Time Segmented Deep Forecasting of Power Grid Frequency Deviations

Casi Setianingsih
Department of Electric Power Engineering
Budapest University of Technology and Economics
Budapest, Hungary
setianingsihcasi@edu.bme.hu

Bálint Hartmann
Department of Electric Power Engineering
Budapest University of Technology and Economics
Budapest, Hungary
hartmann.balint@vik.bme.hu

***Abstract*— Frequency deviations reflect short-term power imbalance and contain information about system-wide synchronization and time-delayed coupling. We analyze about one month of synchronized transmission-grid frequency measurements from 17 monitoring points using lagged Pearson cross-correlation for lags 1–10, reported for two aggregations: an all-month baseline and a weekday–weekend split. Weekends show higher coherence than weekdays across all evaluated lags (e.g., at lag 1 the global mean correlation is 0.258 on weekends versus 0.174 on weekdays). From the lag sweep, we extract peak correlation and best lag to summarize an empirical delay structure; most pairs peak at small lags, while some corridors require larger lags (≈8–10 samples) for maximum alignment. To connect these coupling patterns to forecasting design, we run a controlled univariate benchmark on a reference station (24,083,128 samples in Feb 2025, 0.1 s sampling) using LSTM, TCN, and PatchTST under the same time split, train-only preprocessing, and early stopping, with window lengths k∈{1,3,10,20}. Increasing the window from k=1 to k=20 improves accuracy for all models and reduces architecture sensitivity; at k=20 the three models converge to nearly the same nRMSE (≈11.5%).**

Keywords—power grid frequency, lagged cross correlation, LSTM, PatchTST, Temporal Convolutional Networks.

I. INTRODUCTION

Power grid frequency deviations indicate short-term power imbalance and support operational monitoring and stability assessment [1]–[3]. Wide-area monitoring relies on time-synchronized measurements to observe system dynamics and to support disturbance-aware decisions beyond conventional supervisory signals [1]–[3].

Robust frequency analytics from wide-area measurements remain challenging. Synchronized streams can be incomplete across stations and days, and availability gaps can bias correlation analysis or reduce usable training data [4]–[6]. In addition, empirical studies report non-Gaussian frequency statistics and regime-dependent correlation structure, so modeling and evaluation should match operating regimes [7]. Finally, dynamical heterogeneity implies that station responses can differ substantially; overly uniform assumptions may hide spatio-temporal effects in measured dynamics [8].

These observations motivate correlation analysis beyond a single lag. Lagged dependencies can reflect effective delays that are consistent with finite-speed electromechanical disturbance propagation, linking measurement-based coupling to time-delayed system response [9]. Our earlier 24-hour study showed that lag design can materially affect short-horizon frequency prediction; here we extend the analysis to an approximately one-month dataset and quantify how lagged coupling and dominant delays vary under an all-month baseline and a weekday–weekend regime split [10].

To connect grid-level lag patterns to predictive modeling, we run a controlled deep-forecasting benchmark with a fixed training and evaluation protocol. We evaluate multiple architectures under the same window settings (k=1, 3, 10, 20) to isolate the effect of window length from model choice [10], [11]. The benchmark compares LSTM, PatchTST, and TCN using the same time-based split, preprocessing, and early stopping, and includes training-setting checks to separate the impact of larger k from longer training [10], [11]. These architectures are widely used sequence baselines, including temporal convolution models and patch-based transformers for time-series prediction [11], [12]. We also cite a ConvLSTM-based frequency forecasting study to position our baselines within the broader grid-frequency literature [13].

II. DATA DESCRIPTION

This study uses synchronized grid frequency measurements from the Hungarian transmission system operator (MAVIR) over about one month from 17 measurement points (Station A to Station Q). Availability is checked per segment because the synchronized set is not always complete across days, which can affect wide-area analytics and learning datasets if ignored [14], [15]. To process the high-density stream efficiently, XML records are converted into time-based shards and stored as compact binary arrays to reduce parsing and I/O overhead, consistent with scalable data-processing practice [16], [17]. For the univariate forecasting benchmark, we select a baseline station with zero missing rate and non-flat variability; Station E is used, yielding 24,083,128 samples over Feb 2025 with a 0.1 s sampling step.

III. METHODS

This section summarizes the forecasting formulation, the lagged-correlation component, and the model implementations used in the univariate benchmark. The models are evaluated three sequence models under the same split and preprocessing pipeline: an LSTM regressor, a temporal convolutional network (TCN) [18], and PatchTST [19], all implemented in the same PyTorch framework.

