RADIAN SCALING FOR CONCEPT-DRIFT-RESILIENT SHORT-TERM ELECTRICITY LOAD FORECASTING IN SMART-CITY EDGE SYSTEMS

Wenping Wang
Rolf Brühl

Reliable short-term electricity load forecasting is a foundational analytic service in digitally managed cities because demand balancing, fault anticipation, and operational resilience all depend on accurate near-term estimates of urban electricity use. In practice, however, edge-deployed forecasting models experience concept drift when seasonal demand shifts alter the scale, mean, and median of the incoming load series. This challenge is particularly acute in resource-constrained Internet-of-Things environments, where complex retraining and ensemble-maintenance schemes are often impractical. This paper presents a normalization-centered solution based on radian scaling*, a bounded angular transformation that converts consecutive load differences into values that remain within a fixed interval and thereby preserve a stable learned representation under seasonal drift. The method is evaluated on the 5-minute New York Independent System Operator Long Island zone dataset, using a deliberately drift-inducing split in which training covers 1 January 2018 to 30 April 2018 and evaluation covers 1 May 2018 to 31 December 2020. Five constrained neural architectures are assessed: recurrent neural network, long short-term memory, gated recurrent unit, temporal convolutional network, and transformer. The empirical results show that radian scaling materially improves out-of-sample robustness while also reducing convergence time. On the evaluation partition, the constrained gated recurrent unit reduces average root mean square error from 158.63 MW under the best prior normalization baseline to 43.375 MW under radian scaling, a 72.657% reduction. Across models, average early-stopping epochs fall from 56.240 to 18.320. A Friedman test confirms statistically significant differences among normalization methods on the evaluation dataset ($\chi^2 = 9.2400$, $p = 0.0263$). Framed for urban energy analytics, the findings show that a lightweight transformation layer can substantially improve smart-city forecasting resilience without changing the forecasting architecture or requiring online retraining.*

Index Terms — smart city; electricity load forecasting; concept drift; edge AI; normalization; radian scaling; urban energy management

https://doi.org/10.66033/judsc2025-223

INTRODUCTION

Urban digital infrastructure increasingly relies on short-term forecasting to support operational decisions in smart grids, building systems, and real-time municipal services. In electricity systems, short-term load forecasts contribute directly to balancing supply and demand, anticipating faults, and maintaining power quality. These capabilities are not merely technical conveniences; they are integral to the reliability and resilience of smart-city services whose continuity depends on stable energy delivery.

A recurring obstacle in real-world deployment is *concept drift*: the gradual divergence between the distribution learned during training and the distribution observed during live operation. In electricity demand forecasting, seasonal changes are a major source of this drift. Summer cooling demand and, to a lesser degree, winter heating demand alter the range and central tendency of the load series, causing models trained on a narrower regime to underperform when deployed in a shifted one. In resource-constrained edge settings, this problem is amplified. Lightweight models can be trained and executed efficiently, but their limited representational capacity makes them less able to extrapolate when the input distribution changes materially.

The conventional response is to detect drift and retrain, or to maintain adaptive and ensemble-based systems that continuously update the model. While effective, these approaches increase computational load, memory consumption, and maintenance complexity. Such burdens are poorly aligned with the operational realities of edge-deployed urban analytics, where stability and simplicity matter as much as accuracy.

This study adopts a different design principle: rather than making the model itself adaptive, it redesigns the normalization layer so that the transformed input remains well-behaved under seasonal shifts. Specifically, it formalizes and evaluates *radian scaling*, a bounded transformation that maps the difference between consecutive load observations into an angular representation. Because the transformed values are limited to the interval $(-\pi/2, \pi/2)$, the method reduces the risk that distribution drift will push the model into unlearned numerical regimes.

The contribution of this manuscript is threefold. First, it presents the mathematical basis and implementation logic of radian scaling as a lightweight normalization method for smart-city electricity forecasting. Second, it evaluates the method against min–max normalization, Z-score normalization, and robust scaling across five constrained neural network architectures. Third, it interprets the empirical findings through the lens of urban infrastructure deployment, emphasizing why normalization-centered robustness is especially valuable for edge analytics in city-scale energy systems.

