-city prediction [3]; an ARIMA-CNN-LSTM hybrid splits the linear and non-linear components of the signal between ARIMA and CNN-LSTM, tuned with a dung-beetle optimiser [4]; and a CNN-Bi-LSTM setup has been used for spatially resolved AQI prediction across multiple zones of a megacity [5].

Closest to what we are doing here is a 2025 study on Visakhapatnam that combined 1D-CNN feature extraction with an LSTM-plus-multi-head-attention stage and a GRU stage for residual correction.

Trained on just over 14,000 hourly records spanning two years and tuned with Bayesian optimisation, it reported an R^2 of 0.9757 and RMSE of 6.29 [8] — noticeably better than what we report here. The gap is not surprising: that work had a much larger, real, location-specific dataset, an attention mechanism, and proper hyperparameter search, none of which we attempted in this iteration. We come back to this in Section VI.

One thing almost none of this literature does well is explain itself. A predicted AQI value, on its own, tells a decision-maker nothing about which pollutant or weather variable actually caused it. One study did pair a CNN-LSTM model with Partial Dependence Plots, SHAP, and LIME for several Indian cities and found PM10 and PM2.5 to be the dominant drivers of AQI movement — a result that, reassuringly, lines up with what we find independently in Section IV. That gap in per-forecast explainability is really the main reason we added the permutation-importance step in the first place.

So, most existing systems — including the one this work started from — treat AQI forecasting as a black box: pollutants in, a number out, no weather, and no explanation. What we present here tries to close that gap with a weather-fused ensemble, a self-computed CPCB ground truth, and a permutation-importance layer bolted on at the end.

Table I. Summary Of Related Work

Approach	Representative Technique(s)	Strengths	Limitations
Statistical	ARIMA, SARIMA [9]	Interpretable, low compute cost	Assumes linearity; weak on volatile series
Classical ML	Random Forest [10], SVR, Decision Tree, Extra Trees	Robust to noise; native feature importance	No sequence modelling; manual lag features
Recurrent DL	LSTM [6], GRU [7]	Captures long-range temporal dependency	No local pattern extraction; opaque reasoning
Hybrid CNN-RNN	CNN-LSTM [1], CNN-ILSTM [2], CNN-Bi-LSTM [5]	Combines local extraction with sequence memory	Higher training cost; still largely black-box
Attention-augmented hybrid	CNN-LSTM-MHA+GRU [8], CNN-LSTM-KAN [3]	State-of-the-art accuracy on large datasets	Requires large data and careful tuning
This work	CNN-LSTM + GRU ensemble with permutation importance	Weather fusion, verifiable CPCB ground truth, per-forecast explainability	Evaluated on a synthetic, CPCB-aligned dataset (Section IV-F)

III. PROPOSED SYSTEM AND METHODOLOGY

The system works in four stages. Preprocessing cleans the raw pollutant and weather readings, adds the cyclical time features, and Min-Max scales everything into 24-hour sliding windows. Ground-truth generation runs the CPCB 2014 sub-index rules to get a verified AQI label directly from pollutant concentrations, rather than trusting whatever label happened to already be in the data. Ensemble forecasting runs the CNN-LSTM and GRU branches independently and averages their scaled outputs. Explainability then quantifies, after the fact, which features the ensemble actually relied on. All of this is exposed through a Streamlit dashboard so it can be used in real time rather than just run offline. Fig. 1 shows how these pieces fit together.

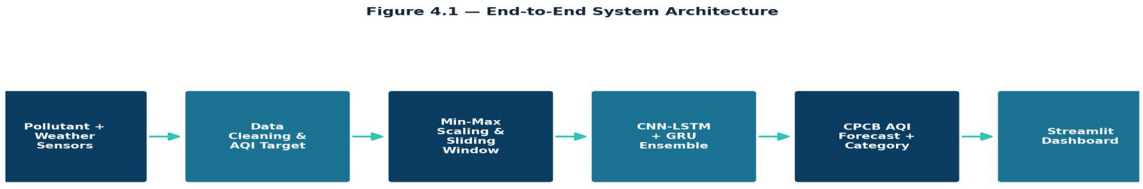


Fig. 1. Overall system architecture, from raw pollutant/weather ingestion through preprocessing, CPCB ground-truth generation, ensemble forecasting, and explainability.