A. Forecasting Setup and Window Length

Forecasting is posed as one-step-ahead regression on the frequency deviation series Δf at the baseline station (Section II). At time index i, the input is a context window containing the most recent k samples, and the target is the next sample. The parameter is treated k as an experimental factor and evaluate k=1, 3, 10, and 20. In the implementation, the command-line argument \`lags\` maps to the context window as: window=1 when lags=0, otherwise window=lags. Thus, the effective context length equals the specified lag window.

B. Lagged Correlation Features

Lagged Pearson cross-correlation is computed lagged Pearson cross-correlation for each station pair for k=1 to k=10, where k is a shift of k samples. Because the sampling interval of the synchronized stream can vary by dataset, we report lags in samples and interpret them as effective delays. For each pair, we extract (i) the peak correlation over the lag sweep and (ii) the corresponding best lag, which is standard in correlation-based delay estimation [20]. Results are reported correlation summaries for two aggregations only: (1) an all-month baseline and (2) a weekday–weekend split. The all-month baseline provides a robust estimate of lag-aligned coupling that is less sensitive to short-lived fluctuations or sparse segments. The weekday–weekend split is a minimal but operationally meaningful regime contrast because demand and dispatch conditions systematically differ between working days and rest days (e.g., office and school loads), which is reflected in consumption profiles and frequency-quality statistics [21]–[23].

C. Data Split and Preprocessing

All benchmark runs use the same chronological split with two fixed cut points, producing training, validation, and test periods. The split is identical for all models and all window settings. Preprocessing is fit on the training period only. First, extreme Δf outliers are clipped using training quantiles (0.0001 and 0.9999), and the resulting thresholds are applied unchanged to validation and test. Second, the clipped series is min–max scaled using parameters computed from the training period only, then applied unchanged to validation and test. This train-only fitting rule follows standard time-series practice and reduces information leakage through preprocessing or model selection [24]. Fig. 1 summarizes the split and sample counts used in all runs.

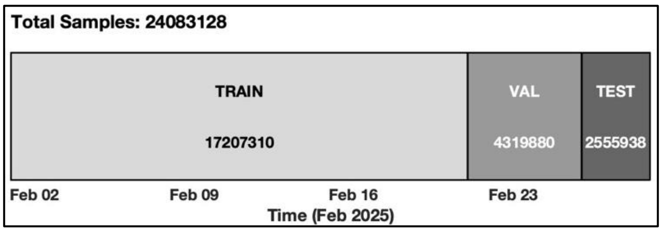

Fig. 1. Time-based split of the univariate benchmark dataset (Station E) showing total samples and the train/validation/test partitions over Feb. 2025.

D. Model Architectures

All models predict a single next step value from a fixed length context window and differ only in the sequence encoder.

a) LSTM regressor

This study uses a stacked LSTM as a recurrent baseline, followed by a linear regression head. LSTM gating supports learning long-range dependencies via an explicit memory mechanism [25]. The model takes inputs of shape [B, k, 1], returns hidden outputs for all time steps, selects the final output vector, and maps it to a scalar prediction with a linear layer. Default settings are hidden_size=64 and num_layers=2.

b) TCN regressor

The TCN model follows a residual temporal convolution design with dilated convolutions that expand the receptive field while remaining fully parallelizable over the time axis [18]. The implementation uses a stack of Temporal Block modules. Each block applies two 1D convolutions with ReLU and dropout, uses causal padding followed by a chomp operation to preserve causality, and adds a residual connection with an optional 1 by 1 convolution for channel matching. Dilation is set as 2^i per block. Default settings are channels=(32,32,32), kernel size ksz=3, and dropout 0.1. The final hidden state is taken at the last time index and passed through a linear head to produce the next step prediction.

c) PatchTST regressor

PatchTST is implemented as a transformer based forecaster that tokenizes the context window into patches along the time axis, reducing token length and enabling attention based learning over patch representations [19]. In the univariate setting, the input has shape [B,1,k]. where B is the batch size and k is the context window length in samples. The time axis is patchified using a patch length (patch_len) and sliding step (stride) via an unfold operation, producing $n$patches patch tokens given by

$$n_{patches} = \left\lfloor \frac{k - patch\_len}{stride} \right\rfloor + 1 \quad (1)$$

Each patch is linearly projected to $d$model (token embedding dimension), a learnable positional embedding is added, and the resulting token sequence is processed by a TransformerEncoder with $n$layers (number of encoder layers) and $n$heads (number of attention heads). The regression head predicts the next step value from the final token representation. Default settings are $d$model=64, $n$heads=4, $n$layers=2, and dropout 0.1.