RELATED WORK AND RESEARCH CONTEXT

Concept-drift mitigation strategies in time-series forecasting typically fall into two broad groups: *online adaptation* and *offline drift detection*. Online approaches dynamically update the model, often through retraining, weighting, or ensemble expansion. Offline approaches focus on detecting distributional shifts and signaling when manual retraining is necessary. Both classes can be effective, but both assume a computational and operational budget that is often unavailable in embedded or edge-deployed systems.

In electricity load forecasting, this issue is particularly salient. Seasonal changes can shift the scale and central tendency of demand while preserving a broadly similar intraday pattern. A model may therefore fail not because the temporal structure disappears, but because the normalized inputs no longer resemble the distribution seen during training. This observation suggests that the normalization layer is not a neutral preprocessing step; it is a central determinant of whether a constrained model can generalize under drift.

Traditional normalization methods behave differently under this condition. Min–max normalization is bounded but tied to the training range. Z-score normalization centers and scales the data but remains dependent on the training mean and standard deviation. Robust scaling reduces sensitivity to outliers by using the median and interquartile range, yet it still inherits distributional assumptions from the training regime. When seasonal drift changes these reference quantities, the transformed evaluation data can become skewed, and shallow models may struggle to extrapolate.

Radian scaling addresses this problem by normalizing not the raw values themselves but the *differences* between consecutive values. It then bounds these differences through an inverse trigonometric transformation. This shifts the burden of robustness away from model adaptation and into the representation layer, making the approach especially attractive for smart-city environments where lightweight deployment is a primary requirement.

MATERIALS AND EXPERIMENTAL DESIGN

Dataset and data quality

The empirical study uses the 5-minute electricity load dataset for the Long Island zone managed by the New York Independent System Operator (NYISO). Long Island is an appropriate test case for urban energy forecasting because it exhibits pronounced seasonal drift, especially during summer, when air-conditioning demand elevates peak load.

The training window spans **1 January 2018 to 30 April 2018**. The evaluation window spans **1 May 2018 to 31 December 2020**. This split intentionally withholds the primary summer drift pattern from the training set, thereby creating a realistic out-of-sample concept-drift regime for evaluation.

During data cleaning, 74 sporadic missing values were identified and imputed using second-order polynomial interpolation. In addition, 7685 outliers were detected using a multiplicative seasonal decomposition with a weekly period of 2016 observations and an interquartile-range rule. The outliers were retained to preserve the operational realism of the benchmark.

Table 1: Data quality summary for the NYISO Long Island 5-minute load series.

Data quality indicator	**Count**
Missing values	74
Detected outliers (IQR-based)	7685

Forecasting task

The forecasting setup is many-to-many. Each model uses the previous 12 time steps (1 hour of history at 5-minute intervals) to forecast the next 12 time steps (1 hour ahead). This horizon is operationally meaningful for short-term urban load management and sufficiently demanding to expose the effect of concept drift.

Models and training protocol

Five constrained neural architectures are evaluated in order to test whether the benefits of radian scaling generalize across different forecasting families:

1. Recurrent Neural Network (RNN),
2. Long Short-Term Memory (LSTM),
3. Gated Recurrent Unit (GRU),
4. Temporal Convolutional Network (TCN),
5. Transformer.

The recurrent models use a single recurrent layer with 10 units. The TCN uses one 1D convolution layer with 2 filters, kernel size 3, and dilation rate 1. The transformer is intentionally constrained to one block, one head, head size 4, feed-forward dimension 1, and a two-unit multilayer perceptron, with dropout disabled. These restrictions are deliberate: they mimic the memory and computational constraints of edge deployment and make the benchmark relevant to smart-city infrastructure rather than server-class training environments.