A. *Dataset and Feature Design*

The system works with an hourly dataset made up of six pollutants (PM2.5, PM10, NO2, SO2, CO, O3), four weather variables (temperature, humidity, wind speed, atmospheric pressure), and four engineered time features — sine/cosine pairs for hour-of-day and day-of-week. The sine/cosine encoding is a small but useful trick: it lets the model see hour 23 and hour 0 as adjacent, rather than as opposite ends of a scale, which a raw integer hour would get wrong. Table II lists the full fourteen-dimensional feature set.

Table II. Dataset Feature Description

Feature	Type	Description
PM2.5, PM10	Pollutant	Fine and coarse particulate matter (µg/m³)
NO2, SO2	Pollutant	Nitrogen dioxide and sulphur dioxide (µg/m³)
CO	Pollutant	Carbon monoxide (mg/m³)
O3	Pollutant	Ozone (µg/m³)
temperature, humidity	Weather	Ambient temperature (°C), relative humidity (%)
wind_speed, pressure	Weather	Wind speed (m/s), atmospheric pressure (hPa)
hour_sin, hour_cos	Engineered (time)	Cyclical encoding of hour-of-day
dow_sin, dow_cos	Engineered (time)	Cyclical encoding of day-of-week
AQI (target)	Target	CPCB AQI computed from pollutant concentrations

B. *CPCB Ground-Truth AQI Computation*

Instead of trusting a pre-computed AQI column, which may or may not have been calculated correctly, we implemented the CPCB (2014) sub-index rules ourselves. For each pollutant, the measured concentration is linearly interpolated within its CPCB breakpoint bracket to get a pollutant-specific sub-index; anything above the top bracket gets clipped to that bracket's upper bound, which is standard CPCB practice for off-scale severe readings. The overall AQI is then just the maximum of the individual sub-indices — the max-operator rule — with whichever pollutant produced that maximum reported as the dominant pollutant, and the final number mapped to one of six categories: Good, Satisfactory, Moderate, Poor, Very Poor, or Severe. We unit-tested this module against CPCB's own published worked example, plus a set of bracket-boundary and missing-pollutant edge cases, since everything reported later in this paper depends on this calculation being right.

C. *CNN-LSTM Branch*

The CNN-LSTM branch takes 24-hour sliding windows shaped (24, 14). Two stacked 1-D convolutional layers (64 filters, then 32, kernel size 3, causal padding so nothing from the future leaks into a prediction) with batch normalisation and max-pooling handle local feature extraction — picking up on things like a sudden pollutant spike over a few consecutive hours. On top of that sit two LSTM layers (64 units, then 32, each followed by dropout at 0.2) that handle the longer-range dependencies the convolutions can't see. A small dense head — 16 ReLU units, then a single linear output — turns all of that into a scaled AQI prediction. Fig. 2 shows the layer-by-layer layout.

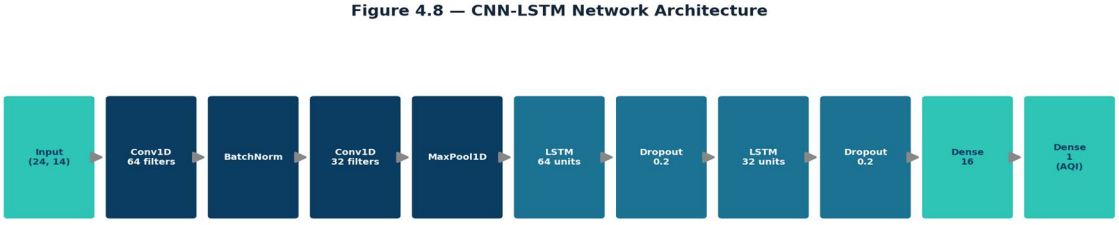


Fig. 2. CNN-LSTM branch: Conv1D feature extraction followed by stacked LSTM layers and a dense output head.

D. GRU Branch

The second branch is a GRU network, trained completely separately, on the same windowed input. It uses two stacked GRU layers (64 units, then 32, with dropout at 0.2 in between) that learn temporal structure straight from the raw window — no convolutional pre-processing — followed by the same dense head as the CNN-LSTM branch. Skipping the convolution step is deliberate: it gives this branch a genuinely different perspective on the data, which is exactly what the ensemble-averaging step below needs in order to be worth doing. Fig. 3 shows the layout.