E. Training and Evaluation Protocol

All architectures are trained with the same objective and evaluation setup. Mean squared error is minimized using Adam with a fixed learning rate $lr=1\times10^{-3}$ [26]. Training runs for a fixed epoch budget with early stopping monitored on validation MSE using a patience of 2, and the best validation checkpoint is used for test evaluation [27]. Performance is reported using nRMSE on inverse scaled predictions in the clipped Δf domain, using the same split and preprocessing for all architectures and all k values.

IV. Results and Analysis

A. Operating Regimes: Weekday vs Weekend Analysis

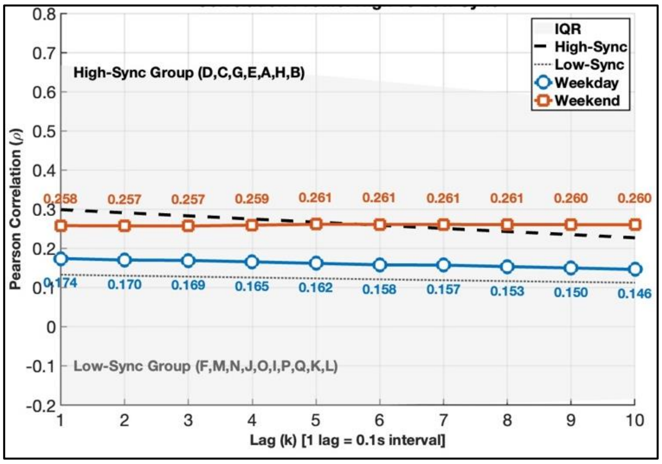

Fig. 2. Mean lagged correlation for weekday vs. weekend regimes, shown for the High-Sync and Low-Sync station groups.

Grid synchronization differs systematically between operating regimes. Fig. 2 shows higher coherence on weekends than on weekdays across all evaluated lags. At lag 1, the global mean correlation is 0.258 on weekends versus

0.174 on weekdays. A plausible explanation is that weekday operation introduces additional variability through demand-side fluctuations and operational actions that reduce alignment across monitoring points [28], [29].

A persistent feature is the gap between the High-Sync and Low-Sync station groups. The High-Sync Group (D, C, G, E, A, H, B) shows strong mutual alignment, consistent with tight coupling through dominant transmission paths and closely aligned measurements for wide-area monitoring. The Low-Sync Group (F, M, N, J, O, I, P, Q, K, L) exhibits lower correlation to the main reference behavior, suggesting stronger influence from local dynamics or regional variations rather than a single shared trend. Importantly, this separation remains visible on both weekdays and weekends: operating regimes shift overall correlation levels, but the relative ordering of consistently high- vs low-synchronization behavior is stable. This is compatible with evidence that dynamic responses are shaped by the physical interconnection structure [30] and the wide-area monitoring context [31].

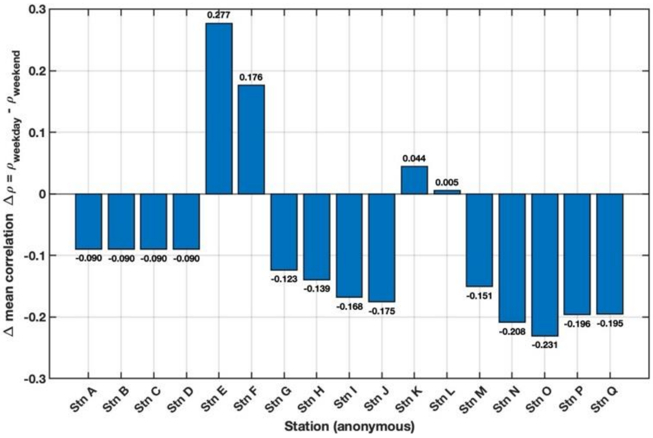

Fig. 3. Per-station weekday–weekend regime shift in mean cross-correlation at lag 1 ($\Delta\bar{\rho} = \bar{\rho}_{\mathrm{wd}} - \bar{\rho}_{\mathrm{we}}$); negative values indicate stronger average coupling on weekends.