All models are trained for up to 300 epochs using the Adam optimizer, a learning rate of 10^{-3}, mean squared error loss, and batch size 1000. Early stopping is applied with a minimum improvement threshold of 0.0001 and patience of 3. Because neural networks are non-deterministic, each experiment is repeated five times, and the reported metrics are averaged across runs.

Table 2: Constrained model configurations used in the benchmark.

Model	**Configuration**
RNN	One SimpleRNN layer, 10 units, tanh activation, orthogonal recurrent initializer
LSTM	One LSTM layer, 10 units, sigmoid recurrent activation, tanh output activation
GRU	One GRU layer, 10 units, sigmoid recurrent activation, tanh output activation
TCN	One Conv1D layer, 2 filters, kernel size 3, dilation rate 1, ReLU activation, followed by flattening
Transformer	One transformer block, one attention head, head size 4, feed-forward dimension 1, two-unit MLP, dropout 0

METHODOLOGY

Baseline normalization methods

Three standard normalization methods are used as baselines.

Min–max normalization For a raw value v, min–max normalization rescales the data to the interval $[-1,1]$:

$$v' = 2 \times \frac{v - \min(v)}{\max(v) - \min(v)} - 1. \tag{1}$$

Z-score normalization Using the training mean $\bar{v}$ and standard deviation σ:

$$v' = \frac{v - \bar{v}}{\sigma}. \tag{2}$$

Robust scaling Using the training median $Q_2(v)$ and interquartile range $Q_3(v) - Q_1(v)$:

$$v' = \frac{v - Q_2(v)}{Q_3(v) - Q_1(v)}. \tag{3}$$

All three methods are fitted on the training data and reused unchanged on the evaluation data, simulating real deployment where the preprocessing pipeline does not have access to continuously refreshed parameters.

Radian scaling

Radian scaling begins by focusing on the difference between consecutive observations:

$$\Delta v_t = v_t - v_{t-1}. \tag{4}$$

Because time-series observations are uniformly spaced, the time-step difference is $\Delta t = 1$. A naive angular representation is therefore

$$\theta_t = \tan^{-1}(\Delta v_t). \tag{5}$$

By construction, the resulting angle is bounded:

$$-\frac{\pi}{2} < \theta_t < \frac{\pi}{2}. \tag{6}$$

This boundedness is valuable because it prevents the transformed values from growing without limit even when the raw load scale changes. However, the direct application of Equation (5) can over-concentrate values near the extremes, producing a bimodal distribution that is not ideal for simple neural models.

To address this, radian scaling introduces a dampening constant k, chosen from the average absolute difference of the sequence:

$$a = \frac{1}{n}\sum_{i=1}^{n} |\Delta v_i|, \tag{7}$$

$$k = 10^{\lfloor \log_{10}(a)+1 \rfloor}. \tag{8}$$

The normalized value is then defined as

$$v'_t = \tan^{-1}\left(\frac{\Delta v_t}{k}\right). \tag{9}$$

On the Long Island dataset, the fitted value is $k = 100$. This dampened angular representation produces a more centralized and substantially more stable distribution under concept drift than conventional scaling methods.

Inverse transformation

To recover the forecast in the original scale, the transformed values are first mapped back to differences:

$$\Delta \hat{v}_t = k \tan(\hat{v}'_t). \tag{10}$$

Given the last observed pivot v_0, the original sequence is restored through cumulative summation:

$$\hat{v}_1 = v_0 + \Delta\hat{v}_1, \tag{11}$$

$$\hat{v}_t = \hat{v}_{t-1} + \Delta\hat{v}_t, \qquad t \geq 2. \tag{12}$$

This pivot-based reconstruction is one of the reasons radian scaling remains responsive to the latest operating regime: the forecast is anchored to the most recent known value rather than to a fixed training-era scale.

