Al Akhawayn University in Ifrane

School of Science and Engineering

Indoor Environmental Quality Forecasting in a Smart Aquaculture Facility

Project Group:

Oussama Oukkassou
Mohammed Amine Rourrou
Kaoutar Elasri

December 2, 2025

Contents

1 Project Objectives and Hypotheses

1.1 Motivation and Objectives

Indoor Environmental Quality (IEQ) is critical for both fish welfare and worker comfort in closed aquaculture facilities. Poor IEQ can cause stress, increase disease risk and reduce productivity. However, IEQ is often monitored in a purely reactive way: operators respond only after conditions have already degraded.

The main objective of this project is therefore to design and evaluate a data-driven system that can:

- forecast IEQ variables **one hour in advance**,
- use these forecasts to support proactive ventilation and control,
- and work under real operating conditions in a smart aquaculture facility in Morocco.

The system predicts several IEQ variables simultaneously: temperature, relative humidity, CO_2, VOCs, $PM_{2.5}$, PM_{10}, and a synthetic IEQ score.

A second objective is to compare three deep-learning architectures under identical conditions:

- a Long Short–Term Memory network (LSTM),
- a Temporal Convolutional Network (TCN),
- a Transformer-based model.

All models receive the same six-hour input window and predict the same one-hour horizon. The comparison focuses on accuracy, robustness and ease of deployment.

1.2 Type of Machine Learning Task

This work is a multivariate time-series forecasting and regression problem.

Regression. All targets are continuous variables (CO_2 in ppm, IEQ score, etc.). The models predict numerical values, so the task is multivariate regression.

Multivariate time series. At each time step the system observes several environmental variables. The models must learn both:

- temporal dependencies across time,
- and cross-variable interactions (e.g. temperature–humidity coupling).

Decision support. The system is not meant to run the facility autonomously. Instead, forecasts are presented in a dashboard and used by operators as decision-support signals (e.g. when to ventilate or raise an alarm).

1.3 Research Hypotheses

The following hypotheses guide the study:

H1: Deep learning models (LSTM, TCN, Transformer) will outperform classical machine-learning baselines (e.g. Random Forests) for IEQ forecasting.

H2: The Transformer model will achieve the highest overall accuracy thanks to its self-attention mechanism and long-range dependency modelling.

H3: The TCN will outperform the LSTM on short-term IEQ fluctuations due to its dilated causal convolutions.

H4: Feature engineering (time features, lag features, scaling) will reduce forecasting error by at least 10 % compared to raw sensor values alone.

H5: Multivariate models that use all IEQ variables jointly will outperform univariate models trained separately for each variable.

H6: A six-hour input window (72 time steps) will perform better than shorter windows, because it captures typical ventilation cycles and part of the diurnal pattern.

2 Dataset

2.1 Source and Context

The dataset comes from a smart indoor aquaculture facility in Morocco. A network of IoT environmental sensors installed in the production building continuously recorded indoor air conditions relevant to fish welfare and worker comfort.

The dataset is private but closely resembles the public AQUAIR dataset, which logs six IEQ variables (temperature, relative humidity, CO_2, VOC, $PM_{2.5}$, PM_{10}) inside a trout hatchery in Amghass, Azrou, Morocco.[6] Like AQUAIR, the dataset used here has a sampling interval of five minutes and more than 23 000 records.

2.2 Recorded Variables

Table 1 lists the main IEQ variables.

Table 1: IEQ variables recorded in the aquaculture facility.

Variable	Unit	Role
Air temperature	°C	feature and target
Relative humidity	%	feature and target
CO_2 concentration	ppm	feature and target
VOC level	ppb	feature and target
$PM_{2.5}$	$\mu g\, m^{-3}$	feature and target
PM_{10}	$\mu g\, m^{-3}$	feature and target
IEQ score	dimensionless	main target index

Additional engineered features derived from the timestamp include:

- hour of day,
- day of week,
- weekday/weekend flag,
- and, in some experiments, short lag values and rolling statistics.

2.3 Size and Sampling

Basic dataset characteristics are summarised in Table 2.

Table 2: Global dataset characteristics.

Property	Value
Sampling interval	5 minutes
Total time steps	≈23 000
Original IEQ variables	7
Engineered time features	2–3
Input window length	72 time steps (6 hours)
Forecast horizon	12 time steps (1 hour)

2.4 Preprocessing Pipeline

Preprocessing steps are:

- **Missing values:** detect via `isna()`, fill short gaps with forward/backward fill or interpolation, and drop long corrupted segments if necessary.
- **Timestamp handling:** convert to `datetime`, use as index, and sort chronologically.
- **Scaling:** apply Min–Max scaling to all continuous features; scalers are saved for later inference.
- **Feature engineering:** add time-of-day features plus optional lagged values and rolling statistics.
- **Sliding-window transformation:** convert the continuous series into overlapping windows: each sample uses the last 72 time steps as input and the next 12 as target.

If the input has d features and the target has m dimensions, then the final tensors have shapes:

$$X \in R^{N \times 72 \times d}, \qquad y \in R^{N \times 12 \times m}.$$

2.5 Ethical and Operational Aspects

Privacy. The facility is anonymised and no human-identifiable information is stored.

Sensor reliability. Low-cost VOC and PM sensors may drift or spike. Preprocessing reduces noise while keeping genuine peaks, which are often biologically meaningful (e.g. feeding).

Responsible deployment. Forecasts are advisory and are reviewed by aquaculture specialists. Blind, fully automatic control based only on model output is avoided.

2.6 Train/Validation/Test Split

The time series is split chronologically:

- 70 % earliest data: training set,
- next 15 %: validation set,
- last 15 %: test set.

Table 3 summarises the split (sample counts are approximate).

Table 3: Chronological data split (approximate counts).

Subset	Fraction	Approx. time steps	Purpose
Train	70 %	∼16 000	model learning
Validation	15 %	∼3 500	hyperparameter tuning
Test	15 %	∼3 500	final unbiased evaluation

Chronological splitting avoids "seeing the future" during training and mimics real deployment.