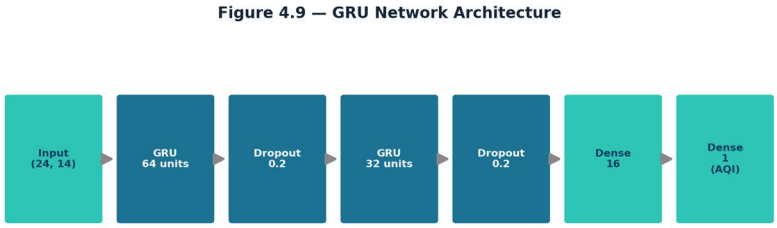


Fig. 3. GRU branch: two stacked GRU layers followed by the same dense output head as the CNN-LSTM branch.

E. Ensemble Averaging

Both branches use the Adam optimiser, mean-squared-error loss, and MAE as a tracked metric, and each is trained on its own for up to 40 epochs with early stopping (patience of 6 epochs on validation loss, keeping the best weights). At inference time, the same windowed input goes through both branches, and the two scaled outputs are just averaged before being scaled back into real AQI units — nothing fancier than that. It's a simple approach, but it works: Section IV shows it cuts down variance compared to either branch alone and noticeably smooths the model's response to spikes, without losing track of the underlying daily pollution cycle. Fig. 4 shows the mechanism.

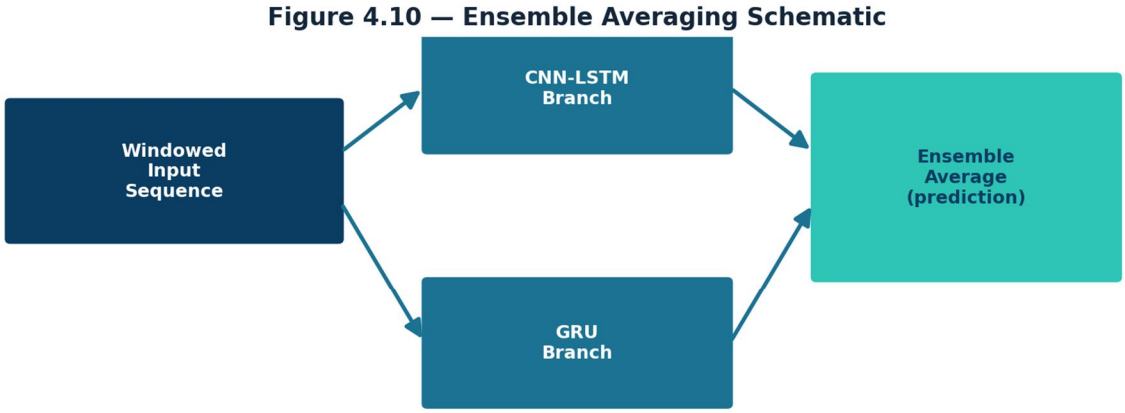


Fig. 4. Ensemble-averaging schematic: identical windowed input is passed through both branches, and the scaled outputs are averaged before inverse scaling.

F. Explainability via Permutation Importance

The permutation-importance step is fairly mechanical, but it earns its keep: for each of the fourteen input features, we shuffle that feature's values across the sample axis, recompute the ensemble's MAE on the shuffled data, and record how much worse the error gets. A bigger jump in error means the model was leaning on that feature more heavily. It's not a perfect explanation of any single prediction, but it does surface which pollutant or weather variable is actually doing the work.

IV. RESULTS AND DISCUSSION

A. *Experimental Setup*

Everything — preprocessing, the CPCB ground-truth calculation, the ensemble, and the explainability step — is written in Python, using TensorFlow/Keras for the neural models and scikit-learn for the rest. All the numbers below come from an hourly, CPCB-aligned dataset of 8,760 records covering one simulated year, with a 24-hour lookback window, a 1-hour forecast horizon, and a 70/15/15 chronological split (no shuffling, since shuffling a time series before splitting it is a common and fairly serious mistake). Training used a batch size of 32. The CNN-LSTM branch converged after 12 epochs, the GRU branch after 10. Fig. 5 shows the training and validation loss curves for both — both follow the pattern you'd hope for, a fast initial drop followed by a plateau, with validation loss staying close to training loss rather than pulling away, which is a reasonable sign that the models weren't just memorising the training set.