Implementation logic

Algorithm 1 Radian-scaling transformation of a 1D sequence

Require: Sequence $v = [v_1, v_2, \ldots, v_n]$, pivot point v_0, optional constant k
Ensure: Transformed sequence v', fitted constant k
1: Insert pivot point at the beginning: $v_p \leftarrow [v_0, v_1, v_2, \ldots, v_n]$
2: Compute consecutive differences: $\Delta v_i \leftarrow v_{p,i+1} - v_{p,i}$ for $i = 0, \ldots, n-1$
3: **if** k is not provided **then**
4: Compute absolute average: $a \leftarrow \frac{1}{n}\sum_{i=1}^{n} |\Delta v_i|$
5: Compute constant: $k \leftarrow 10^{\lfloor \log_{10}(a)+1 \rfloor}$
6: Transform each difference: $v'_i \leftarrow \tan^{-1}(\Delta v_i / k)$
7: **return** v', k

Algorithm 2 Inverse transformation back to the original scale

Require: Transformed sequence $v' = [v'_1, v'_2, \ldots, v'_n]$, pivot point v_0, constant k
Ensure: Restored sequence $v = [v_1, v_2, \ldots, v_n]$
1: Compute inverse differences: $\Delta v_i \leftarrow \tan(v'_i) \times k$
2: Restore first element: $v_1 \leftarrow v_0 + \Delta v_1$
3: **for** $i = 2$ to n **do**
4: $v_i \leftarrow v_{i-1} + \Delta v_i$
5: **return** v

Conceptual interpretation

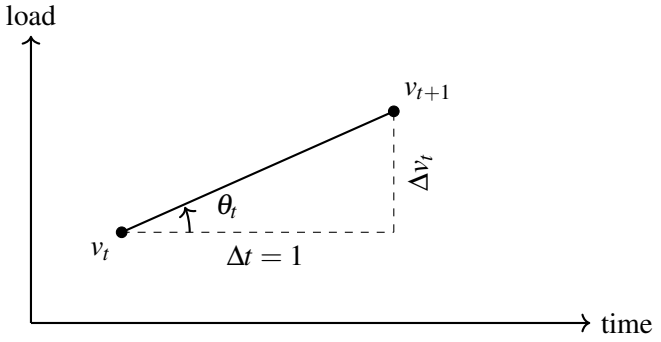

Figure 1: Geometric interpretation of radian scaling: the model learns the bounded angle induced by the difference between two consecutive observations rather than the raw value scale itself.

RESULTS

Training efficiency

A central practical result is that radian scaling reduces the number of epochs required before early stopping is triggered. This matters in edge-oriented workflows because faster convergence reduces development time and lowers the computational cost of retraining when retraining is unavoidable.

Table 3: Average training epochs before early stopping on the training dataset.

Model	**Min–Max**	**Z-Score**	**Robust**	**Radian**
RNN	49.600	105.80	75.800	18.400
LSTM	42.000	79.800	63.800	16.000
GRU	47.600	94.000	67.000	13.000
TCN	70.200	106.40	86.200	24.000
Transformer	71.800	94.600	105.20	20.200
Average	56.240	96.120	79.600	**18.320**

Relative to the average min–max baseline, radian scaling reduces the average epoch count from 56.240 to 18.320, corresponding to a 67.425% reduction. This shows that the method is not only robust at inference time but also efficient during model preparation.