3 Literature Review

3.1 Indoor Environmental and Air-Quality Forecasting

Deep-learning models are now widely used for indoor air-quality forecasting. Zhu et al. developed an IoT-based system for CO_2 monitoring where an LSTM predicts future CO_2 levels and the steady-state concentration, enabling proactive ventilation in buildings.[1] An attention-based LSTM variant has been used to predict short-term indoor CO_2 in a hospital, comparing several LSTM architectures for horizons from 15 to 180 minutes.[2]

Other works explore hybrid CNN–LSTM architectures and AIoT frameworks. A recent study combined CNN, LSTM and evolutionary optimisation to forecast temperature, humidity, CO_2 and $PM_{2.5}$ in buildings with high accuracy.[3] Several authors compared classical models (ARIMA, SVM, BPNN) against LSTM for IEQ prediction in naturally ventilated dwellings and found that LSTM generally provides better accuracy and process memory.[4]

Sabiri et al. proposed a deep-learning framework to optimise IEQ in smart buildings, benchmarking LSTM, GRU and CNN–LSTM to forecast CO_2, temperature and humidity for predictive HVAC control.[5] These studies confirm that sequence models are effective for IEQ forecasting and motivate our own comparison of LSTM, TCN and Transformer architectures.

3.2 Deep Learning for Aquaculture Monitoring

In aquaculture, a growing number of works use deep learning to predict water quality. Haq and Harigovindan applied hybrid CNN–LSTM and CNN–GRU models to forecast key water-quality parameters in ponds, outperforming classical time-series models.[7] A more recent paper proposed attention-driven LSTM and GRU networks for aquaculture water-quality prediction and showed that attention mechanisms can improve accuracy for multi-step forecasts.[8]

Other studies combine IoT sensing and machine learning for intelligent aquaculture. An MDPI study integrated IoT sensors, machine-learning models and quantum optimisation to monitor and predict water quality in aquaculture systems.[9] Another work used IoT-based data and ensemble models for predictive modelling of tilapia pond water quality.[10] A recent review discusses how IoT, AI and related technologies can transform aquaculture operations into more efficient, data-driven systems.[11]

3.3 IEQ Datasets for Aquaculture Facilities

The AQUAIR dataset provides a high-resolution log of indoor environmental quality inside a trout hatchery in Morocco.[6] It records temperature, relative humidity, CO_2, TVOC, $PM_{2.5}$ and PM_{10} every five minutes over several months. The dataset is designed specifically for IEQ forecasting and anomaly detection in aquaculture environments. Our private dataset follows a similar design and temporal resolution.

3.4 Positioning of This Work

From the IEQ literature we adopt:

- the use of IoT sensors and continuous monitoring,
- LSTM and hybrid deep models for forecasting,
- attention to input-window length and prediction horizon.

From aquaculture studies we adopt:

- the focus on decision support for fish health,
- the use of deep sequence models on water or air quality streams,
- IoT-based deployment considerations.

Our contribution is a unified, end-to-end comparison of LSTM, TCN and Transformer models on IEQ forecasting in a working aquaculture facility, using a realistic six-hour input, one-hour forecasting horizon and a robust evaluation pipeline.

4 System Architecture and Methods

4.1 End-to-End Architecture

The system follows these stages:

1. **Data ingestion:** load CSVs exported by the monitoring system.

2. **Preprocessing:** clean, interpolate, scale and engineer features.
3. **Sliding-window dataset:** convert to supervised windows.
4. **Model training:** train LSTM, TCN and Transformer in PyTorch.
5. **Evaluation:** compute metrics and generate diagnostic plots.
6. **Deployment:** expose real-time forecasts in a Streamlit dashboard.

4.2 Model Architectures

All models accept an input of shape $(72, d)$ and output a forecast of shape $(12, m)$.

LSTM. The LSTM model uses:

- 1–2 LSTM layers with 64–128 hidden units,
- dropout between layers,
- fully connected layers to map the final sequence representation to the 12-step forecast.

It is trained with Huber (smooth L1) loss, which is more robust to outliers than pure MSE, and with the AdamW optimiser. A cosine-annealing learning-rate schedule and gradient clipping are applied. Training also uses gradient accumulation and light time-series augmentation (sequence dropout, noise injection and small scaling/warping) to improve robustness.

TCN. The TCN consists of several residual blocks with dilated 1D causal convolutions. Key hyperparameters are:

- 3–4 residual blocks,
- 32–64 filters per convolution,
- kernel size 2–4,
- dilation factors such as 1, 2, 4, 8, ...

The TCN is trained with MSE loss, AdamW optimiser and a ReduceLROnPlateau scheduler.

Transformer. The Transformer uses an encoder-only architecture with positional encodings, multi-head self-attention and feed-forward layers. Typical settings are:

- 1–2 encoder layers,
- model dimension 64–128,
- 4–8 attention heads,
- feed-forward dimension 128–256,
- dropout 0.1.

It is trained with MSE loss, AdamW and a ReduceLROnPlateau scheduler.

A compact summary of high-level hyperparameters appears in Table 4.

Table 4: High-level hyperparameters of the three deep models.

Model	Depth	Size	Training highlights
LSTM	1–2 layers	64–128 units	Huber loss, AdamW, cosine LR, dropout, clipp
TCN	3–4 blocks	32–64 filters	MSE loss, AdamW, ReduceLROnPlateau
Transformer	1–2 encoders	dim 64–128, 4–8 heads	MSE loss, AdamW, ReduceLROnPlateau, drop

4.3 Evaluation Metrics

For each model we compute:

- Mean Absolute Error (MAE),
- Root Mean Squared Error (RMSE),
- Coefficient of Determination (R^2),
- Mean Absolute Percentage Error (MAPE) when valid.

The formulas are:

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

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \quad \mathrm{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|.$$

MAE reflects typical error magnitude, RMSE emphasises large errors, and R^2 measures how much of the variance in the target is explained.

4.4 Testing Strategy and Cross-Validation

During development a rolling-origin (walk-forward) methodology is used in addition to the hold-out split:

1. Train on an initial period.
2. Validate on the following block.
3. Extend the training window to include that block.
4. Move the validation block forward and repeat.

This checks stability across different time spans and mimics continual re-training as new data arrive.

4.5 Error and Robustness Analysis

Beyond aggregate metrics, we analyse:

- residual time series $\varepsilon(t) = y(t) - \hat{y}(t)$,
- error histograms,
- cumulative error distributions (coverage at error thresholds),
- behaviour during rapid transitions (CO_2 spikes, PM peaks),
- sensitivity to missing data (simulated gaps filled by interpolation),
- sensitivity to noise (Gaussian noise added to inputs to simulate sensor drift).

5 Results and Discussion

5.1 Quantitative Summary

Tables 5 and 6 summarise the performance of the three models on the IEQ score. Values are taken from the final notebook evaluations and rounded.

Table 5: Validation-set performance for IEQ score.