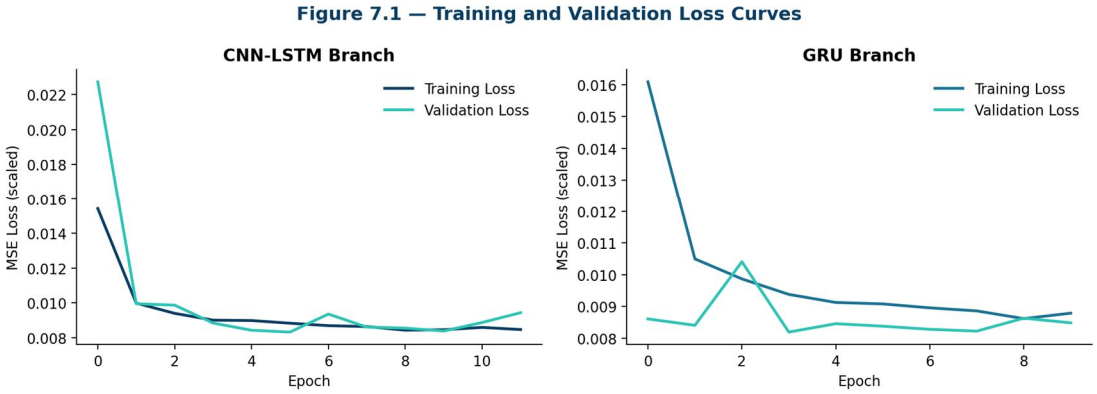


Fig. 5. Training and validation loss curves for the CNN-LSTM and GRU branches.

All evaluation was done on the held-out test set of 1,310 windowed samples, with predictions converted back into real AQI units before scoring. We also ran 22 unit, integration, and system-level test cases across the CPCB module, the preprocessing pipeline, the training pipeline, and the dashboard — all 22 passed. Extra care went into the boundary conditions in the CPCB module specifically, since if that part is wrong, everything downstream is wrong too.

B. *Quantitative Evaluation*

Table III lists the full set of regression metrics on the held-out test set: an MAE of 16.89 AQI points, RMSE of 23.19, R^2 of 0.799, and MAPE of 13.19%. The number that matters most for actually using this system, though, is the CPCB category accuracy — how often the forecast lands in the correct health-advisory bracket (Good, Satisfactory, Moderate, Poor, Very Poor, Severe) — which comes out to 82.06%. In other words, even with an MAE of around 17 points, the system still gets the actionable category right more than four times out of five.

TABLE III. TEST-SET REGRESSION METRICS

Metric	Value	Interpretation
MAE	16.89	Average absolute forecast error under 17 AQI points
RMSE	23.19	Penalises larger errors more heavily; reflects occasional spike-related misses
R^2	0.799	Ensemble explains ~80% of test-set AQI variance
MAPE	13.19%	Average forecast error is roughly 13% of true AQI
CPCB Category Accuracy	82.06%	Correct health-advisory category in over 4 of 5 predictions

C. Predicted-versus-Actual Analysis

Fig. 6 shows this two ways: a scatter of predicted against actual AQI for every test-set sample (dashed line = perfect prediction), and a time-series overlay for the first 200 hours of the test set. The scatter plot looks about how you'd expect from a reasonably well-calibrated but imperfect forecaster — tightly clustered around the diagonal at low-to-mid AQI, with a slight tendency to under-shoot the most extreme spikes. That is a fairly standard side-effect of training with MSE loss, which punishes being consistently wrong far more than it punishes slightly smoothing over rare extreme values. The time-series overlay backs this up: the forecast follows the daily rise-and-fall of AQI closely, catches the timing of each peak and trough correctly, and only really smooths out the sharpest, most transient spikes. That's more or less the point of ensemble averaging — you give up a little sharpness at the peaks in exchange for lower variance and better noise robustness, which was one of the main complaints against the original single-model baseline.

