Predicting Air Quality in Californian Counties

Stanford CS229 Project

Abraham Yeung
Department of Computer Science
Stanford University
ayeung16@stanford.edu

Yashas Mattur
Department of Computer Science
Stanford University
ykmattur@stanford.edu

1 Introduction

Air quality forecasting is critical for public health, urban planning, and environmental policy, particularly in wildfire-prone California. Accurate county-level AQI predictions are essential to guarantee timely alerts for communities. Our model processes a 10-year time series of daily AQI values for 52 California counties, where each sample includes a 30-day AQI sequence and the mean latitude/longitude of monitoring stations. We construct a graph capturing spatial relationships between counties, which a GNN processes to generate embeddings that incorporate both local and neighboring influences. An LSTM models temporal dependencies in AQI trends, while various hybrid-model combinations of GNN, CNN, and LSTM feed into a fully connected network to predict the next day's AQI, leveraging both spatial and temporal patterns to generate an optimal AQI-prediction model.

2 Related Works

Air quality forecasting has been studied extensively, with approaches ranging from traditional statistical models to deep learning techniques. Early methods such as autoregressive integrated moving average (ARIMA) and support vector regression (SVR) provided interpretable results but struggled to capture complex spatial and temporal dependencies (1; 2). In contrast, deep learning models like long short-term memory (LSTM) networks and convolutional neural networks (CNNs) have demonstrated superior accuracy by effectively modeling temporal sequences and spatial relationships (3). More recently, graph neural networks (GNNs) have emerged as a state-of-the-art approach by incorporating spatial correlations between monitoring stations (4; 5).

Hybrid methods integrating deep learning with domain-specific knowledge further enhance predictive performance. For instance, variational mode decomposition combined with a hybrid CNN-LSTM model has been used to improve time-series forecasting (3), while physics-informed GNNs have leveraged real-time sensor data with physical constraints for better generalization (6). Expandable graph attention networks have also been used to dynamically adjust graph structures to adapt to changing pollution patterns (7).

While present LSTM- and CNN/GNN-based AQI-forecasting models have gained prevalence for their high accuracy, research has yet to be done on combining these models with an attention-head-based model, utilizing data from surrounding regions in forecasting AQI predictions. Our work builds upon this basis in an effort to increase accuracy in the geographic AQI prediction abilities of our model.

3 Dataset and Features

Our dataset is comprised of daily air quality measurements collected from multiple counties. We obtained the data from the U.S. Environmental Protection Agency's (EPA) Outdoor Air Quality API. We collated together 10 datasets from 2015-2024. The raw dataset contains over 545,000 records in 52 Californian counties, each corresponding to measurements from various monitoring stations. For our analysis, we aggregated these measurements to produce a single Daily AQI value per county, and then applied the Interquartile Range (IQR) method to remove extreme outliers.

The time-series data is discretized into days. Our training/validation/test split is 80:10:10. Thus, we had roughly 400,000 training samples, 50,000 validation samples and 50,000 test samples. We mainly use the AQI, county, and date features but we also explored the PM 2.5 feature as well.

For pre-processing, we converted date strings into datetime objects and sort the data chronologically and removed outliers. During our preliminary experiments, we also used Fourier transforms to capture seasonal periodicities in AQI data. We also computed a mean latitude and longitude for each county and normalized these to zero mean and unit variance to construct a k-nearest neighbor graph, which our Graph Neural Network (GNN) used to generate county embeddings. These embeddings capture the spatial relationships between counties and are integrated into our hybrid model.

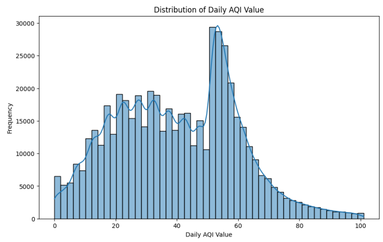

Figure 1: AQI Dataset Distribution

Finally, we noted that, even after removing outliers, our data distribution followed a somewhat right-skewed distribution, as demonstrated in Figure 1.

4 Methods

4.1 Baseline and Classical ML Methods

For a baseline, we use a persistent model where the prediction is $\hat{y}_{t+1} = y_t + b_c + b_g$, where y_t is the last observed AQI value, b_c is a county-specific bias learned via an embedding, and b_g is a global bias. For classical ML methods, we employ gradient boosting models such as XGBoost and CatBoost as well as ensemble methods like Random Forests. XGBoost and CatBoost are implementations of gradient boosting trees that builds an ensemble of decision trees in a sequential manner. Random Forests is an ensemble learning method that constructs a large number of decision trees independently using bootstrapped subsets of the training data and a random subset of features at each split.

4.2 Long Short-Term Memory Network

Our key deep learning algorithm is the LSTM (Long Short-Term Memory) Network. They are a special kind of recurrent neural network (RNN) designed to capture long-range dependencies in sequential data while mitigating the vanishing gradient problem in time series data. Using a series of gates, it can selectively keep and remove information to keep the most important features. All of the LSTM models we trained also had a county-embedding feature.

4.3 Our Novel Models

We try **two novel GNN-LSTM models**. Both use a GNN (Graph Neural Network) processes the mean longitude and latitude data and k-nearest neighbors graph we produced to produce an embedding for each county $z_c \in \mathbb{R}^d$. Our first model simply uses this GNN embedding on top of LSTM. For our second model, we include a **county-neighbor attention** mechanism. For each county c with embedding z_c, we determine its neighbors $N(c)$ from the k-NN graph. An attention mechanism computes weights for neighboring embeddings $\{z_n : n \in N(c)\}$ and with the aggregated neighbor embedding:

$$a_n = \frac{\exp(z_c^\top z_n)}{\sum_{j \in N(c)} \exp(z_c^\top z_j)}, \quad \bar{z}_c = \sum_{n \in N(c)} a_n z_n.$$

We then concatenate the county's own embedding with the aggregated neighbor embedding and project it:

$$c_{\text{proj}} = \text{ReLU}(W_{\text{proj}}[z_c \oplus \bar{z}_c] + b_{\text{proj}}).$$

Finally, the LSTM output h_t and c_{proj} are concatenated and the final AQI forecast is given by:

$$\hat{y}_{t+1} = W_{\text{fc}}[h_t \oplus c_{\text{proj}}] + b_{\text{fc}}.$$

We also train a novel **CNN-LSTM with hidden state attention model** that uses attention to focus on important time steps. Let $X \in \mathbb{R}^{B \times L \times 1}$ denote the input AQI time series, where B is the batch size and $L = 30$ is the sequence length. The first stage applies a 1D convolution to extract local features:

$$F = \text{ReLU}\big(\text{BatchNorm}(\text{Conv1d}(X))\big) \quad \in \mathbb{R}^{B \times L \times C},$$

with C being the number of CNN channels.

The CNN features are passed to an LSTM that processes the temporal dynamics. The LSTM produces hidden states $\{h_1, h_2, \ldots, h_L\}$, with $h_t \in \mathbb{R}^H$. To focus on the most relevant time steps, we compute attention weights and form a context vector:

$$\alpha_t = \frac{\exp\left(w_a^\top h_t\right)}{\sum_{s=1}^{L} \exp\left(w_a^\top h_s\right)}, \quad c = \sum_{t=1}^{L} \alpha_t h_t.$$

In parallel, each county is associated with a learned embedding $e_c \in \mathbb{R}^E$ (with $E = 16$). We concatenate the LSTM context c with the county embedding and project:

$$z_{\text{proj}} = \text{ReLU}\left(W_{\text{proj}}[c \oplus e_c] + b_{\text{proj}}\right),$$

and further concatenated with the final LSTM hidden state h_L to yield the final feature vector, with the forecast being:

$$z_{\text{final}} = h_L \oplus z_{\text{proj}}, \quad \hat{y}_{t+1} = W_{\text{fc}}\, z_{\text{final}} + b_{\text{fc}}.$$

5 Experiments

5.1 Primary Metrics

The primary evaluation metrics for our models are Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). These are defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \quad MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

RMSE penalizes larger errors more heavily than MAE, making it sensitive to outliers, while MAE provides a direct measure of average prediction error.

5.2 Hyperparameter Optimization

We employed a Genetic Algorithm to optimize hyperparameters for our LSTM and GNN, training for ten generations, considering 15 candidates per generation. We used discrete parameters with 20% mutation rate. For each, we varied the number of layers and dropout rate. We also varied the hidden layer dimension and graph embedding dimension for LSTM and GNN respectively. We used a 3-fold cross-validation to assess each model. The best hyperparameters found were LSTM Layers: 2, GNN Layers: 2, Dropout Rate: 0.5, LSTM Hidden Dimension: 100, GNN Embedding Dimension: 16.

5.3 Overfitting Mitigation and Generalization

To mitigate overfitting, we tried three key techniques on our LSTM model:

(a) Weight Decay and Dropout We tried preventing overfitting with weight decay and dropout in one model. Our weight decay regularized the model by adding an L2 penalty of 10^{-5} to the loss: $\mathcal{L}_{\text{total}} = \mathcal{L} + 10^{-5}\|w\|^2$. This discouraged overly large weights. For dropout, we randomly disabled 50% of neurons during training, forcing redundancy in learned representations.

(b) Early Stopping and EMA Smoothing We also tried early stopping, which halted training when validation loss stagnates or increases for more than 5 epochs in a row, preventing the model from fitting noise. We also added EMA (Exponential Moving Average) smoothing, which stabilizes training by averaging model parameters over time: $\bar{\theta}_t = 0.99 \cdot \theta_t + (1 - 0.99)\bar{\theta}_{t-1}$. This improves generalization by reducing the impact of noisy updates.

(c) Ensemble Learning In our ensemble model, we trained 3 separate county-conditioned LSTM models and took the mean of the predictions to reduce variance and improve robustness. This helped mitigate individual model errors.

We observed that these strategies had improvements on RMSE, with weight decay + dropout (RMSE = 14.14) and ensemble performing (RMSE = 14.11) better than benchmark county-conditional LSTM (RMSE = 14.16). Collectively, they enhanced our model stability and generalization, improving regularization and predictive performance.

5.4 Prediction Range Constraint

As was discussed earlier with Figure 1, our dataset is significantly right-skewed; as such, our models consistently predict AQI values within 5–80, despite the dataset containing the full range of non-outlier values from 0 to 101 (an example of which is illustrated in Figure 2). This suggests a bias toward the most common AQI range, limiting generalization to extreme pollution events. One likely cause is **loss function bias**—RMSE and MAE prioritize minimizing errors where data is most concentrated, reducing sensitivity to rare outliers. **Data imbalance** further exacerbates this, as severe pollution events are underrepresented, making them harder to learn. Additionally, **regularization effects**, such as dropout and weight decay, may suppress extreme predictions by discouraging large parameter updates.

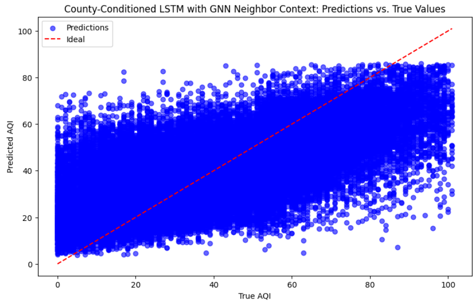

Figure 2: An Example of Prediction Range Constraints with our LSTM-GNN-Attention Model

5.5 Evaluation and Discussion

Figure 3 summarizes the performance of our models. Not included in the graph is our baseline persistent model performed much worse (RMSE = 17.77) and all ML methods outperformed this. The LSTM-based models outperformed tree-based methods, with the best results from the **County-Conditioned LSTM with CNN Embeddings and Hidden State Attention** and the **County-Conditioned LSTM with GNN Embeddings and County-Neighbor Attention**, achieving RMSEs of 14.08 and 14.10, respectively. This improvement can be attributed to:

- **Sequential Modeling:** LSTMs are well-suited for time-series data as they maintain long-term dependencies through memory cells and gating mechanisms, mitigating issues like vanishing gradients. This allows the model to learn seasonal AQI trends and capture persistent pollution patterns that tree-based models struggle with.
- **CNN and GNN Embeddings:** CNN embeddings capture temporal patterns by applying localized filters, identifying key pollution patterns. In contrast, GNN embeddings use graph-based message passing to model spatial relationships between counties, allowing the model to incorporate pollution influences from both neighboring and distant but correlated regions. Both captured different features that yielded improvements.
- **Novel County-Neighbor Attention:** The attention mechanism dynamically weighs contributions from neighboring counties, allowing the model to focus on the most relevant spatial dependencies. This is particularly important as we see marginal improvements when only using the GNN embeddings without attention. The model with attention captured the spatial dependency much better. This is particularly useful for AQI prediction, where pollution levels in a county can be influenced by distant but highly correlated areas too. Attention enhanced interpretability by highlighting influential regions in air quality forecasting.

Each of these components contributes uniquely to model performance, but their combined effect yielded great improvement. The integration of sequential modeling, spatial embeddings, and attention mechanisms allows the model to capture complex temporal/regional dependencies, leading to a more comprehensive AQI prediction system.

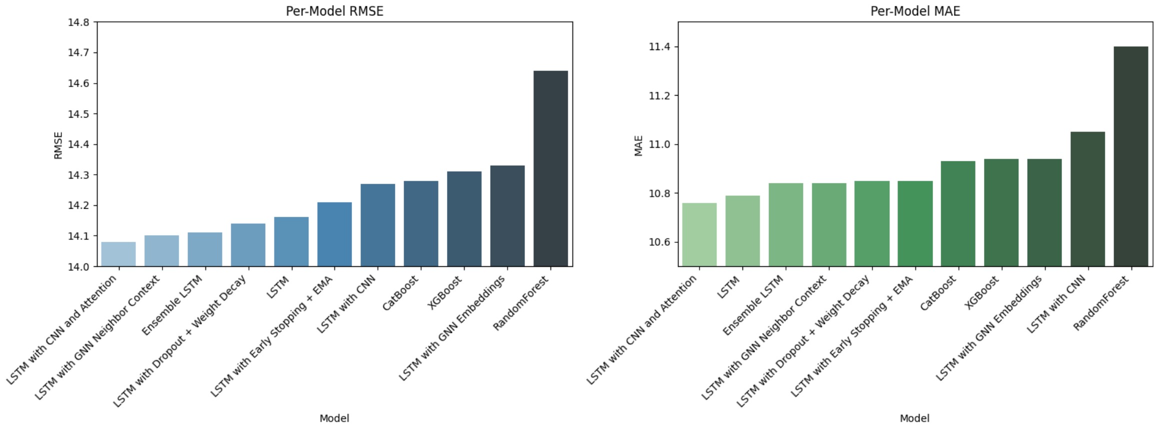

Figure 3: Model Performances

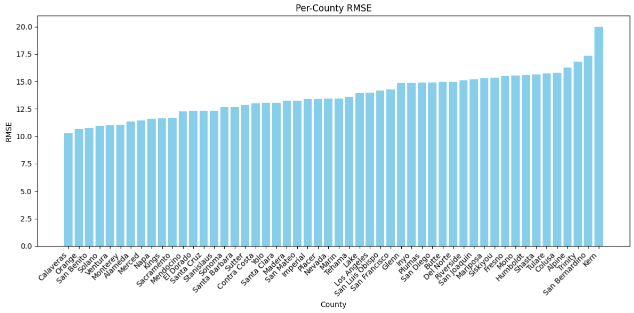

Figure 4: Models' RMSE Across All California Counties

5.6 Further Geographical Constraints

While our models performed well overall, their effectiveness varies across counties, as shown by the RMSE differences in Figure 4. Counties with stable AQI trends align well with our models, whereas regions like Los Angeles exhibit greater volatility, likely due to factors such as urban sprawl and localized pollution. These influences, not fully captured in our dataset, suggest the need for additional urban-specific features, such as traffic density, to improve predictions in metropolitan areas.

6 Conclusion and Future Steps

We explored various ML and deep learning approaches for AQI prediction in California. Hybrid models like GNN-LSTM with Geographical Attention and CNN-LSTM with Temporal Attention outperformed traditional baselines by capturing spatial and temporal dependencies. Deep learning models, particularly those integrating attention and graph-based representations, excelled in structured environmental data, while classical models like XGBoost struggled with complex dependencies.

Future steps we may take for our project include incorporating meteorological and satellite data, self-supervised learning (e.g., masked autoencoders), and real-time sensor readings. Transformer-based architectures and real-time forecasting with uncertainty quantification could further enhance predictions. Also, training-time weighting or data oversampling for underrepresented high-AQI days could combat our prediction range constraint. We were also unable to run Genetic Algorithm hyperparameter-tuning for a greater, more significant number of generations due to Cloud Compute Credit limits, but better hyperparameters could yield better results. Nevertheless, our findings highlight deep learning's potential in AQI forecasting and set the stage for future advancements.

7 Contributions

Abraham Yeung: Worked on persistent baseline, XGBoost and CatBoost. Also worked on preventing overfitting in models with weight decay, dropout, smoothing, early stopping and ensemble methods. Also, researched GNNs and trained k-nearest neighbors graph and came up with county-neighbor attention.

Yashas Mattur: Pre-processed dataset and conducted Exploratory Data Analysis. Worked on the Random Forest baseline model; also worked on researching CNNs in related works, and also implementing CNN models as part of baseline models, as embeddings in the LSTM models, and with Hidden-State Attention. Also worked on Genetic Algorithm to optimize hyperparameters for the model.

References

[1] W. Leong, R. Kelani, and Z. Ahmad, "Prediction of air pollution index (api) using support vector machine (svm)," *Journal of Environmental Chemical Engineering*, 2019.

[2] W. Gao, T. Xiao, L. Zou, H. Li, and S. Gu, "Analysis and prediction of atmospheric environmental quality based on the autoregressive integrated moving average model (arima model) in hunan province, china," *Sustainability*, 2024.

[3] G. Drewil and R. Al-Bahadili, "Air pollution prediction using lstm deep learning and metaheuristics algorithms," *Measurement: Sensors*, vol. 24, 2022.

[4] L. Chen, J. Xu, B. Wu, Y. Qian, Z. Du, Y. Li, and Y. Zhang, "Group-aware graph neural network for nationwide city air quality forecasting," 2021.

[5] D. Iskandaryan, J. Ramos, and S. Oliver, "Graph neural network for air quality prediction: A case study in madrid," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[6] D. Lorenzo, V. Champaney, C. Ghnatios, and E. Cueto, "Physics-informed and graph neural networks for enhanced inverse analysis," *Engineering Computations*, 2024.

[7] J. Zuo, W. Li, M. Baldo, and H. Hacid, "Opportunistic air quality monitoring and forecasting with expandable graph neural networks," *arXiv*, 2023.