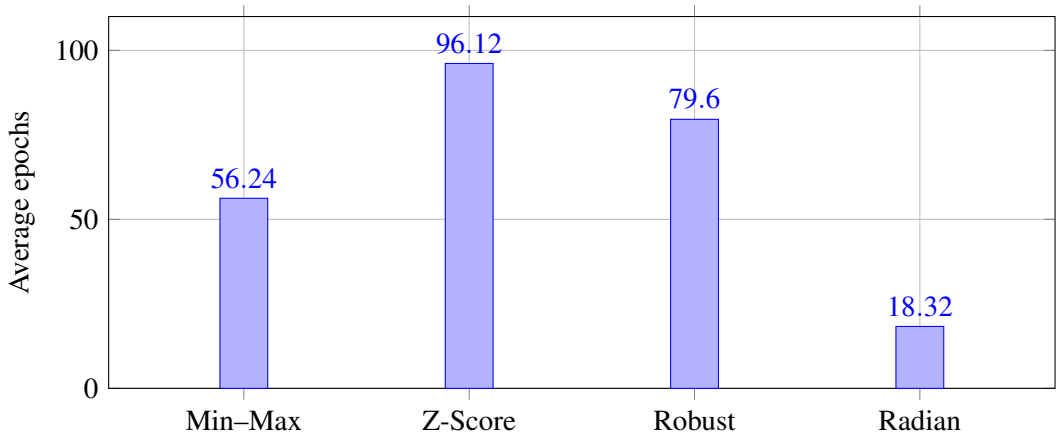

Figure 2: Average training epochs across all five models. Radian scaling reaches early stopping substantially faster than the comparison normalizations.

Training-set forecasting accuracy

On the training set, all normalization methods perform competitively. The relatively small differences indicate that, within the regime represented during fitting, the constrained models can learn effectively under multiple preprocessing schemes.

Table 4: Average forecasting accuracy on the training dataset.

Model	Metric	Min–Max	Z-Score	Robust	Radian
RNN	R^2	0.9789	0.9870	0.9830	0.9874
	RMSE	49.386	38.955	44.622	38.406
LSTM	R^2	0.9818	0.9875	0.9863	0.9871
	RMSE	46.212	38.297	40.049	38.856
GRU	R^2	0.9837	0.9884	0.9879	0.9868
	RMSE	43.729	36.888	37.682	39.353
TCN	R^2	0.9828	0.9863	0.9855	0.9872
	RMSE	44.873	40.081	41.231	38.783
Transformer	R^2	0.9821	0.9820	0.9847	0.9819
	RMSE	45.745	45.763	42.324	44.626

The Friedman test applied to the training RMSE ranks yields $\chi^2 = 7.3200$ with $p = 0.0624$, indicating that the differences on the training partition are not statistically significant at the 0.05 level. This is an important benchmark result: the method does not depend on artificially improving in-sample performance to achieve better generalization later.

Evaluation-set forecasting accuracy under concept drift

The decisive evidence appears on the evaluation partition, where the models encounter sustained seasonal drift. Here the advantage of radian scaling becomes pronounced.

Table 5: Average forecasting accuracy on the evaluation dataset.

Model	Metric	Min–Max	Z-Score	Robust	Radian
RNN	R^2	0.9401	0.8892	0.8248	0.9960
	RMSE	168.80	230.16	288.64	43.485
LSTM	R^2	0.9224	0.8747	0.8873	0.9957
	RMSE	192.69	244.83	232.15	45.231
GRU	R^2	0.9474	0.8889	0.9002	0.9961
	RMSE	158.63	228.94	217.71	**43.375**
TCN	R^2	0.9479	0.9948	0.9903	0.9960
	RMSE	111.67	49.697	63.627	43.477
Transformer	R^2	0.8080	0.9566	0.8804	0.9939
	RMSE	283.43	107.57	235.49	51.694

The strongest single result is the constrained GRU with radian scaling, which reduces RMSE from 158.63 MW under the best prior baseline (min–max normalization) to 43.375 MW. This corresponds to a 72.657% reduction:

$$\frac{158.63 - 43.375}{158.63} \times 100 = 72.657\%. \tag{13}$$

The result is not confined to one model family. Every architecture achieves its best evaluation RMSE under radian scaling, with the TCN also performing strongly. The TCN's relative resilience under conventional

normalization is consistent with its ReLU activation, which avoids the saturation effects that can constrain tanh-based recurrent models when shifted inputs exceed the regime observed during training.

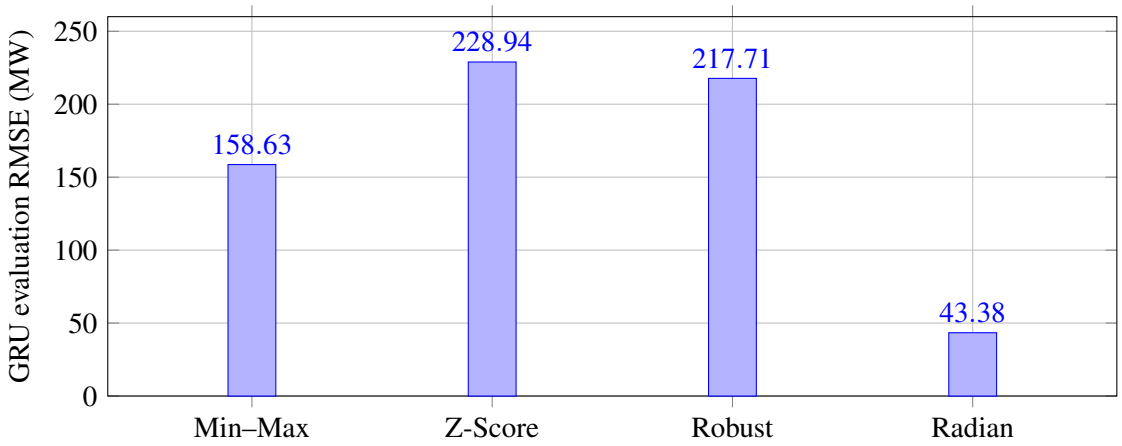

Figure 3: Evaluation RMSE of the constrained GRU across normalization methods. The angular representation produces a marked reduction in forecast error under seasonal drift.

Statistical comparison

To assess whether the performance differences across normalization methods are systematic rather than anecdotal, rank-based statistical tests were applied.

Table 6: Statistical tests comparing normalization methods.

Test	**Statistic**	***p*-value**	**Interpretation**
Friedman test on training RMSE ranks	7.3200	0.0624	No significant difference on the training partition
Friedman test on evaluation RMSE ranks	9.2400	0.0263	Significant difference among normalization methods under concept drift
Nemenyi: Radian vs Min–Max	—	0.1221	Radian scaling ranked better, but pairwise difference is not significant at 0.05
Nemenyi: Radian vs Z-Score	—	0.0682	Radian scaling ranked better; directional separation but not significant at 0.05
Nemenyi: Radian vs Robust	—	0.0355	Radian scaling significantly outperforms robust scaling

The evaluation Friedman test rejects the null hypothesis of equal rank performance, confirming that normalization choice materially affects out-of-sample robustness when concept drift is present. The Nemenyi post hoc comparison places radian scaling as the best-ranked method overall. At the conventional 0.05 threshold, the strongest pairwise evidence is against robust scaling, while comparisons against Z-score and min–max normalization remain directionally favorable but not individually significant. Taken together, the results support the conclusion that radian scaling is the most effective normalization strategy in this benchmark.

DISCUSSION

Why radian scaling improves drift robustness

The principal mechanism is representational stability. Conventional normalizations inherit their parameters directly from the training set, so the transformed evaluation data can change shape when the underlying scale, mean, or median shifts. Under seasonal drift, this causes the neural network to receive inputs that differ materially from what it learned during training.

Radian scaling avoids this by encoding the *change* between consecutive observations instead of the absolute magnitude of the signal. The inverse tangent function then bounds these changes within a fixed interval, preventing the transformed values from exploding as the raw load scale increases during summer peaks. As long as the local temporal pattern remains structurally similar, the model continues to receive inputs that fall within a familiar numerical regime.

Why the gains are operationally important for smart cities

The urban significance of the result lies in its simplicity. Smart-city energy analytics often operate in settings where infrastructure must be stable, lightweight, and easy to maintain. A normalization method that improves drift tolerance without requiring model retraining, architectural redesign, or ensemble maintenance reduces operational burden while preserving forecasting quality.

This is especially valuable for:

- edge-deployed grid monitoring,
- municipal energy management dashboards,
- district-level demand balancing,
- resilient forecasting in constrained IoT gateways.

By improving both convergence speed and evaluation robustness, radian scaling offers a rare combination: lower cost in model preparation and better performance during deployment.

Limitation

The most important current limitation is the method's dependence on imputation. Because the transform relies on consecutive differences and the inverse transform relies on cumulative restoration from a pivot, missing values must be filled before transformation. In the benchmark, linear interpolation is preferred for this role because of its lower computational cost, even though higher-order spline and polynomial methods may offer better local interpolation accuracy in some settings.

This limitation does not negate the value of the method, but it does define its next developmental frontier. A future version that handles missing values directly would make the approach more robust in unstable real-world telemetry pipelines.

IMPLICATIONS FOR THE JOURNAL OF URBAN DEVELOPMENT AND SMART CITIES

The manuscript fits the scope of an urban-development and smart-cities venue because it addresses a core layer of smart-city infrastructure: reliable, computationally feasible analytics for city-scale electricity demand. The work is not merely a machine-learning exercise. Its contribution is tightly connected to urban service continuity, infrastructure resilience, and deployable analytics for digital city operations. The results therefore speak directly to the design of practical urban intelligence systems, especially where computational resources are limited but service reliability remains critical.

CONCLUSION

This study demonstrates that concept drift in short-term electricity load forecasting can be mitigated effectively through the normalization layer rather than through increasingly complex adaptive model-management schemes. Using a bounded angular transformation of consecutive load differences, radian scaling preserves a stable learned representation even when seasonal changes shift the scale and central tendency of the raw series.

On the NYISO Long Island 5-minute dataset, the method delivers two practically important gains. First, it shortens training substantially, reducing the average epoch count from 56.240 to 18.320 across five constrained neural models. Second, it markedly improves out-of-sample forecasting under seasonal drift. The strongest result is observed for the constrained GRU, whose evaluation RMSE drops from 158.63 MW to 43.375 MW.

The broader implication is methodological and operational. In smart-city edge systems, where complex online retraining is often infeasible, carefully designed normalization can be a powerful route to resilience. Future work should prioritize three directions: (i) imputation-free variants that handle missing values directly, (ii) transfer of the same idea to other urban sensing domains such as traffic prediction and air-quality forecasting, and (iii) integration with broader resilient forecasting networks that address both concept drift and system instability.

DATA AVAILABILITY STATEMENT

The electricity load data used in this study are publicly available from the New York Independent System Operator at `https://www.nyiso.com/load-data`.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] New York Independent System Operator. *Load Data*. Available online: `https://www.nyiso.com/load-data`.

[2] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks for action segmentation and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 156–165, 2017.

[3] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

[5] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937.

[6] M. A. Terpilowski. scikit-posthocs: Pairwise multiple comparison tests in Python. *Journal of Open Source Software*, 4(36):1169, 2019.

[7] S. Herbold. Autorank: A Python package for automated ranking of classifiers. *Journal of Open Source Software*, 5(48):2173, 2020.

[8] X. Wen, J. Liao, Q. Niu, N. Shen, and Y. Bao. Deep learning-driven hybrid model for short-term load forecasting and smart grid information management. *Scientific Reports*, 14:13720, 2024.

[9] M. Shaygan, C. Meese, W. Li, X. Zhao, and M. Nejad. Traffic prediction using artificial intelligence: Review of recent advances and emerging opportunities. *Transportation Research Part C: Emerging Technologies*, 145:103921, 2022.

[10] M. Méndez, M. G. Merayo, and M. Núñez. Machine learning algorithms to forecast air quality: A survey. *Artificial Intelligence Review*, 56:10031–10066, 2023.

[11] M. H. Bin Kamilin and S. Yamaguchi. Resilient electricity load forecasting network with collective intelligence predictor for smart cities. *Electronics*, 13:718, 2024.

Wenping Wang, ESCP Business School, Heubnerweg 8-10, 14059, Berlin, Germany

Rolf Brühl, ESCP Business School, Heubnerweg 8-10, 14059, Berlin, Germany

Manuscript Published; 21 October 2025.