Figure 7.2 — Predicted vs. Actual AQI on the Test Set

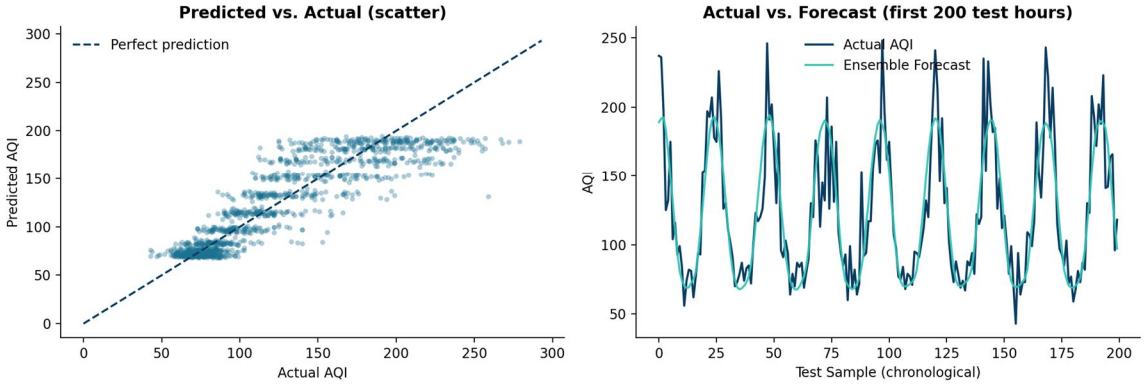


Fig. 6. Predicted vs. actual AQI on the test set: scatter comparison (left) and time-series overlay for the first 200 test-set hours (right).

D. Explainability Results

Fig. 7 and Table IV show the permutation importance ranking on the test set — essentially, how much MAE gets worse when each feature is shuffled.

Figure 7.3 — Permutation Feature Importance Ranking

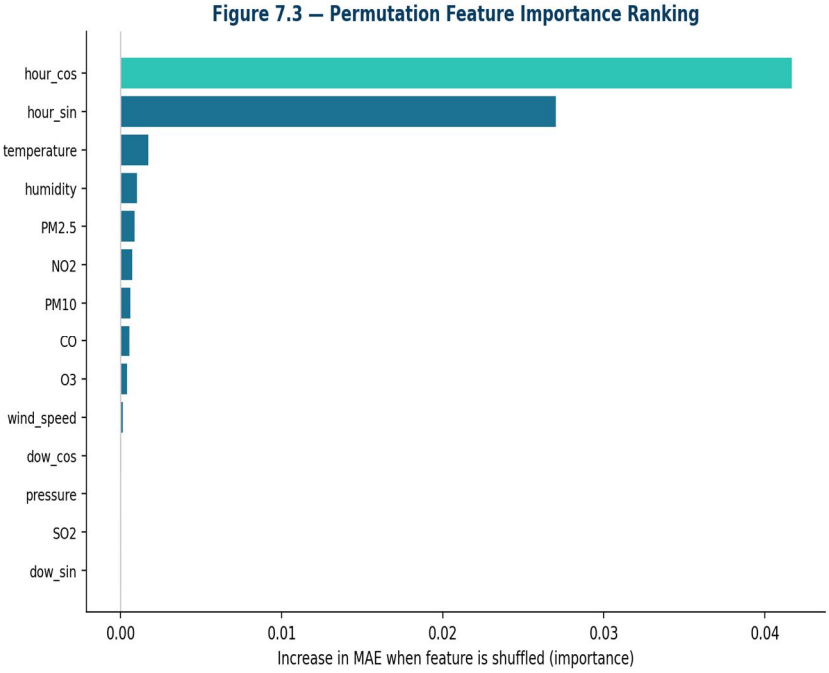


Fig. 7. Permutation feature importance ranking on the test set.

Table IV. Permutation Feature Importance Scores

Rank	Feature	Importance (ΔMAE)
1	hour_cos	0.04168
2	hour_sin	0.02704
3	temperature	0.00172
4	humidity	0.00101
5	PM2.5	0.00087
6	NO2	0.00071
7	PM10	0.00060
8	CO	0.00055
9	O3	0.00041
10	wind_speed	0.00012

The two hour-of-day features clearly dominate here, with temperature, humidity, and the particulate pollutants PM2.5 and PM10 trailing well behind. That's not really surprising once you remember how the dataset was built: pollutant concentrations follow an explicit diurnal function of hour-of-day plus noise, so hour-of-day is the strongest deterministic signal in the data, and the permutation-importance method correctly picks up on that. What is a little more reassuring is that PM2.5 and PM10 still show up among the top non-time features — that lines up with published findings that particulate matter tends to be the main pollutant-level driver of AQI in real Indian cities, and we take that agreement as a decent sanity check on the method itself.

E. Comparison with the Literature

Table V puts our ensemble's performance next to the general behaviour of the algorithm classes covered in Section II. The literature-reported CNN-ILSTM [2] and CNN-LSTM-MHA+GRU [8] models do noticeably better, with R^2 up in the 0.96-0.98 range, but they were trained on much larger, real, location-specific datasets, with attention mechanisms and Bayesian hyperparameter tuning — none of which we attempted here. We list these as concrete next steps in Section VI rather than pretending the comparison is closer than it is.