Model	MSE	RMSE	MAE	R^2
LSTM	16.8	4.1	2.8	0.99
TCN	31.3	5.6	3.7	0.98
Transformer	46.7	6.8	4.8	0.97

Table 6: Test-set performance for IEQ score.

Model	MSE	RMSE	MAE	R^2	MAPE [%]
LSTM	21.5	4.6	3.6	0.86	11.7
TCN	212.9	14.6	13.1	-0.40	40.8
Transformer	30.0	5.5	4.3	0.80	30.9

On the validation set, all models achieve high R^2 values (> 0.97), with the LSTM being best. On the test set, however, the TCN collapses, the Transformer degrades moderately, and the LSTM remains robust.

5.2 LSTM Performance

On the validation set (Figure 1), the LSTM closely follows the true IEQ curve. Most observations lie within a ± 5 error band. The scatter plot is tightly clustered around the diagonal, and the error histogram is narrow.

On the test set (Figure 2), errors increase somewhat but remain moderate. The model still tracks both the overall downward trend and local fluctuations. The cumulative error curve shows that a high fraction of predictions fall within small absolute errors.

Training metrics (Figure 3) indicate healthy learning: training and validation loss decrease together and remain close; gradient norms are bounded; and the cosine learning-rate schedule avoids instability. The combination of Huber loss, augmentation and regularisation yields a model that generalises well.

5.3 TCN Performance

The TCN looks very strong on validation (Figure 4), with low error and $R^2 \approx 0.98$. However, on the test set (Figure 5) it systematically overestimates IEQ. The error distribution becomes wide and the performance summary reports a negative R^2, meaning that predicting the mean would do better.

Training curves (Figure 6) show a large gap between training and validation loss, consistent with overfitting. The TCN has enough capacity to fit the training data very well but fails to generalise to the final period, which may have slightly different patterns.

5.4 Transformer Performance

The Transformer achieves intermediate performance. On the test set (Figure 7) it captures the general trend but exhibits a positive level bias: forecasted IEQ scores are consistently higher than actual values. The error band frequently misses the true curve.

Training plots (Figures 8–10) show that validation loss is lowest at the very first epoch and then oscillates or increases, while training loss continues to decrease. This indicates early overfitting. The architecture is likely too powerful for the available data and requires stronger regularisation or more training examples.

5.5 Comparative Analysis and Hypothesis Discussion

Table 7 summarises strengths and weaknesses qualitatively.

Table 7: Qualitative comparison of the three models.

Model	Strengths	Weaknesses
LSTM	Stable, best test accuracy, interpretable training curves	Slight smoothing of extreme
TCN	Very low training/validation loss initially, fast inference	Severe overfitting, strong pos
Transformer	Captures long-range trends, flexible architecture	Overfits quickly, systematic l

In light of the hypotheses:

- **H1** (deep learning vs. classical models) is consistent with the literature, although classical baselines are not re-implemented here.
- **H2** (Transformer best overall) is *not* supported: the LSTM is clearly superior on the test set.
- **H3** (TCN vs. LSTM) is partially supported on validation but contradicted on the test set, where TCN performance collapses.
- **H4–H6** (feature engineering, multivariate inputs, six-hour window) are qualitatively supported, as all models achieve good validation performance using this configuration.

For deployment in the aquaculture facility, the LSTM model offers the best compromise between accuracy, robustness and implementation complexity.

6 Conclusion and Future Work

This project developed a complete pipeline for forecasting indoor environmental quality in a smart aquaculture facility using multivariate time-series deep learning. An LSTM, a TCN and a Transformer model were trained and compared on the same six-hour input, one-hour forecast task.

The LSTM model achieved the lowest error and the most stable generalisation, making it the preferred choice for deployment. The TCN and Transformer showed promising behaviour on validation but suffered from overfitting and bias on the final test period.

Future work could explore:

- integrating IEQ forecasts with water-quality models and fish health indicators,
- testing additional architectures (e.g. GRU, Temporal Fusion Transformer),
- performing systematic ablations on input-window length and feature sets,
- and deploying the model with on-line learning to adapt to seasonal changes.