Define the station-level mean coupling at lag 1 as

$$\bar{\rho}_i^{(r)}(1) = \frac{1}{N-1}\sum_{j=1,\, j\neq i}^{N} \rho_{ij}^{(r)}(1) \tag{2}$$

where i and j index stations, N is the number of stations, $\rho_{ij}^{(r)}(1)$ is the Pearson cross-correlation between stations i and j at lag k=1, computed separately on the weekday subset ($r = \mathrm{wd}$) and the weekend subset ($r = \mathrm{we}$); diagonal terms ($i = j$) are excluded. The regime shift is $\Delta\bar{\rho}_i = \bar{\rho}_i^{(\mathrm{wd})}(1) - \bar{\rho}_i^{(\mathrm{we})}(1)$, with $\Delta\bar{\rho}_i < 0$ indicating stronger weekend coupling and $\Delta\bar{\rho}_i > 0$ indicating stronger weekday coupling. This lagged cross-correlation formulation is consistent with standard correlation-based alignment practice [20]. As shown in Fig. 3, most stations exhibit negative $\Delta\bar{\rho}$, confirming that the weekday–weekend coherence gap is broadly distributed across the monitoring set. The strongest weekend-favored shifts occur at Station O (-0.231) and Station N (-0.208), followed by Station P (-0.196) and Station Q (-0.195), while Station L (+0.005) and Station K (+0.044) show near-invariant behavior. In contrast, Station E (+0.277) and Station F (+0.176) exhibit higher mean coupling on weekdays, highlighting heterogeneous station-level sensitivity despite the overall increase in weekend coherence. Overall, this station-level pattern is consistent with the negative off-diagonal global mean Δρ (weekday minus weekend) at lag 1 (global mean Δρ ≈ -0.0849).

B. *Spatial Interaction and Delay Structure: All Month Analysis*

Spatial interaction is visualized spatial interaction using four reference sources (Station D, Station M, Station A, and Station K) and a fixed station ordering on the horizontal axis (used only for visualization). The peak-correlation map reveals clear coupling heterogeneity. Several interactions remain strongly aligned, while Station F shows counter-swing behavior; for example, its peak correlation drops to −0.83 relative to Station M, indicating pair-specific dynamics that can oppose the dominant system trend. Marker colors encode the best lag, showing that the lag of maximum alignment varies by pair. Small-delay behavior appears for close pairs such as Station D–Station C, which peak at lag 1. Larger effective delays occur toward the south-east end (e.g., involving Station K and Station L), where peak alignment requires lags 8–10. These maps provide a month-long baseline delay structure referenced in IV-D. Fig. 4 provides the corresponding peak-correlation and best-lag map for these four references, making the coupling heterogeneity and the pair-dependent delay structure visually explicit. A fixed station ordering is used only to arrange the station-pair results in the figures for readability. This ordering is a visualization choice and does not affect the computed correlations or best-lag values.

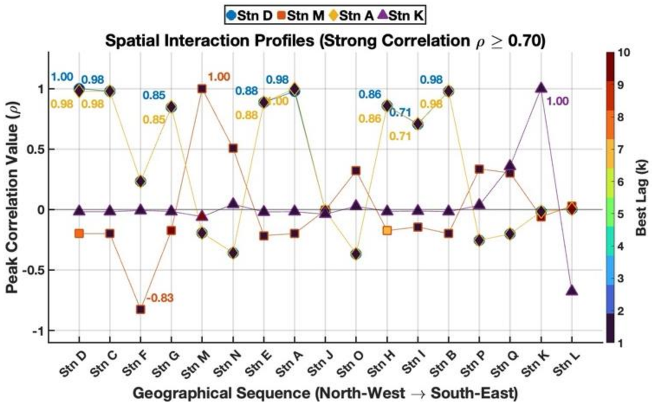

Fig. 4. All-month peak correlation and best-lag (k) profiles across the station sequence for the selected reference stations. Stations are shown in a fixed index order for visualization; the order is not used in the correlation computation.

C. *Forecasting Performance Results*

This section evaluates the predictive performance of LSTM, PatchTST, and TCN models using a univariate baseline from Station E.

TABLE I. NRMSE (%) AT EPOCH 100 ACROSS CONTEXT WINDOWS k AND NET CHANGE FROM $k = 1$ TO $k = 20$.

Model	k = 1 (%)	k = 3 (%)	k = 10 (%)	k = 20 (%)	Δ nRMSE (k=20 – k=1) (%)
LSTM	11.75	11.74	11.62	11.54	-0.21 (↓)
PatchTST	11.73	11.71	11.63	11.55	-0.18 (↓)
TCN	11.75	11.71	11.61	11.56	-0.19 (↓)

Performance is measured performance using nRMSE for each context window length k under the same split and preprocessing. Table I reports final (epoch 100) nRMSE values and shows a consistent decrease in error as k increases, indicating that longer input history provides additional predictive information. The net gain from expanding the window from k=1 to k=20 is largest for LSTM and smaller for PatchTST and TCN, while results converge at k=20 within a narrow band (11.54–11.56%). Fig. 5 visualizes the same trend and highlights reduced architecture sensitivity once the context window is sufficiently long.

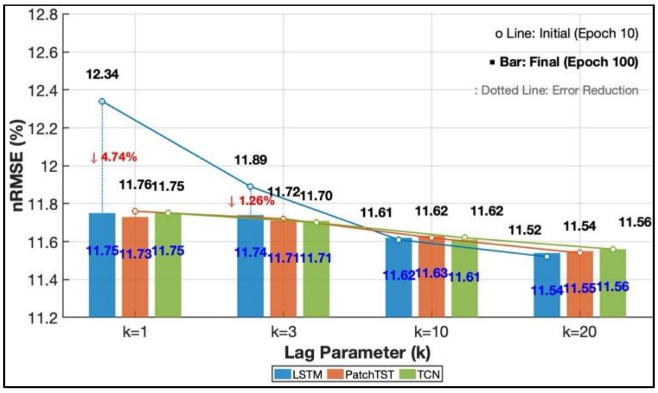

Fig. 5. nRMSE of LSTM, PatchTST, and TCN for context window lengths k ∈ {1, 3, 10, 20}.

1. Impact of Context Window Length:

The results show that context window length is a primary factor in improving forecasting accuracy. For all models, the nRMSE decreases as k increases from 1 to 20. For example, in the initial baseline (Epoch 10), the LSTM model starts with an nRMSE of 12.34% at k=1 and drops to 11.52% when the window is expanded to k=20. This aligns with broader time-series forecasting practice that the amount of relevant historical context available to the model can strongly influence predictive accuracy [32][33].

2. Model Comparison and Architecture Sensitivity:

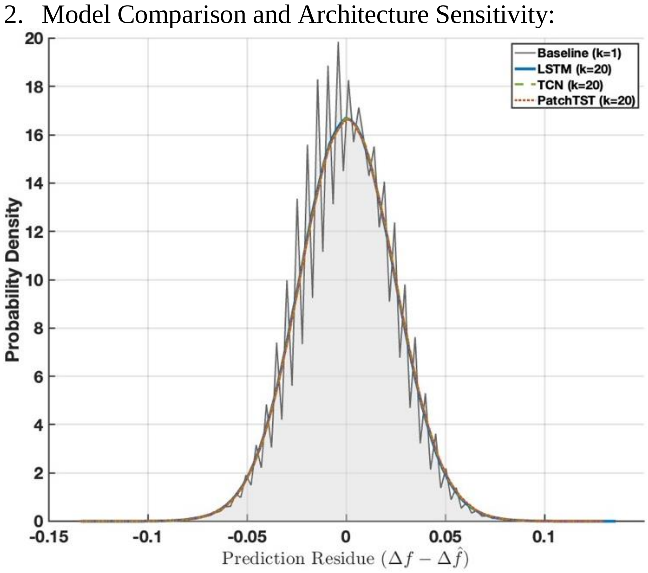

Fig. 6. Probability density functions of prediction residuals at station E comparing the k=1 baseline with k=20 optimized models.

At short windows, the convolutional and transformer-based models are more efficient than the recurrent baseline. At k=1, both TCN (11.75%) and PatchTST (11.76%) outperform LSTM (12.34%). As the window grows to k=20, all three architectures converge, with final nRMSE between 11.54% and 11.56%. This supports the view that capturing time-delayed effects depends strongly on whether the input window spans the relevant delays; once sufficient delayed information is included, remaining differences between architectures can be small in practice [32], [34].

Residual distributions provide an additional view of the window-length effect. Fig. 6 compares prediction residuals at Station E (≈24 million samples, 100% availability). Moving from k=1 (0.1 s history) to k=20 (2.0 s history) narrows the residual distribution and reduces error variance, producing a distribution more concentrated around zero. This indicates higher precision when the input window contains more delay-relevant history.

At k=20, the residual distributions of LSTM, TCN, and PatchTST largely overlap. This indicates that, under the current one-step setup and preprocessing, the three architectures extract similar predictive information once sufficient history is provided. The residual peak near zero also suggests limited systematic bias (no consistent over- or under-prediction). Together, these observations support a practical predictability limit under this experimental setup: additional architectural changes or longer history yield diminishing changes in the residual shape at this station.

3. Gains from Extended Training:

Comparing epoch 10 to epoch 100 shows that longer training mainly benefits short windows. The largest improvement occurs for LSTM at k=1, where nRMSE drops by 4.74% (12.34% to 11.75%), with a smaller gain of 1.26% at k=3. For k=10 and k=20, additional training brings little further improvement, suggesting that performance saturates early once the window already contains sufficient history. This diminishing-return behavior is consistent with empirical observations in time-series training [33], [34].

D. Discussion: Delay Propagation and Predictability Ceiling

The key novelty is the empirical link between measurement-based lagged coupling and the context requirements of deep forecasting. Best-lag analysis shows that some station pairs require larger shifts (up to ≈8–10 samples) to achieve maximum alignment. This helps explain why very short context (k=1) yields the highest errors: the model input cannot span delayed interactions implied by the coupling structure. When the window is expanded to k=20, it covers the observed lag range and the three architectures converge within about 0.3% nRMSE. This suggests that lag/window design is a physically grounded equalizer: once the input contains the relevant delay information, architecture choice becomes secondary to the information content of the time series [34].

The weekday–weekend coherence gap is consistent with weekly seasonality in electricity demand, where consumption patterns differ between workdays and weekends [35]. Combined with the best-lag maps, this supports the interpretation that window design is physically motivated: a sufficiently long input window can internalize delay effects and reduce sensitivity to architecture choice. Fig. 7 further illustrates that some residual spikes occur at the same times across LSTM, TCN, and PatchTST, indicating shared hard-to-predict events rather than model-specific failures.

Aligned prediction failures across models suggest that remaining errors are driven by components not captured by the available history (e.g., stochastic disturbances or unobserved exogenous actions), rather than by a specific algorithm. Since k=20 already spans the dominant delay range observed in the coupling analysis, synchronized residual spikes likely correspond to events with weak precursors in the measured signal. Across runs, all three architectures consistently approach an nRMSE around 11.5%, which we interpret as an information-saturation point for this station under the current one-step setup.

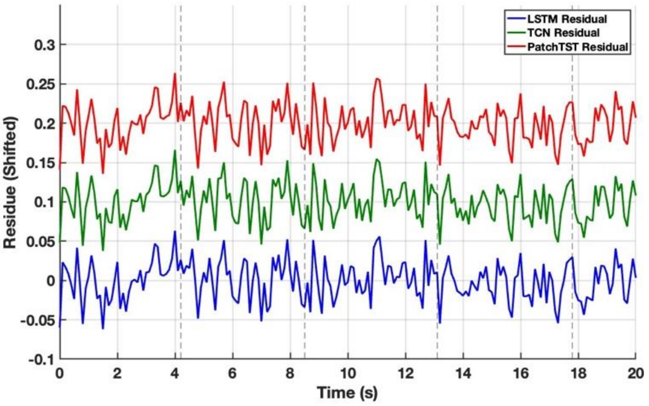

Fig. 7. Synchronized temporal residual analysis at station E comparing LSTM, TCN, and PatchTST architectures at k=20.

V. Conclusion

This study finds that frequency synchronization patterns are stable across regimes, even though average correlation is higher on weekends (0.258) than on weekdays (0.174). The station groups that remain consistently high- vs low-synchronization are largely unchanged, suggesting that relative coupling structure is shaped by the underlying interconnection. Best-lag maps reveal effective delays up to about 8–10 samples for some station pairs. In the univariate benchmark, using a forecasting window that spans these delays (k=20) reduces error and leads LSTM, TCN, and PatchTST to converge near 11.5% nRMSE. Overall, the results support window/lag design as a key factor for short-horizon forecasting and as a practical bridge between measurement-based coupling and model performance.

Acknowledgment

The authors acknowledge the financial support of the Stipendium Hungaricum Scholarship. Bálint Hartmann acknowledges the support of the Bolyai János Research Scholarship of the Hungarian Academy of Sciences (BO/131/23), and the National Research Excellence Programme of the National Research, Development and Innovation Office (ADVANCED 150405).

References

[1] B. Appasani and D. K. Mohanta, "A review on synchrophasor communication system: communication technologies, standards and applications," *Protection and Control of Modern Power Systems*, vol. 3, art. no. 37, 2018, doi: https://doi.org/10.1186/s41601-018-0110-4

[2] B. Appasani, A. V. Jha, S. K. Mishra, and A. N. Ghazali, "Communication infrastructure for situational awareness enhancement in WAMS with optimal PMU placement," *Protection and Control of Modern Power Systems*, vol. 6, art. no. 9, 2021, doi: https://doi.org/10.1186/s41601-021-00189-9

[3] D. S. Kumar, N. S. N. Manjula, and K. P. Vittal, "Micro-synchrophasor based special protection scheme for distribution system automation in a smart city," *Protection and Control of Modern Power Systems*, vol. 5, art. no. 7, 2020, doi: https://doi.org/10.1186/s41601-020-0153-1

[4] A. G. Phadke and T. Bi, "Phasor measurement units, WAMS, and their applications in protection and control of power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 6, pp. 619–629, 2018, doi: https://doi.org/10.1007/s40565-018-0423-3 (Springer)

[5] A. Sundararajan, T. Khan, A. Moghadasi, and A. I. Sarwat, "Survey on synchrophasor data quality and cybersecurity challenges, and evaluation of their interdependencies," *Journal of Modern Power Systems and Clean Energy*, vol. 7, pp. 449–467, 2019, doi: https://doi.org/10.1007/s40565-018-0473-6 (Springer)

[6] C. Huang, F. F. Li, D. Zhou, J. Guo, Z. Pan, Y. Liu, and Y. Liu, "Data quality issues for synchrophasor applications Part I: a review," *Journal of Modern Power Systems and Clean Energy*, vol. 4, pp. 342–352, 2016, doi: https://doi.org/10.1007/s40565-016-0217-4 (Springer)

[7] X. Wen, M. Anvari, L. Rydin Gorjão, G. C. Yalcin, V. Hagenmeyer, and B. Schäfer, "Nonstandard power grid frequency statistics across continents," *Scientific Reports*, 2025, doi: https://doi.org/10.1038/s41598-025-25334-4

[8] B. Hartmann *et al*., "Dynamical heterogeneity and universality of power-grids," *Sustainable Energy, Grids and Networks*, 2024, doi: https://doi.org/10.1016/j.segan.2024.101491

[9] Y. Xu, F. Wen, G. Ledwich, and Y. Xue, "Electromechanical wave in power systems: theory and applications," *Journal of Modern Power Systems and Clean Energy*, vol. 2, pp. 163–172, 2014, doi: https://doi.org/10.1007/s40565-014-0054-2 (Springer)

[10] C. Setianingsih and B. Hartmann, "Lag-Enhanced LSTM for Power Grid Frequency Prediction," in *Proc. Int. Youth Conf. on Energy (IYCE)*, 2025, doi: https://doi.org/10.1109/IYCE66046.2025.11155029

[11] T. Limouni, R. Yaagoubi, K. Bouziane, K. Guissi, and E. H. Baali, "Accurate one step and multistep forecasting of very short-term PV power using LSTM-TCN model," *Renewable Energy*, 2023, doi: https://doi.org/10.1016/j.renene.2023.01.118

[12] M. Gong, Y. Wang, J. Huang, H. Cui, S. Jing, and F. Zhang, "A novel wind power prediction model based on PatchTST and temporal convolutional network," *Environmental Progress & Sustainable Energy*, 2025, doi: https://doi.org/10.1002/ep.14584

[13] A. Sathe and W. R. Yang, "Grid Frequency Forecasting in University Campuses using Convolutional LSTM," *arXiv*, 2023, doi: https://doi.org/10.48550/arXiv.2310.16071

[14] C. Huang, F. F. Li, L. Zhan, Y. Xu, Q. Hu, D. Zhou, and Y. Liu, "Data quality issues for synchrophasor applications Part II: problem formulation and potential solutions," *Journal of Modern Power Systems and Clean Energy*, vol. 4, pp. 353–361, 2016, doi: https://doi.org/10.1007/s40565-016-0213-8 (Springer)

[15] S. Biswas, T. Yin, S. A. R. Naqvi, J. D. Follum, A. Varghese, T. Ahmad, and P. V. Etingov, "An Open Access Repository of Synchrophasor Data Quality Examples: Curation and Example Applications," *IEEE Access*, 2025, doi: https://doi.org/10.1109/ACCESS.2025.3560602

[16] Y. Zhang, T. Huang, and E. F. Bompard, "Big data analytics in smart grids: a review," *Energy Informatics*, vol. 1, art. no. 8, 2018, doi: https://doi.org/10.1186/s42162-018-0007-5

[17] L. Belcastro *et al*., "Programming big data analysis: principles and solutions," *Journal of Big Data*, 2022, doi: https://doi.org/10.1186/s40537-021-00555-2

[18] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *arXiv*, 2018, doi: https://doi.org/10.48550/arXiv.1803.01271

[19] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A Time Series is Worth 64 Words: Long-term Forecasting with Transformers," *arXiv*, 2022, doi: https://doi.org/10.48550/arXiv.2211.14730

[20] M. Sun, Z. Niu, X. Zhu, and Z. Huang, "High-Precision Time Delay Estimation Algorithm Based on Generalized Quadratic Cross-Correlation," *Mathematics*, vol. 13, no. 15, art. 2397, 2025, doi: https://doi.org/10.3390/math13152397

[21] Y. Mohammadi, B. Polajžer, R. C. Leborgne, and D. Khodadad, "Quantifying power system frequency quality and extracting typical patterns within short time scales below one hour," *Sustainable Energy, Grids and Networks*, vol. 38, art. 101359, 2024, doi: https://doi.org/10.1016/j.segan.2024.101359

[22] F. Møller Andersen, P. A. Gunkel, C. Bjerregaard, and H. K. Jacobsen, "Changes in hourly electricity consumption profiles and price elasticities in Denmark 2019–2022," *Energy*, vol. 308, art. 132798, 2024, doi: https://doi.org/10.1016/j.energy.2024.132798

[23] A. Jornaz and V. A. Samaranayake, "A Multi-Step Approach to Modeling the 24-hour Daily Profiles of Electricity Load using Daily

Splines," *Energies*, vol. 12, no. 21, art. 4169, 2019, doi: https://doi.org/10.3390/en12214169

[24] X. Yang *et al*., "Research on information leakage in time series prediction based on empirical mode decomposition," *Scientific Reports*, 2024, doi: https://doi.org/10.1038/s41598-024-80018-9

[25] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: https://doi.org/10.1162/neco.1997.9.8.1735

[26] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv*, 2014, doi: https://doi.org/10.48550/arXiv.1412.6980

[27] L. Prechelt, "Early Stopping – But When?," in *Neural Networks: Tricks of the Trade*, G. B. Orr and K.-R. Müller, Eds. Berlin, Heidelberg: Springer, pp. 55–69, 2002, doi: https://doi.org/10.1007/3-540-49430-8_3

[28] F. Hao, "Impact of electric vehicle charging demand on clean energy regional power grid control," *Energy Informatics*, vol. 8, art. no. 83, 2025, doi: https://doi.org/10.1186/s42162-025-00538-0 (Springer)

[29] X. Ding, J. Xu, Y. Sun, *et al*., "A demand side controller of electrolytic aluminum industrial microgrids considering wind power fluctuations," *Protection and Control of Modern Power Systems*, vol. 7, art. no. 49, 2022, doi: https://doi.org/10.1186/s41601-022-00270-x (Springer)

[30] F. Bai, Y. Liu, Y. Liu, *et al*., "A measurement-based approach for power system instability early warning," *Protection and Control of Modern Power Systems*, vol. 1, art. no. 4, 2016, doi: https://doi.org/10.1186/s41601-016-0014-0 (Springer)

[31] M. U. Usman and M. O. Faruque, "Applications of synchrophasor technologies in power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 7, pp. 211–226, 2019, doi: https://doi.org/10.1007/s40565-018-0455-8 (Springer)

[32] M. Alruqimi and L. Di Persio, "Multistep Brent oil price forecasting with a multi-aspect meta-heuristic optimization and ensemble deep learning model," *Energy Informatics*, vol. 7, art. no. 130, 2024, doi: https://doi.org/10.1186/s42162-024-00421-4 (Springer)

[33] H. Dhungana, "A machine learning approach for wind turbine power forecasting for maintenance planning," *Energy Informatics*, vol. 8, art. no. 2, 2025, doi: https://doi.org/10.1186/s42162-024-00459-4 (Springer)

[34] J. Wibbeke, D. Alves, and S. Rohjans, "Estimating time-delayed variables using transformer-based soft sensors," *Energy Informatics*, vol. 6, art. no. 16, 2023, doi: https://doi.org/10.1186/s42162-023-00274-3 (Springer)

[35] Groß, A. Lenders, F. Schwenker, and D. A. Braun, "Comparison of short-term electrical load forecasting methods for different building types," *Energy Informatics*, vol. 4, art. no. 13, 2021, doi: https://doi.org/10.1186/s42162-021-00172-6 (SpringerLink)