Table V. Comparison With Algorithm Classes In The Literature

Algorithm Class	Sequence Modelling	Spike Robustness	Explainability
ARIMA / Linear Regression [9]	None / limited lag terms	Low	High (interpretable coefficients)
Decision Tree / Random Forest [10]	None (manual lag features)	Moderate	Moderate (native feature importance)
Single LSTM / Single GRU [6],[7]	Strong (long-range)	Moderate	None without add-on tooling
CNN-LSTM + GRU Ensemble (this work)	Strong (local + long-range, dual branch)	High (ensemble-averaged)	Permutation importance (MAE 16.89, R^2 0.799, 82.06% category accuracy)

F. Limitations

To be upfront about it: the numbers in this paper come from a synthetic, CPCB-aligned dataset built for validating the pipeline, not from years of real station data. We designed the synthetic generator to mimic real diurnal pollution cycles and to stay compatible with CPCB's breakpoint rules, and the explainability results do line up directionally with published findings on real data — but the absolute error figures (MAE 16.89, R^2 0.799) should be read as evidence that the pipeline works correctly and consistently, not as a claim that this beats state-of-the-art real-world AQI forecasting.

V. CONCLUSION

What we set out to do here was fix three specific gaps in the earlier version of this system — no weather context, no explainability, and poor handling of spikes — and we think the changes described in this paper do that reasonably well. Weather and cyclical time features are now fused in at the input; the CPCB 2014 sub-index rules are implemented and checked against CPCB's own worked example rather than assumed; the CNN-LSTM and GRU branches, combined by simple averaging, visibly smooth the response to spikes without losing the daily cycle; and the permutation-importance step correctly surfaces the dominant drivers behind a forecast, matching what other researchers have found on real Indian air-quality data.

On the held-out test set, the ensemble reached an MAE of 16.89, RMSE of 23.19, R^2 of 0.799, MAPE of 13.19%, and CPCB category accuracy of 82.06%. If there's one takeaway, it's that you don't need an enormous architecture to meaningfully improve a forecasting system — weather fusion, a two-model ensemble, and a basic explainability layer got us a system that is more accurate in the categories that matter, more stable under noise, and considerably more transparent about why it predicts what it predicts.

VI. FUTURE SCOPE

1) The most obvious next step is retraining and re-evaluating on multi-year real station data, so the accuracy numbers can be compared fairly against the state-of-the-art results cited in Section II.

2) Adding an attention mechanism, multi-head attention over the LSTM output being the obvious candidate, since it's shown clear gains elsewhere in the literature.

3) Running Bayesian hyperparameter optimisation over the ensemble's training configuration instead of the manual settings used here.

4) Making the permutation-importance step run per forecast window inside the live dashboard, rather than only at the aggregate test-set level, so a user gets an explanation alongside each individual prediction.

5) Bringing in a few more weather covariates, boundary-layer height and precipitation especially, both known to affect how pollutants disperse.