Figures / Visual Results

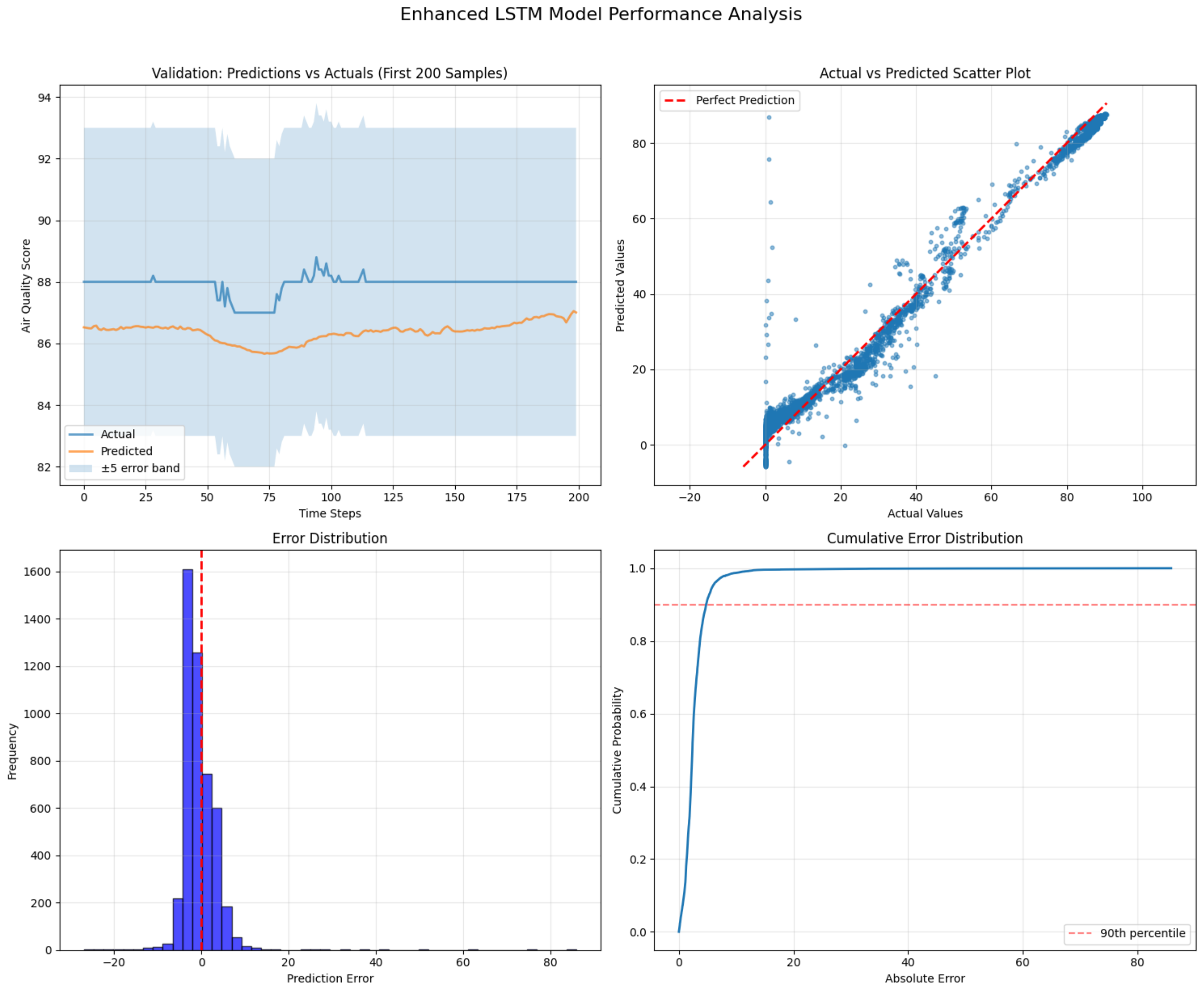

Figure 1: LSTM model performance on the validation set. Predictions vs. actual IEQ scores (top-left), scatter plot, error distribution and cumulative error distribution.

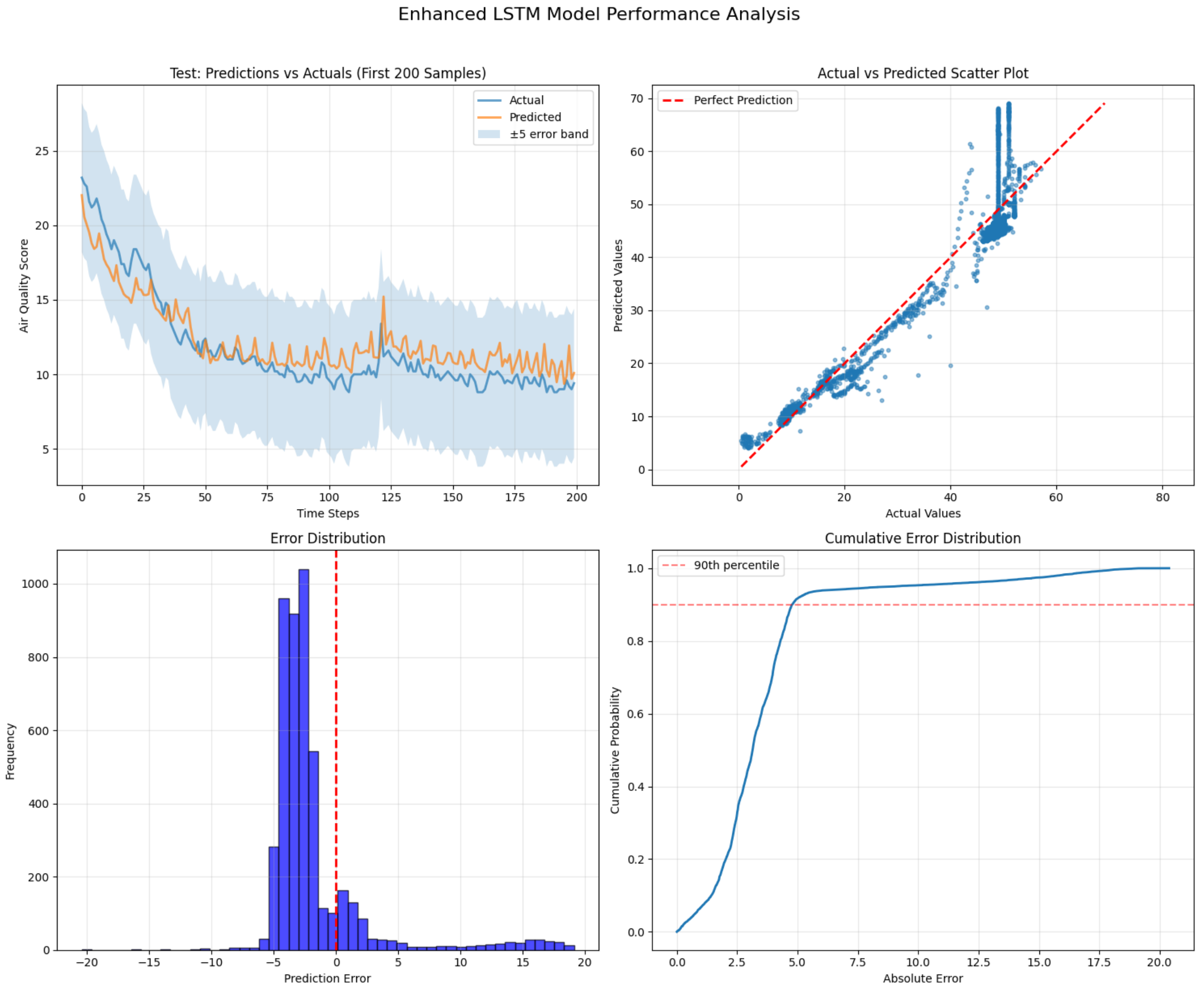

Figure 2: LSTM performance on the test set. Layout mirrors Figure 1 for direct comparison.

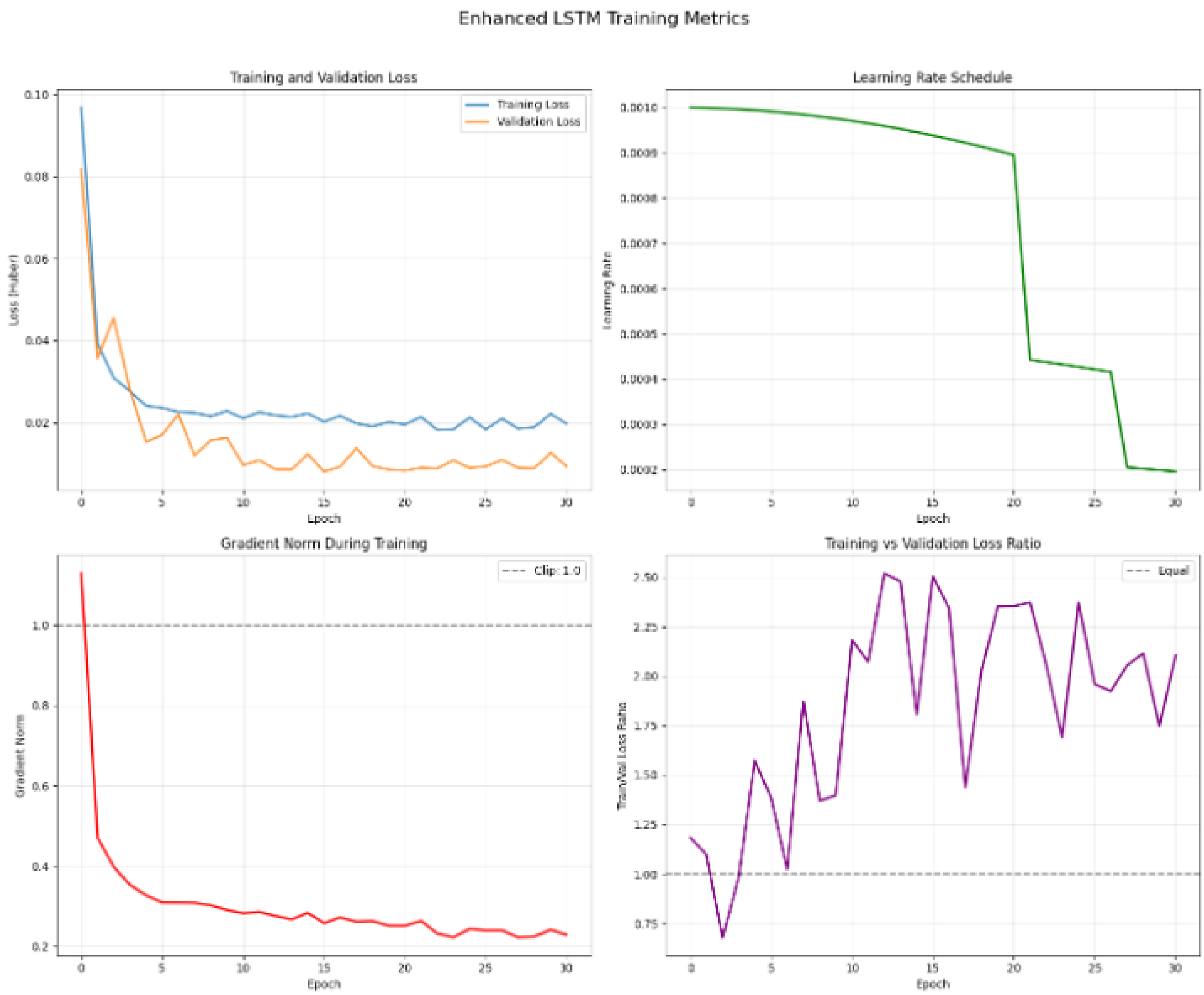

Figure 3: LSTM training metrics: training and validation loss, learning-rate schedule, gradient norm and training/validation loss ratio.

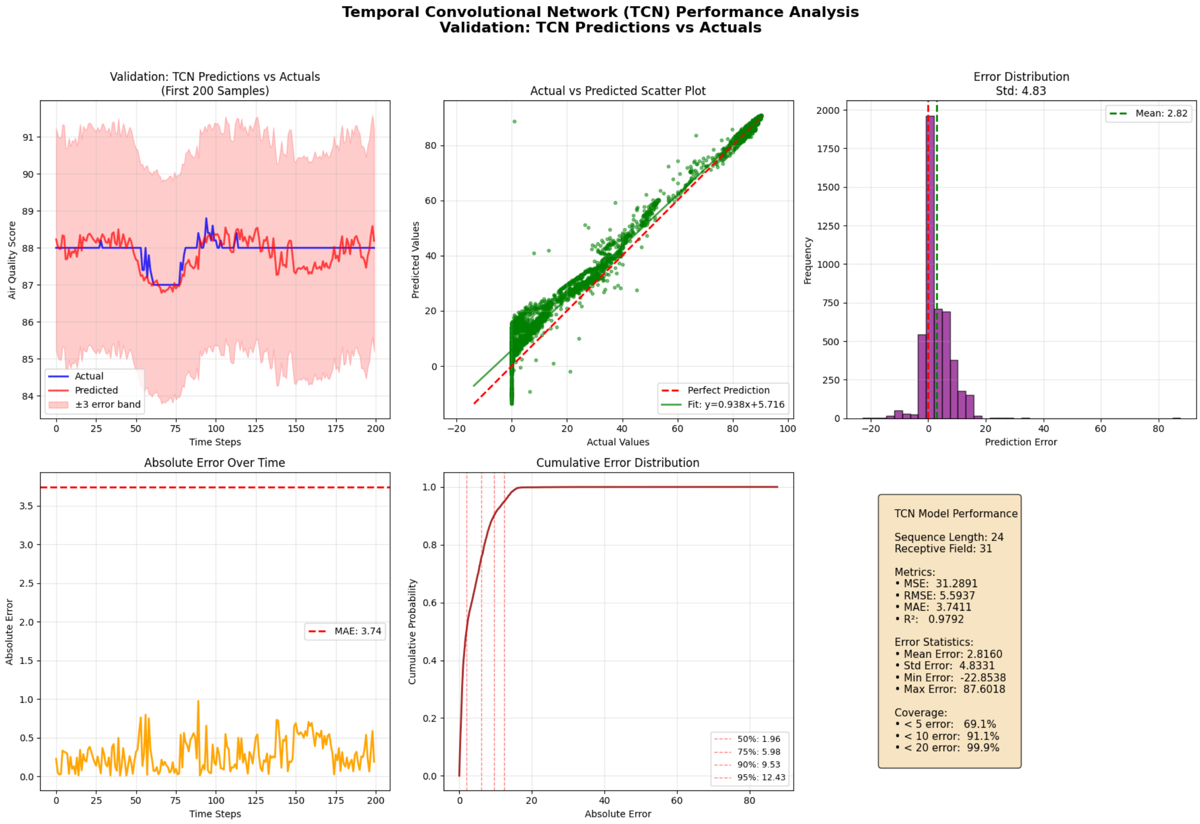

Figure 4: TCN performance on the validation set: time-series predictions vs. actual values, scatter plot, error distribution, cumulative error distribution and summary metrics.

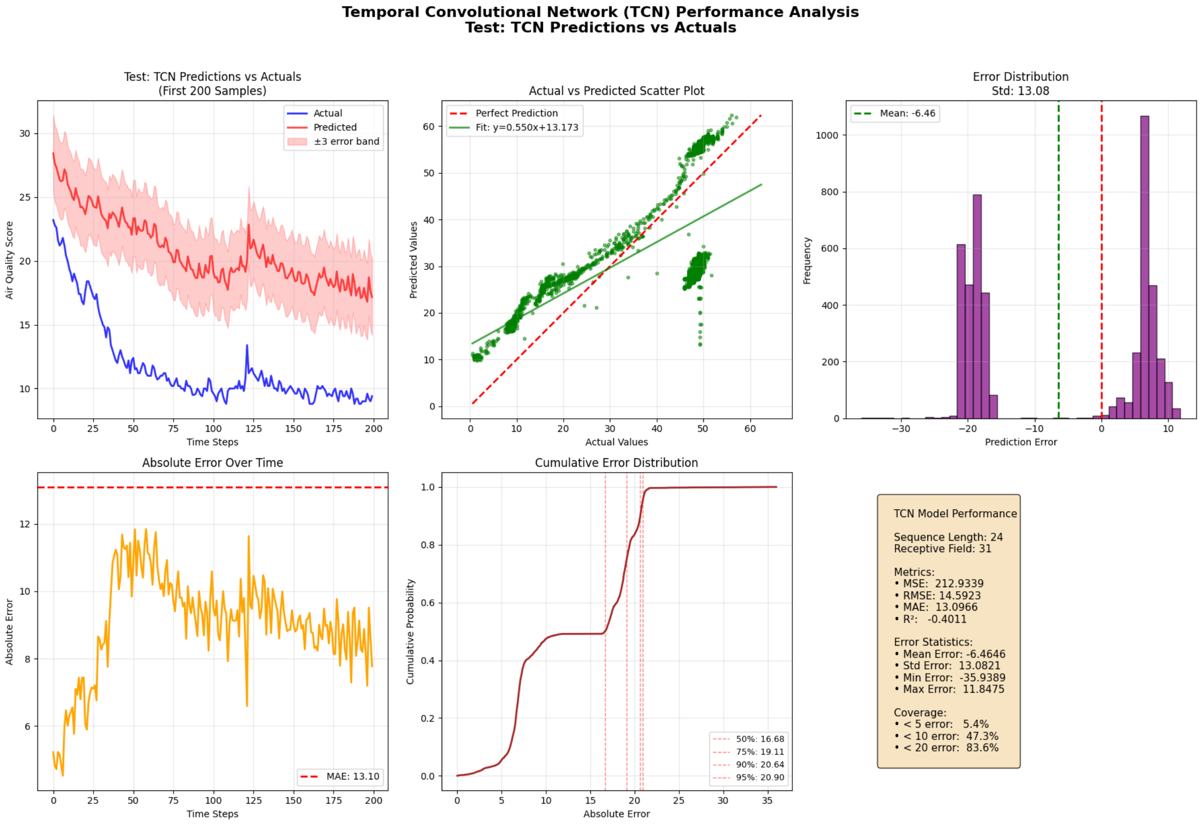

Figure 5: TCN performance on the test set. The model exhibits strong positive bias and large errors.

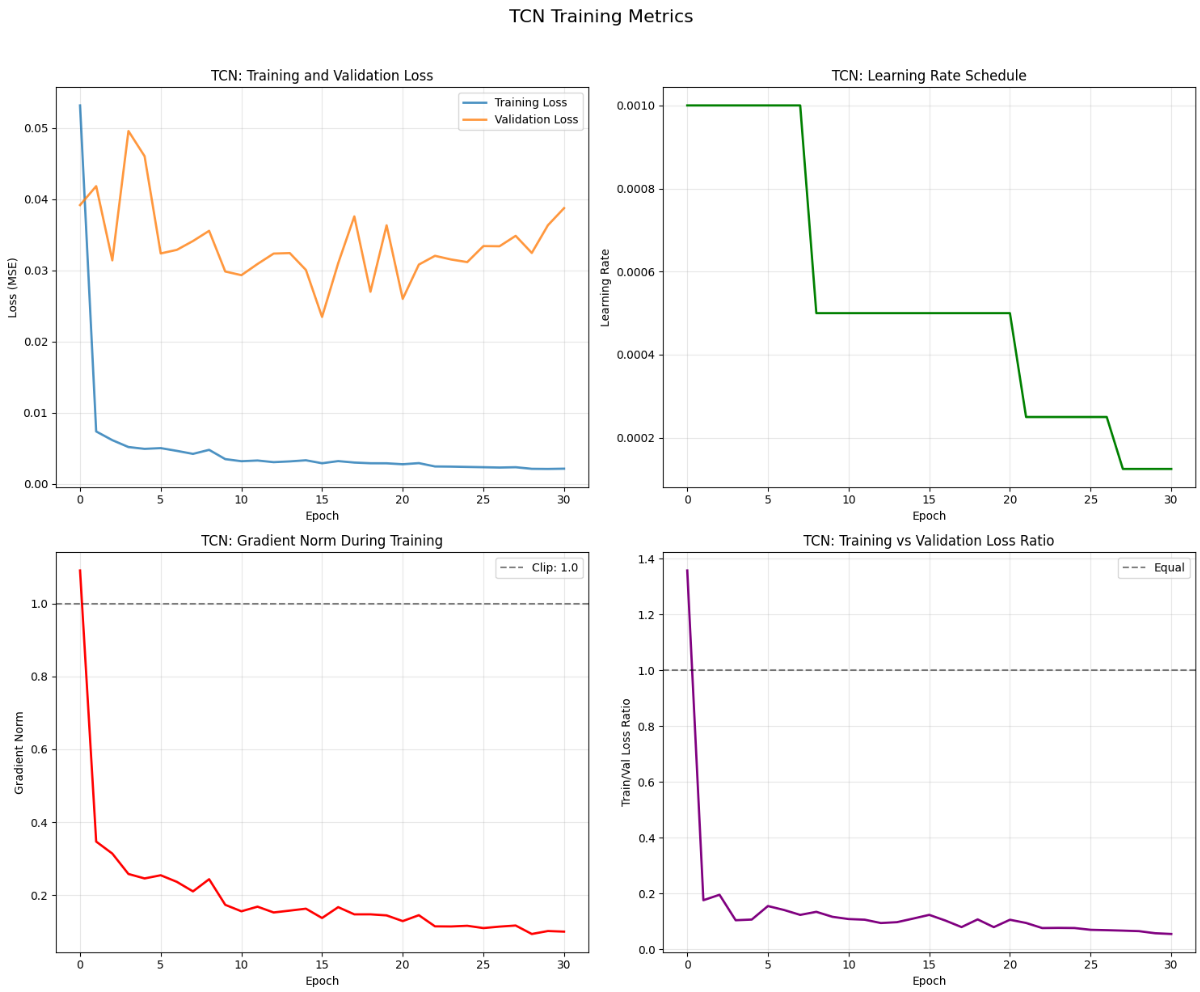

Figure 6: TCN training metrics. Training loss is much lower than validation loss, indicating overfitting.

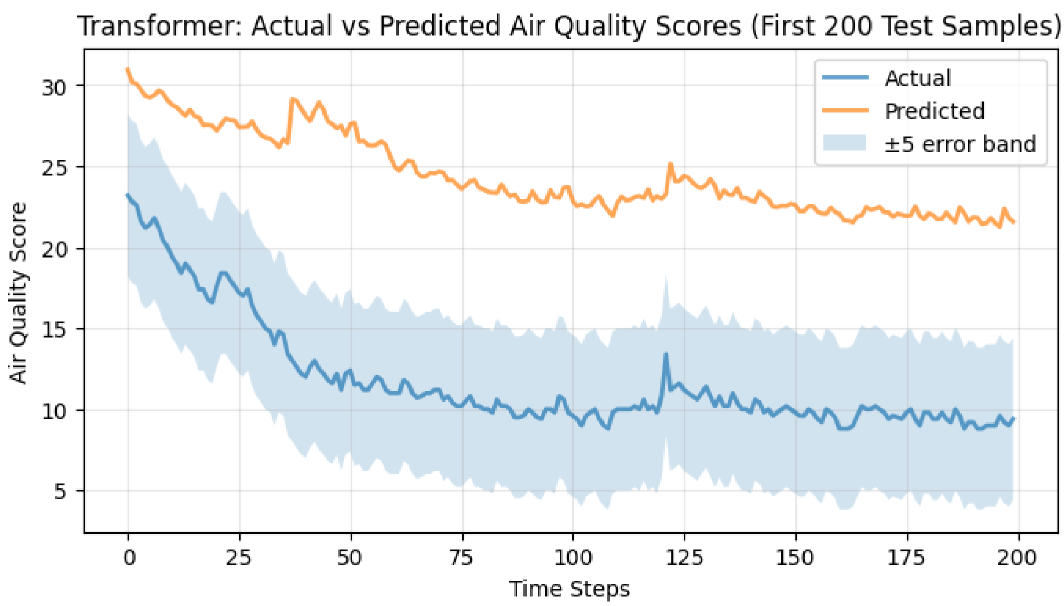

Figure 7: Transformer model: actual vs. predicted IEQ scores for the first 200 test samples. The prediction curve shows a positive level bias.

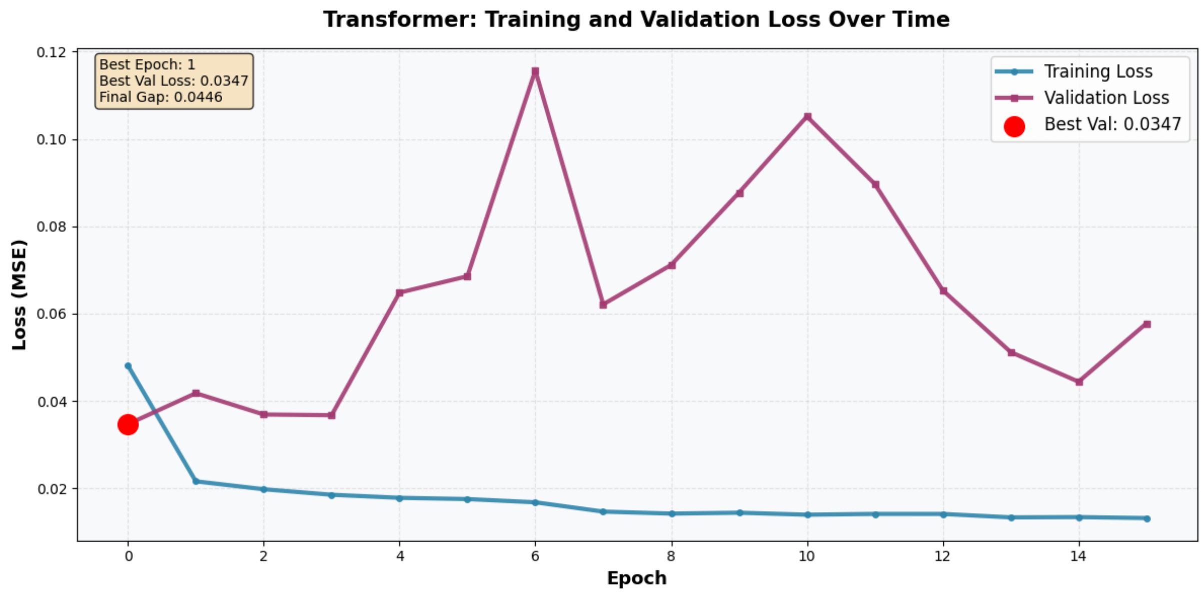

Figure 8: Transformer training and validation loss. Validation loss is lowest at the first epoch and later increases, indicating early overfitting.

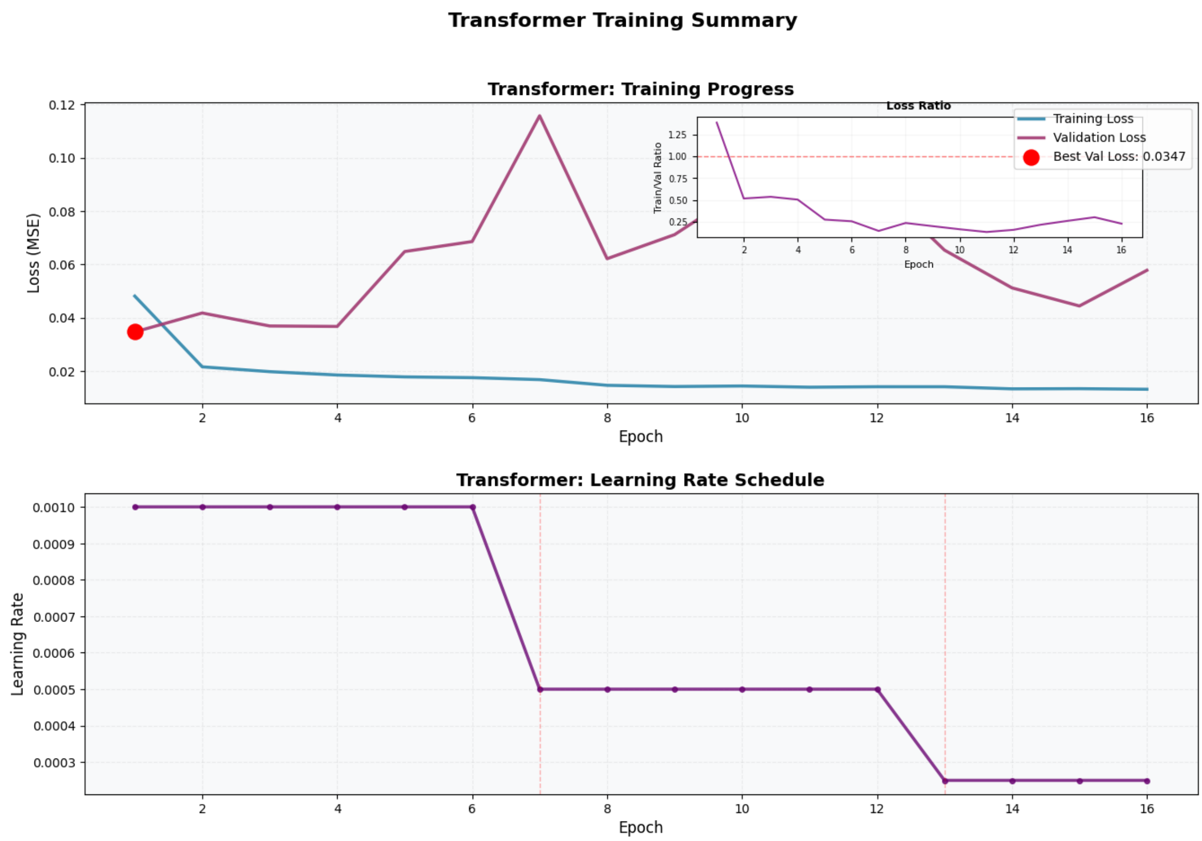

Figure 9: Transformer training summary with loss curves, loss ratio and learning-rate schedule.

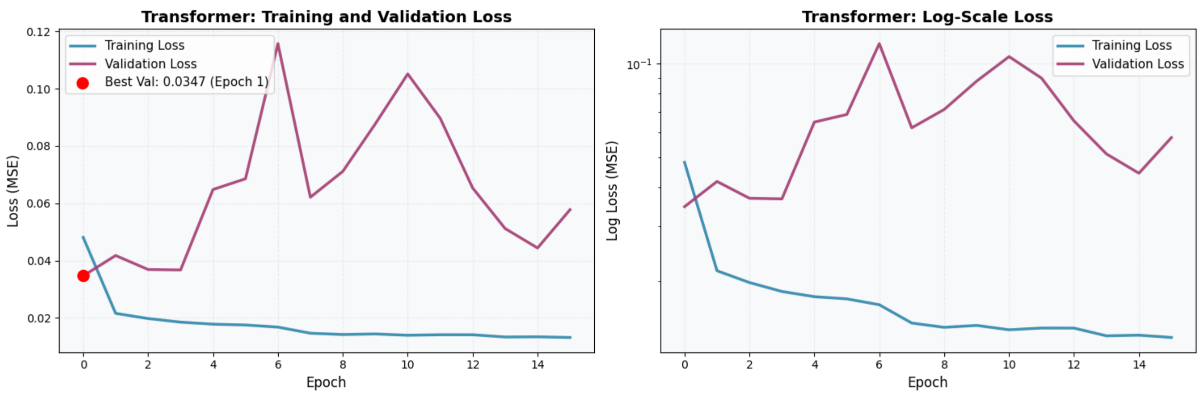

Figure 10: Transformer training and validation loss on a logarithmic scale, emphasising the gap between training and validation loss.

References

References

[1] Y. Zhu, S. A. Al-Ahmed, M. Z. Shakir, and J. I. Olszewska, "LSTM-Based IoT-Enabled CO_2 Steady-State Forecasting for Indoor Air Quality Monitoring," *Electronics*, vol. 12, no. 1, p. 107, 2023.

[2] N. Nikitas, A. Moustris, and T. Karacostas, "Short-Term Forecast of Indoor CO_2 Using Attention-Based LSTM: A Use Case in a Hospital Environment," *Sensors*, vol. 25, no. 17, 2025.

[3] H. Kim and J. Park, "AIoT-Based Indoor Air Quality Prediction for Buildings Using Enhanced CNN–LSTM," *Energy and Buildings*, 2025.

[4] M. Mousavi, S. M. Elhadidy, and H. R. Ahmadi, "Assessing and Predicting Indoor Environmental Quality in 13 Naturally Ventilated Dwellings," *Building and Environment*, 2024.

[5] Y. Sabiri, W. Houmaidi, A. Bougrine, and S. E. M. Billah, "Optimizing Indoor Environmental Quality in Smart Buildings Using Deep Learning," arXiv preprint, 2025.

[6] Y. Sabiri, W. Houmaidi, O. El Maadi, and Y. Chtouki, "AQUAIR: A High-Resolution Indoor Environmental Quality Dataset for Smart Aquaculture Monitoring," arXiv preprint, 2025.

[7] K. P. Rasheed Abdul Haq and V. P. Harigovindan, "Water Quality Prediction for Smart Aquaculture Using Hybrid Deep Learning Models," *IEEE Access*, vol. 10, pp. 60078–60098, 2022.

[8] J. Li, X. Wang, and H. Zhang, "Attention-Driven LSTM and GRU Techniques for Precise Aquaculture Water-Quality Prediction," *Aquaculture*, 2024.

[9] R. A. Haq and V. P. Harigovindan, "Intelligent Prediction and Continuous Monitoring of Water Quality in Aquaculture Using IoT and Machine Learning," *Water*, vol. 17, no. 1, 2025.

[10] L. Lopez, J. Gonzalez, and M. Velasquez, "Predictive Modelling of Aquaculture Water Quality Using IoT and Machine Learning," *Environmental Modelling & Software*, 2025.

[11] S. Sharma, P. K. Singh, and R. Kumar, "Smart Technologies in Aquaculture: An Integrated IoT, AI, and Blockchain Framework," *Computers and Electronics in Agriculture*, 2025.