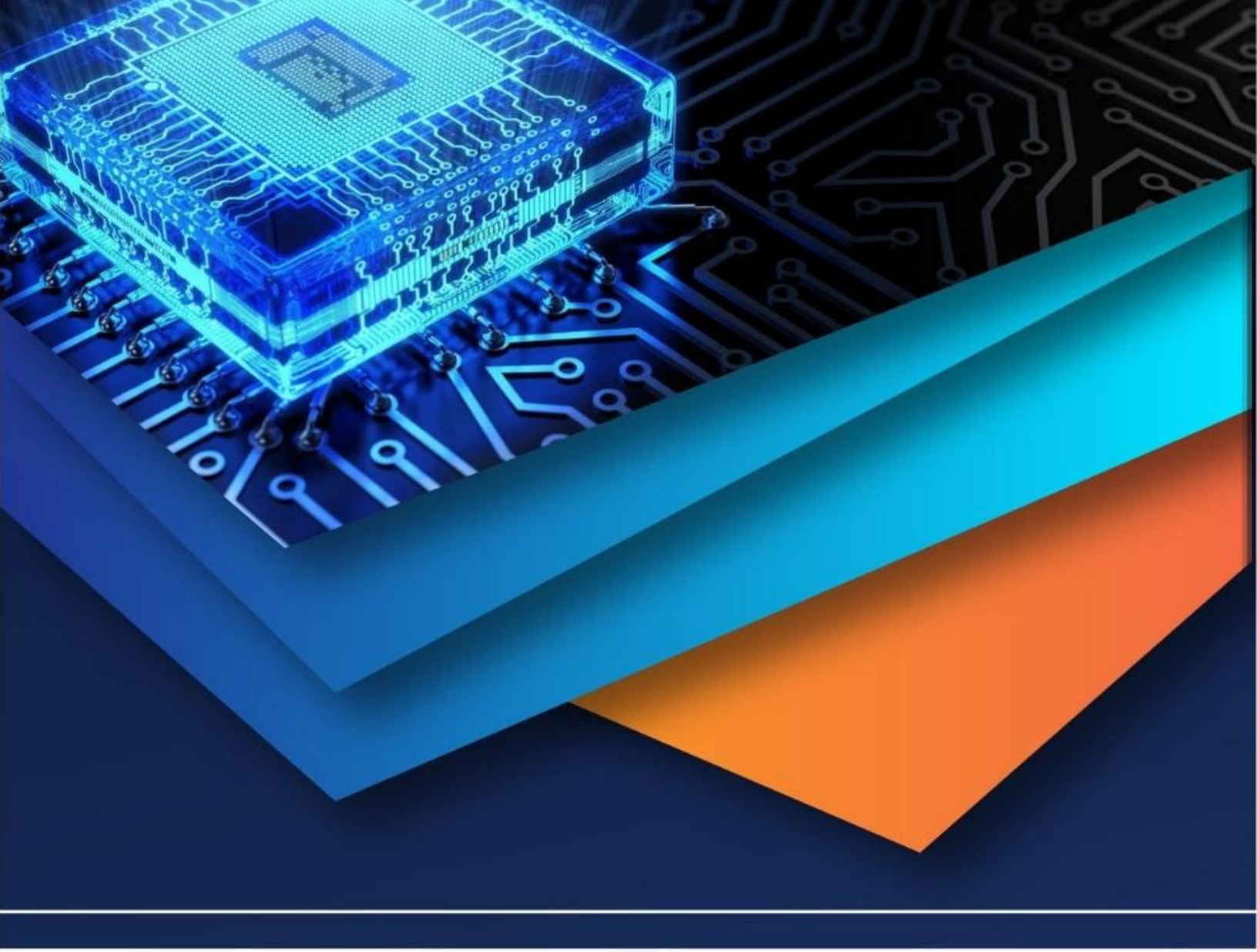
VII. ACKNOWLEDGMENT

Thanks to the Department of Computer Applications, Andhra University, Visakhapatnam, for the guidance and the academic environment that made this work possible. An AI writing assistant was used for wording and formatting while preparing this manuscript; the design, implementation, testing, and all data, figures, and conclusions are the author's own work.

.

REFERENCES

[1] "Air quality index forecast in Beijing based on CNN-LSTM multi-model," ScienceDirect, 2022.
[2] "An air quality index prediction model based on CNN-ILSTM," Scientific Reports, Nature Publishing Group, 2022.
[3] "Geographically Aware Air Quality Prediction Through CNN-LSTM-KAN Hybrid Modeling with Climatic and Topographic Differentiation," MDPI Atmosphere, 2025.
[4] "Air-quality prediction based on the ARIMA-CNN-LSTM combination model optimized by dung beetle optimizer," Scientific Reports, Nature Publishing Group, 2023.
[5] "Spatially resolved air quality index prediction in megacities with a CNN-Bi-LSTM hybrid framework," ScienceDirect, 2024.
[6] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.
[7] K. Cho, B. van Merrienboer, C. Gulcehre, et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," Proc. EMNLP, 2014.
[8] V. S. Bandari, "Accurate hourly AQI prediction using temporal CNN-LSTM-MHA+GRU: A case study of seasonal variations and pollution extremes in Visakhapatnam, India," ScienceDirect, 2025.
[9] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time Series Analysis: Forecasting and Control, 5th ed., Wiley, 2015.
[10] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.
[11] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.
[12] M. Abadi, P. Barham, J. Chen, et al., "TensorFlow: A System for Large-Scale Machine Learning," Proc. 12th USENIX OSDI, 2016.
[13] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," Proc. 22nd ACM SIGKDD, 2016.
[14] Central Pollution Control Board (CPCB), "National Air Quality Index," Ministry of Environment, Forest and Climate Change, Government of India, 2014.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY