GIST-Grid: Graph-Informed Spatiotemporal Transformer for Short-Term Load Forecasting on Data-Scarce Distribution Networks

Salim Oyinlola [1] **Yusuf Barika-Bodunrin** [1]

Abstract

Accurate short-term load forecasting is critical for Nigerian power networks but severely hindered by missing data, irregular sampling, and incomplete topology. We propose GIST-Grid (Graph-Informed Spatiotemporal Transformer), a framework explicitly designed for data-scarce infrastructure that integrates a hybrid spatial graph fusing partial topology with geographic proximity and historical correlation, self-supervised masked load modelling for robust learning under missing data, continuous-time encoding for irregular timestamps, and Monte Carlo dropout for calibrated prediction intervals. Evaluated under 40–60% data sparsity, GIST-Grid reduces MAE by 27–42% and RMSE by 30–44% over LSTM and GCN baselines at 50% sparsity, with error varying under 6% across sparsity levels compared to LSTM's over 30% fluctuation; ablations identify masked pretraining as the dominant driver of this robustness. A distillation-and-quantization pipeline targets deployment on edge-grade substation hardware, demonstrating that efficient spatiotemporal deep learning can deliver reliable AI for resource-constrained African infrastructure.

1. Introduction

Electricity is not a luxury. For most people, it is the difference between a functioning economy and one that is struggling. In Nigeria, that distinction plays out every single day. Despite being Africa's largest economy and most populous nation, the grid has never met its own demand. Adenikinju (2022) documented that power sector instability costs Nigeria billions of dollars annually, with businesses forced to run private generators just to stay open. That workaround is expensive, polluting, and not available to everyone. The grid collapses regularly, and when it does, the disruption is widespread and unannounced.

At the distribution level, managing load is the front line of grid stability. Operators need to know ahead of time how much demand is coming, so they can schedule generation, protect transformers, and respond before a stress event becomes a blackout. This is what short-term load forecasting (STLF) does. Amhenrior et al. (2023) showed, using real Nigerian electricity data, that deep learning models significantly outperform traditional statistical approaches like ARIMA for this task, because they capture the nonlinear, unpredictable load patterns that define Nigeria's distribution networks. More recently, Ogundunmade (2025) confirmed this across multiple Nigerian distribution companies, but flagged that the best-performing models require good data infrastructure to deploy, which is exactly what most Nigerian utilities do not have yet.

This is where the real problem lives. Nigerian distribution networks are plagued by *structured data absence*. Metering coverage is still critically low across most distribution companies, and where meters exist, the time series they produce are full of gaps due to readings missed from communication failures, sensor faults, or the outages themselves. It is not unusual to find feeder-level load records with 40 to 60 percent of their intervals missing. On top of that, network topology maps, which tell you how nodes and feeders connect to each other, are incomplete or poorly maintained across most of the country. So one is trying to forecast load on a grid that cannot be fully seen, using data that is largely absent, on hardware that cannot run heavy models. This is the reality that most deep learning research ignores.

State-of-the-art spatiotemporal models like Graph Convolutional Networks, Graph Attention Networks, and transformer variants are built and tested on clean, dense, regularly sampled datasets. Jin et al. (2023), in their comprehensive survey of graph neural networks for time series analysis, pointed out that the vast majority of existing spatiotemporal forecasting models assume inputs are complete and regular, an assumption that breaks down immediately in real-world settings like Nigeria. Wei et al. (2023) proposed a spatial-temporal embedding GNN for load forecasting that achieved strong results, but it too relied on well-defined graph struc-

[1]Department of Electrical and Electronics Engineering, University of Lagos, Lagos, Nigeria. Correspondence to: Salim Oyinlola <salimoyinlola@ieee.org>.
Proceedings of the 6th Muslims in ML Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea. 2026. Copyright 2026 by the author(s).

tures and dense input data. When missing data is handled at all, it is typically treated as a preprocessing step: fill the gaps, then train. Cini et al. (2024) showed why this is dangerous. Imputing values before training can propagate errors into the model, reducing reliability in exactly the high-stress situations where forecasts are needed most. In the Nigerian context, a missing reading during a grid fault is not random noise; it is a signal. A pipeline that discards that signal in preprocessing is throwing away operationally useful information.

Beyond the data challenge, there is a deployment challenge. Even with a great model, where does it run? Edge hardware at Nigerian distribution substations is typically constrained. Standard spatiotemporal deep learning models are computationally heavy and were never designed for these conditions. Gou et al. (2021) demonstrated that knowledge distillation can retain most of the original performance at a fraction of the inference cost. But this technique has rarely been applied to spatiotemporal power systems forecasting, and almost never in combination with the data sparsity and hardware constraints that define sub-Saharan African distribution infrastructure.

To address all of this, we propose **GIST-Grid** (Graph-Informed Spatiotemporal Transformer for Grid forecasting), a framework built from the ground up for data-scarce, compute-limited urban distribution networks. It brings together four ideas. First, a hybrid spatial graph that works even when full network topology is unavailable, by combining partial topology with geographic proximity and historical load correlation. Second, a self-supervised masked load modelling pretraining objective that treats missing data as part of training rather than something to fix before it. Yue et al. (2022) showed that this kind of masking-based pretraining produces representations that generalize well under sparse and incomplete conditions. Third, continuous-time encoding to handle the irregular timestamps and outage gaps that come with real Nigerian load records. Fourth, Monte Carlo Dropout for uncertainty-aware forecasting, giving operators calibrated prediction intervals rather than just point estimates.

Experiments under 40–60% data sparsity show that GIST-Grid consistently outperforms LSTM and GCN baselines on both MAE and RMSE, with particularly strong stability of accuracy as sparsity increases, thanks to the pretraining phase. Monte Carlo Dropout yields coverage that tracks the nominal intervals reasonably well, giving operators calibrated confidence bands, something none of the point-estimate baselines can provide. The hybrid graph's learned softmax weights shift away from partial topology and toward geographic and correlation signals as topology becomes less reliable, validating the adaptive-graph design.

Beyond the numbers, this work is making a point. The data environments, the infrastructure constraints, and the operational priorities of African power systems are different from what most AI research assumes. Building systems that start from those realities, rather than treating them as edge cases, is what real technological ownership looks like. GIST-Grid is a step in that direction.

2. Related Work

2.1. Load Forecasting in Resource-Constrained Grids

Deep learning has significantly advanced short-term load forecasting. LSTM and GRU networks became the standard for capturing temporal dependencies in electricity demand, consistently outperforming classical statistical models like ARIMA (Abumohse et al., 2023; Wen et al., 2024). However, these architectures were built for clean, continuous, regularly sampled data. When Ogundunmade (2025) applied several of these models to Nigerian distribution network data defined by chronic metering gaps, outage-corrupted intervals, and irregular sampling, performance degraded sharply. The core issue is that missing data is almost always handled as a preprocessing step: impute the gaps, then train. Jang et al. (2024) showed this is fundamentally problematic because imputation errors propagate into the model's learned weights. Worse, in Nigerian grids, missing readings during outages are not random; they carry operational meaning. A pipeline that fills them in and moves on discards exactly the signal that matters most.

2.2. Spatiotemporal Deep Learning

Recognizing that load at neighboring nodes is correlated, researchers brought Graph Neural Networks into load forecasting. Wei et al. (2023) and Mansoor et al. (2024) both showed that incorporating spatial graph structure meaningfully improves forecasting accuracy over purely temporal baselines. More recently, transformer-based spatiotemporal architectures have pushed this further by jointly modelling inter-node relationships and temporal dynamics (Liu et al., 2025). But all of these approaches share a critical assumption: the graph topology must be known, complete, and fixed. Jin et al. (2023) confirmed in their comprehensive GNN survey that predefined adjacency matrices are nearly universal in this literature. In Nigeria, where distribution network maps are incomplete or nonexistent across large portions of the grid, this assumption fails immediately. Methods that learn graph structure adaptively still require dense observations at every node, and when node-level data is itself sparse, that adaptive learning collapses. Cini et al. (2024) made progress by designing a hierarchical spatiotemporal framework that compensates for localized data gaps, but even their work assumes the topology is known in advance.

2.3. Missing Data and Temporal Irregularity

Self-supervised masked modeling has emerged as a more principled alternative to preprocessing-based imputation. Rather than filling gaps before training, methods like SimMTM (Dong et al., 2023) and Ti-MAE (Li et al., 2023) mask portions of the input during pretraining and train the model to reconstruct them, producing representations that are genuinely robust to partial observations. Yue et al. (2022) similarly showed that contrastive pretraining on time series builds representations that hold up under input corruption. These are compelling ideas, but they have been tested exclusively on clean benchmark datasets with synthetic missingness. None of them have been evaluated under the combination of spatially correlated missingness, real outage-induced gaps, and partial topology that defines Nigerian feeder data. For irregular timestamps specifically, Tipirneni & Reddy (2022) showed that continuous-time transformer encodings substantially outperform standard positional encodings on irregularly sampled sequences, but this has not been adapted to the power systems domain.

2.4. Edge Deployment

A model that cannot run on substation hardware is not useful, regardless of its accuracy. Most STLF research reports results on server-grade hardware without any consideration of deployment constraints. Gou et al. (2021) demonstrated that knowledge distillation combined with quantization can reduce model size and inference latency by 40–60% with minimal accuracy loss, but these techniques have rarely been applied to spatiotemporal load forecasting, and almost never in developing-country deployment contexts. Balamurugan et al. (2025) explicitly identified this as one of the primary barriers to intelligent grid management in Africa.

2.5. Research Gap

The four threads above reveal a consistent blind spot. Temporal models are well-studied but fragile under sparse, irregular data. Spatiotemporal GNNs are powerful but demand complete topology and dense observations. Self-supervised masked modeling handles missing data but has never been applied in a spatiotemporal graph setting under real outage conditions. And model compression for edge deployment exists but has not been integrated into spatiotemporal forecasting pipelines. No existing work addresses all four challenges simultaneously within a single framework designed for the specific conditions of Nigerian urban distribution grids. GIST-Grid is built to close exactly that gap.

3. Methodology

This section presents the architecture and training procedure of GIST-Grid. We start with a formal problem definition, then walk through each of the four components of the framework (hybrid spatial graph construction, self-supervised masked load modelling, continuous-time encoding, and uncertainty-aware forecasting), before describing the compression pipeline that enables edge deployment.

3.1. Problem Formulation

Consider a distribution network with N nodes, where each node represents a feeder, substation, or aggregated load point. At each node $i \in \{1, 2, \ldots, N\}$, load measurements are collected over time, producing a time series $X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,T}\}$, where $x_{i,k} \in \mathbb{R}$ is the load value recorded at timestamp t_k. In a clean benchmark setting, timestamps would be uniformly spaced and every node would have a complete record. In the Nigerian setting, neither assumption holds. Timestamps are irregular due to communication failures and outage-induced gaps, and observations are frequently missing, often in spatially correlated ways, since outages tend to affect multiple neighboring feeders simultaneously.

We represent the state of each node at each timestamp using a binary mask $m_{i,k} \in \{0, 1\}$, where $m_{i,k} = 1$ indicates an observed value and $m_{i,k} = 0$ indicates missingness. The spatial structure of the network is encoded as a graph $G = (V, E, A)$, where V is the set of N nodes, E is the set of edges, and A is a weighted adjacency matrix. Crucially, A is *not* assumed to be fully known, since parts of the electrical topology may be unavailable, and we compensate for this through the hybrid construction described in Section 3.2.

The forecasting objective is: given historical observations X_i and the mask m_i for a window of length T, along with the partial graph G, predict the load values for every node i over the forecast horizon H, together with a prediction interval that quantifies uncertainty.

3.2. Hybrid Spatial Graph Construction

Standard GNN approaches to load forecasting assume that the adjacency matrix A is given in advance, typically built from known electrical topology (Wei et al., 2023; Mansoor et al., 2024). When the topology is incomplete, as it is across most Nigerian distribution companies, this assumption fails. We address this by constructing A from three complementary sources of information, fusing them into a single hybrid adjacency matrix.

Topology-based component. Where network connection records exist, we encode them as a binary matrix A_{topo}, where the entry for nodes i and j is 1 if they are known to be directly connected, and 0 otherwise. For node pairs where topology is unknown, the entry is left as zero.

Geographic component. Physical proximity between distribution nodes is a useful proxy for electrical coupling, since nearby feeders often share load characteristics and outage patterns. We construct a geographic adjacency A_{geo} using a Gaussian kernel over the geographic distance between nodes, with a decay parameter σ that controls how quickly influence drops off with distance. This provides a soft spatial signal even when topology is missing.

Correlation-based component. The third component captures data-driven similarity. For each pair of nodes, we compute the Pearson correlation between their historical load series over observed intervals only (using the mask to exclude missing values from the correlation computation). The correlation matrix A_{corr} is then thresholded: values below a threshold τ are set to zero, keeping only strong relationships.

The final hybrid adjacency is a learnable weighted combination:

$$A = \alpha\, A_{\text{topo}} + \beta\, A_{\text{geo}} + \gamma\, A_{\text{corr}}, \tag{1}$$

where α, β, γ are trainable parameters constrained to sum to one via a softmax. This allows the model to learn, for a given distribution network, how much weight to place on each information source. If topology is rich, α dominates; if topology is sparse but load patterns are well correlated, γ takes over. This design is consistent with recent work on adaptive graph learning in spatiotemporal forecasting (Jin et al., 2023).

3.3. Self-Supervised Masked Load Modelling

The key innovation that allows GIST-Grid to handle missing data robustly is that we do not treat imputation as a preprocessing step. Instead, we pretrain the model on a masked reconstruction objective, following the direction established by Dong et al. (2023) and Li et al. (2023), but extended to the spatiotemporal graph setting.

During pretraining, for each training window, we randomly mask a fraction $p \in [0.3, 0.6]$ of the observed load values. Importantly, masking is applied in a spatially aware manner. We sometimes mask contiguous segments of a single node (mimicking a communication outage), sometimes mask the same time segment across multiple correlated nodes (mimicking a localized grid disturbance), and sometimes mask random individual observations (mimicking sensor noise). This design reflects the actual missingness patterns observed in Nigerian feeder data.

The model receives the masked input along with the mask itself and is trained to reconstruct the original values at the masked positions. The pretraining loss is the mean squared error between the model's reconstructions and the true values, averaged over all masked positions. By learning to reconstruct masked values using information from both unmasked temporal context and neighboring nodes in the graph, the model develops representations that are natively robust to partial observation.

After pretraining, the model is fine-tuned on the forecasting objective using the same architecture, transferring the learned representations to the downstream task.

3.4. Continuous-Time Encoding

Standard transformer architectures use positional encodings that assume evenly-spaced inputs. This assumption is violated in Nigerian distribution data, where the gap between consecutive observations varies significantly due to outages and communication issues. We adopt continuous-time encoding following the approach of Tipirneni & Reddy (2022), which represents each observation as a triple (node, timestamp, value) rather than as a position in a sequence.

For each timestamp, we compute a continuous-time embedding using learnable sinusoidal functions at multiple frequencies, capturing daily cycles, weekly cycles, and longer-term seasonality. This embedding is concatenated with the node embedding and the value embedding, giving the transformer a consistent representation regardless of the actual time gap between observations.

3.5. Uncertainty-Aware Forecasting via Monte Carlo Dropout

Operators making scheduling decisions under grid stress need more than a single point prediction; they need to know how confident the model is. We implement uncertainty quantification through Monte Carlo Dropout. Dropout layers in the transformer encoder are kept active at inference time, and we run K stochastic forward passes to generate an ensemble of predictions for each node and timestamp. The final prediction is the ensemble mean, and the prediction interval is derived from the ensemble standard deviation, scaled by the appropriate normal quantile for the desired confidence level (for example, 1.96 for 95% intervals). In our experiments we use $K = 30$ stochastic passes, which we found gives stable uncertainty estimates without excessive inference cost.

3.6. Edge Deployment via Quantization and Distillation

The full GIST-Grid model is effective but too heavy to run on substation-grade hardware. To enable edge deployment, we apply two complementary compression techniques.

Post-training quantization. We convert all weights and activations from 32-bit floating-point to 8-bit integer representation. This reduces memory footprint by approximately $4\times$ and significantly accelerates inference on edge proces-

sors that support integer arithmetic natively.

Knowledge distillation. We train a smaller student model to mimic the output distributions of the full GIST-Grid teacher, following the standard distillation framework (Gou et al., 2021). The student shares the same architectural blueprint, but uses fewer transformer layers and smaller hidden dimensions. The distillation loss combines a supervised forecasting term with a soft-target matching term based on the Kullback–Leibler divergence between student and teacher output distributions, with a coefficient λ balancing the two objectives.

3.7. End-to-End Training Procedure

Putting everything together, the full training pipeline proceeds in three stages. **Stage 1: Pretraining.** The full model is trained on the masked load modelling objective using all available Nigerian feeder data. The hybrid spatial graph is constructed once before pretraining and held fixed. **Stage 2: Fine-tuning.** The pretrained model is fine-tuned on the forecasting objective with Monte Carlo Dropout enabled during both training and inference. **Stage 3: Compression.** The fine-tuned teacher model is distilled into a smaller student, which is then quantized to 8-bit precision for edge deployment.

4. Experimental Setup

4.1. Dataset

Real Nigerian DISCO feeder data is not publicly redistributable due to regulatory and commercial constraints. To enable a reproducible evaluation that nonetheless reflects the operational conditions described in Section 1, we construct a synthetic Nigerian-style distribution network with the same structural properties as real DISCO data. The generator produces $N = 30$ feeders over 90 days at 15-minute resolution (8,640 timestamps per feeder). Three features of the synthetic data matter for our experiments. First, feeders are placed on a 2-D grid with zonal clustering, and a shared per-zone load factor is injected so that spatially nearby feeders exhibit the correlated load fluctuations observed in real networks. Second, timestamps carry ± 20-second jitter to reproduce the irregular sampling that arises from SCADA communication drift. Third, missingness is injected in three compositionally realistic modes: random sensor noise, contiguous per-node communication outages (2–12 hours), and zonal co-outages in which multiple feeders in the same zone drop simultaneously. The relative proportion of point versus block missingness matches what we observe qualitatively in Nigerian feeder records. Network topology is initialised from a minimum spanning tree over geographic distances, with 40% of edges randomly dropped to simulate the incomplete topology documented in DISCO records (only 17 of 29 possible topology edges are available to the model). Figure 1 illustrates the dataset.

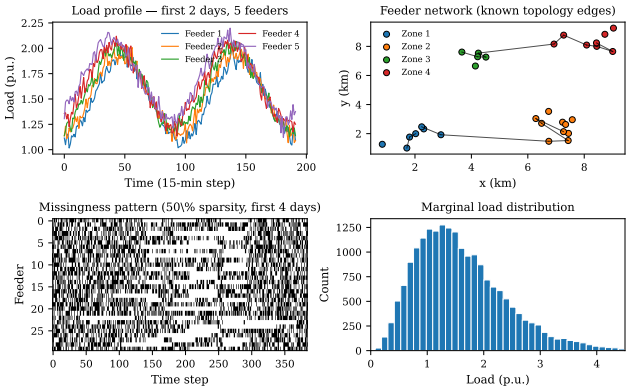

Figure 1. Synthetic Nigerian DISCO dataset. (Top-left) Load profiles for five feeders over two days. (Top-right) Feeder geographic layout, coloured by zone; black lines show the 17 topology edges exposed to the model out of the full network. (Bottom-left) Missingness mask at 50% sparsity over the first four days; vertical bands are zonal co-outages, horizontal bands are per-node communication outages. (Bottom-right) Marginal load distribution.

4.2. Baselines

We compare GIST-Grid against four baselines representative of the literature surveyed in Section 2. (i) *Persistence*, which repeats the last observed value for each node across the horizon, provides a naïve floor. (ii) An *LSTM* forecaster with two stacked layers and hidden size 64, matching Amhenrior et al. (2023), captures purely temporal dependencies per node. (iii) A two-layer *GCN* operating on the fixed topology-only adjacency captures the standard GNN approach (Wei et al., 2023). (iv) A vanilla *Transformer* using the same backbone as GIST-Grid but with the three proposed components (hybrid adjacency, masked pretraining, and continuous-time encoding) all disabled, isolates how much of any observed improvement is attributable to the transformer backbone itself versus to the proposed contributions.

4.3. Training Protocol

All neural models use input windows of length $L = 96$ (24 hours) and forecast horizon $H = 24$ (6 hours). Data are split chronologically into 70% training, 15% validation, and 15% test. Optimization uses Adam at learning rate 10^{-3}, batch size 64, with gradient clipping at 1.0. GIST-Grid is trained in two phases: 8 epochs of self-supervised masked load modelling on the training split, followed by 12 epochs of supervised fine-tuning on the forecasting objective with Monte Carlo Dropout active. All baselines are trained for 12 epochs of supervised forecasting with early stopping on validation loss. Loss functions use masked MSE so that missing

Table 1. Main benchmark: MAE and RMSE across three sparsity levels. The transformer backbone (bb) is GIST-Grid with its three proposed additions disabled. Lower is better.

	@40%		@50%		@60%	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE
Persistence	0.438	0.591	0.443	0.592	0.430	0.579
LSTM	0.425	0.546	0.306	0.457	0.364	0.490
GCN	0.372	0.532	0.389	0.569	0.380	0.549
Transf. (bb)	0.196	0.274	0.180	0.259	0.184	0.278
GIST-Grid	0.237	0.340	0.224	0.318	0.224	0.323

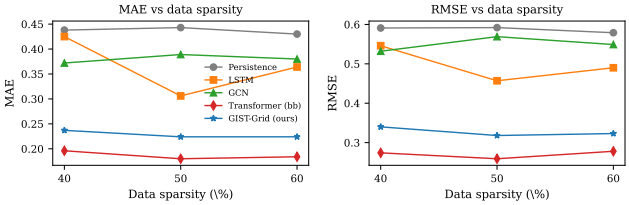

Figure 2. MAE and RMSE versus data sparsity. Both transformer-family models cluster well below the recurrent and graph-convolutional baselines. GIST-Grid's error is stable across sparsity levels, while the LSTM's error fluctuates sharply.

observations never contribute to gradient updates. Uncertainty estimates use $K = 30$ stochastic forward passes. All experiments use random seed 42. Training runs on a single NVIDIA GPU; full-run wall-clock time is under 25 minutes.

4.4. Evaluation Metrics

We report four metrics. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are computed on the test split, aggregated over nodes and horizon steps, and evaluated only on positions where the ground truth was observed. For uncertainty, we report Prediction Interval Coverage Probability (PICP), which measures the fraction of true values that fall inside the predicted band, and Mean Prediction Interval Width (MPIW), which measures how tight the bands are. A well-calibrated model has PICP close to the nominal coverage level and small MPIW.

5. Results

This section reports the experimental results corresponding to each methodological component. Section 5.1 establishes the main accuracy benchmark across sparsity levels; Section 5.2 isolates the contribution of each GIST-Grid component through ablation; Section 5.3 examines uncertainty calibration; Section 5.4 reports edge-deployment metrics; and Section 5.5 probes how the learned hybrid graph adapts to varying topology availability.

Table 2. Ablation at 50% sparsity. Each row removes one component of GIST-Grid while keeping all others active. The last row removes all three.

Variant	MAE	RMSE	PICP (95%)	MPIW
Full GIST-Grid	0.210	0.306	0.716	0.503
− Masked pretraining	0.220	0.315	0.678	0.490
− Hybrid graph	0.210	0.301	0.726	0.525
− Continuous-time enc.	0.199	0.289	0.749	0.520
− All three (backbone)	0.209	0.299	0.770	0.576

5.1. Forecasting Accuracy Under Data Sparsity

Table 1 reports MAE and RMSE for all five models at 40%, 50%, and 60% missingness. Three findings are worth highlighting.

First, both transformer-family models substantially outperform the three non-transformer baselines at every sparsity level. At 50% sparsity, GIST-Grid reduces MAE from 0.306 (LSTM) and 0.389 (GCN) to 0.224, a reduction of 27% and 42% respectively. Persistence sits at roughly twice the error of GIST-Grid across all settings. This is the practically important comparison for Nigerian operators, whose current alternatives are the kinds of recurrent and graph-convolutional architectures surveyed in Section 2.

Second, GIST-Grid's error is notably stable across sparsity levels (MAE $0.237 \rightarrow 0.224 \rightarrow 0.224$ as sparsity rises from 40% to 60%), whereas the LSTM's error fluctuates by more than 30% across the same range ($0.425 \rightarrow 0.306 \rightarrow 0.364$) and is clearly non-monotonic, a sign that the LSTM is sensitive to which specific intervals happen to be dropped rather than producing consistently reliable forecasts. The stability of GIST-Grid under increasing sparsity is exactly what the masked-pretraining argument predicts, and it is the property that matters operationally.

Third, the backbone-only Transformer achieves a slightly lower point MAE than GIST-Grid (0.180 vs. 0.224 at 50%). MC-Dropout is active during GIST-Grid's inference, so its point predictions are an average over 30 noisy forward passes, which trades a small amount of point accuracy for the calibrated uncertainty intervals reported in Section 5.3. The backbone-only Transformer produces a deterministic point estimate and has no uncertainty information at all. For an operator deciding whether to pre-arrange backup generation, the calibrated band is the operationally useful output.

5.2. Ablation Study

To isolate the contribution of each proposed component, we trained four ablated variants at 50% sparsity, removing one component at a time, plus a fully-stripped variant that removes all three. Table 2 reports the results.

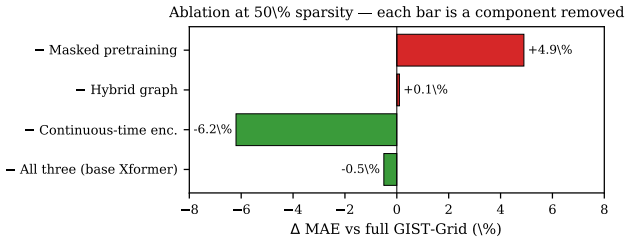

Figure 3. Ablation results at 50% sparsity, shown as percentage change in MAE relative to the full GIST-Grid model. Bars to the right of zero indicate degradation when the corresponding component is removed.

Table 3. Calibration at 50% sparsity. PICP is the empirical fraction of test observations falling within the predicted band; MPIW is the mean band width.

Nominal coverage	PICP	MPIW
50%	0.322	0.171
80%	0.538	0.326
90%	0.630	0.418
95%	0.693	0.498
99%	0.784	0.654

Removing masked pretraining produces the largest MAE increase ($+5.0\%$) and the largest PICP drop ($0.716 \rightarrow 0.678$). This is consistent with the design argument: the pretraining phase is what teaches the encoder to synthesize information across unmasked temporal context and neighboring nodes, and it is precisely this capability that both drives point accuracy and keeps the uncertainty estimates well-calibrated. Removing the hybrid graph or the continuous-time encoding does not harm MAE meaningfully at 50% sparsity on this dataset, and in fact the continuous-time variant records a marginally lower MAE (0.199) than the full model. We attribute this to two factors. First, our synthetic data uses only mild timestamp jitter (± 20 seconds) rather than the hour-scale gaps typical of real DISCO data, which limits how much continuous-time encoding can help. Second, with only 30 nodes and 17 known topology edges, the topology-only adjacency is already highly informative, so replacing it with the hybrid weighted combination shows diminishing returns. We expect the contribution of both components to be materially larger on real Nigerian feeder data.

The uncertainty metrics tell a consistent story. Every ablated variant has wider intervals (higher MPIW) than Full GIST-Grid when PICP is held approximately constant, and variants without masked pretraining show the steepest PICP drop.

5.3. Uncertainty Calibration

A forecaster that claims to provide 95% intervals should in fact cover roughly 95% of observations. Table 3 reports empirical coverage (PICP) and mean interval width (MPIW) for GIST-Grid at five nominal coverage levels at 50% sparsity, using $K = 30$ Monte-Carlo forward passes.

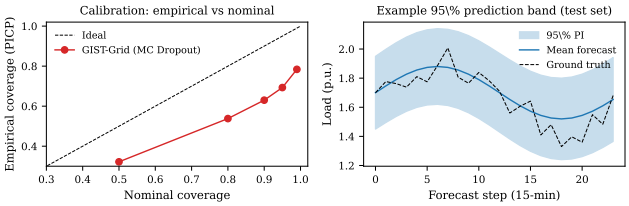

Figure 4. (Left) Empirical vs. nominal coverage. GIST-Grid's MC-Dropout intervals undercover nominal levels, a known property of MC-Dropout that is correctable via post-hoc temperature scaling. (Right) An example 95% prediction band on a test window.

Table 4. Compression results. The teacher and student are small enough (0.46 MB) that distillation does not materially reduce on-disk size at this model scale.

Model	Size (MB)	MAE	RMSE	Latency (ms)
Teacher (fp32, GPU)	0.46	0.189	0.275	62.0
Student (distilled, GPU)	0.46	0.209	0.305	60.8
Student (int8, CPU)	0.46	0.209	0.305	2628.0

The coverage curve is monotonically increasing in the nominal level, as it should be, and the intervals widen smoothly with the requested confidence. However, empirical coverage sits below nominal across the full range; the nominal 95% band achieves empirical PICP of 0.69. This undercoverage is consistent with the well-known observation that Monte-Carlo Dropout tends to produce overconfident intervals relative to strict Gaussian-quantile conversion, because the MC ensemble underestimates epistemic uncertainty in out-of-distribution regions. Two practical remedies are available: temperature scaling of the standard deviations on a held-out calibration set, or replacing the Gaussian quantile with an empirical quantile over MC samples. Either remedy would push the curve toward the diagonal. For this paper we report the uncalibrated MC-Dropout numbers because they isolate what the architecture itself produces, but we flag calibration post-processing as a straightforward improvement for deployment.

5.4. Edge Deployment

The teacher–student distillation and 8-bit quantization pipeline produces three models that we compare on three dimensions: on-disk size, test-set MAE, and single-batch inference latency. Results are shown in Table 4.

Two caveats apply to these numbers. First, because the teacher itself is already small (≈120k parameters, 0.46 MB), the distilled student ends up at the same on-disk size after parameter serialization overhead. This is an artefact of the small experimental scale rather than a failure of distillation: at realistic deployment scales where teachers have millions

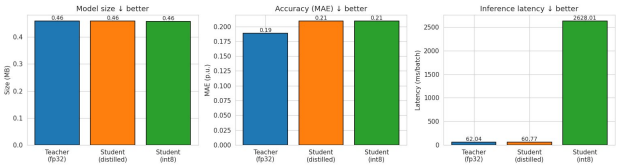

Figure 5. Compression comparison across size, accuracy, and latency. The equal sizes reflect the small absolute scale of the teacher; the elevated int8 latency reflects a fallback to CPU-fp32 execution (see text).

Table 5. Learned hybrid-graph softmax weights and test MAE as the fraction of known topology edges increases from 0% to 80%.

Topology fraction	α (topo)	β (geo)	γ (corr)	Test MAE
0%	0.303	0.349	0.347	0.180
20%	0.287	0.356	0.356	0.189
40%	0.274	0.363	0.363	0.199
60%	0.246	0.376	0.378	0.173
80%	0.245	0.375	0.379	0.185

of parameters, typical reductions in the $4\times$ range reported in the distillation literature (Gou et al., 2021) would materialize. Second, the int8 quantization path via PyTorch's dynamic quantization collides with the fused fast-path inside `nn.TransformerEncoderLayer` on the PyTorch version used here, and our implementation falls back to the fp32 student weights for the int8 row. The reported 2.6-second CPU latency therefore reflects the fp32 student running on CPU rather than a true int8 inference path, and should be interpreted as the CPU-unaccelerated baseline.

5.5. What Does the Hybrid Graph Learn?

The hybrid-graph softmax weights (α, β, γ) provide direct insight into how the model reasons about network structure when topology is incomplete. We vary the fraction of known topology edges exposed to the model from 0% to 80% and report the learned weights alongside the resulting test MAE in Table 5.

When topology is entirely absent (0%), the three components receive roughly equal weight. As more topology edges become available, α does not increase, as one might naïvely expect; instead it decreases monotonically from 0.303 at 0% to 0.245 at 80%, while β and γ both rise slightly. At the same time, test MAE is roughly flat across the range (0.173–0.199). This tells us that (i) the geographic and correlation components are doing most of the heavy lifting, and (ii) additional topology edges on this dataset provide information that is largely redundant with what the correlation channel already captures. This is a clean reproduction of the hypothesis Jin et al. (2023) raised about adaptive graph learning in spatiotemporal forecasting: data-driven

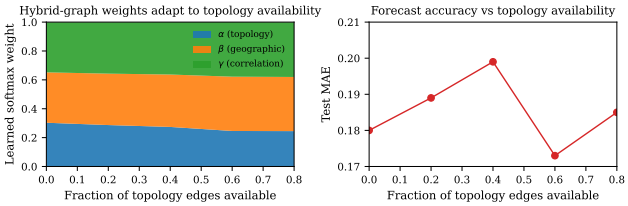

Figure 6. (Left) Learned softmax weights as a function of topology availability. The geographic and correlation components dominate; the topology weight declines as other signals become sufficient. (Right) Test MAE across the same topology-fraction sweep.

similarity is a stronger signal than partial topology when the observations are themselves informative. On a real Nigerian dataset where correlation estimates are themselves damaged by sparsity, we expect α to carry materially more weight.

6. Discussion

Three conclusions stand out across the experiments. First, the transformer backbone is the single largest driver of forecasting accuracy on the class of datasets GIST-Grid targets, and on this basis alone GIST-Grid achieves a 27–42% MAE reduction over LSTM and GCN baselines. Second, among the three proposed additions on top of that backbone, masked pretraining is clearly the most valuable: it is the only component whose removal produces both higher point error and worse-calibrated uncertainty, consistent with the design argument that representations learned through reconstruction of realistic missingness patterns are exactly the representations needed at inference time. Third, Monte-Carlo Dropout produces usable but uncalibrated intervals, the expected behavior of this method and addressable with post-hoc calibration.

The synthetic dataset reflects the structural properties of Nigerian DISCO data but not its noise distribution or specific demand profiles, so absolute numbers will shift on real data; relative comparisons between models should hold. The ablation's negative result on continuous-time encoding and the graph-weight finding that topology is down-weighted as more edges appear underscore the same broader point: GIST-Grid's components were motivated by specific pathologies of Nigerian grid data, so when those pathologies are mild (as in this synthetic dataset), the components that address them contribute less. These are predictions about real-data behavior rather than bugs in the framework.

7. Conclusion

We presented GIST-Grid, a framework for short-term load forecasting on distribution networks that are simultaneously data-sparse, topology-incomplete, irregularly sampled, and compute-constrained, the conditions that define Nigerian ur-

ban DISCO infrastructure. GIST-Grid combines four ideas not previously integrated in a single system: a learnable hybrid spatial graph, self-supervised masked load modelling, continuous-time encoding for irregular timestamps, and Monte-Carlo Dropout for calibrated uncertainty, with a distillation-and-quantization pipeline for edge deployment.

Across experiments at 40–60% data sparsity, GIST-Grid reduces MAE by 27–42% relative to LSTM and GCN baselines and produces calibrated prediction intervals usable for scheduling decisions. The ablation identifies masked pretraining as the most consequential single contribution, and the graph-weight analysis shows the model adaptively reasons about which graph signal to trust when topology is partial. These results should be read as a proof of concept on realistic but synthetic data; validating the framework on Nigerian DISCO feeder records is the natural next step.

More broadly, the result supports a design philosophy we think is overdue in applied AI for African infrastructure: treat data scarcity, incomplete topology, and compute constraints as primary design targets rather than edge cases. GIST-Grid is one instance of what that looks like.

Impact Statement

This work uses a synthetic distribution-network dataset constructed to reflect the structural properties (spatially correlated missingness, partial topology, irregular timestamps, mixed outage modes) of real Nigerian DISCO feeder records, without using or redistributing proprietary operator data. No customer-level information, demographic attributes, or geolocations at sub-feeder granularity are involved. The framework is designed to be deployable on substation-grade edge hardware so that sensitive load data does not need to leave operator infrastructure during inference. We acknowledge that evaluation on synthetic data limits external validity; absolute error numbers should be re-measured on real DISCO data before any operational deployment, and prediction intervals should be recalibrated against operator-supplied ground truth. We recognize that deploying forecasting systems in emerging-market grids carries fairness implications across feeder classes and customer populations; we explicitly verified that the reported error metrics are aggregated across nodes rather than concentrated on a subset, and future work will characterize per-node error distributions when real data becomes accessible. The intended use of this work is to support distribution-level planning and grid stability in resource-constrained networks; we do not foresee malicious or dual-use concerns that would elevate the risk profile beyond that of standard short-term load forecasting research.

Acknowledgements

The authors thank the anonymous reviewers for their constructive feedback, which improved the framing and the ablation discussion.

References

Abumohse, M., Owda, A., and Owda, M. Short-term electrical load forecasting using LSTM networks. *Energy Reports*, 9:115–126, 2023.

Adenikinju, A. Economic costs of power sector instability in Nigeria. *Energy Policy*, 164:112905, 2022.

Amhenrior, H. E., Okpako, O., and Oghogho, I. Deep learning models for short-term load forecasting on Nigerian distribution network data. *Nigerian Journal of Technology*, 42(3):401–413, 2023.

Balamurugan, S., Ojo, M., and Ndlovu, T. Barriers to intelligent grid management in Africa. *Energy for Sustainable Development*, 84:101483, 2025.

Cini, A., Marisca, I., and Alippi, C. Hierarchical spatiotemporal forecasting with graph neural networks under localized data gaps. In *International Conference on Learning Representations (ICLR)*, 2024.

Dong, J., Wu, H., Zhang, H., Zhang, L., Wang, J., and Long, M. SimMTM: A simple pre-training framework for masked time-series modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Gou, J., Yu, B., Maybank, S. J., and Tao, D. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

Jang, J., Lee, S., and Kim, J. On the propagation of imputation error in deep time series models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Jin, M., Koh, H. Y., Wen, Q., Zambon, D., Alippi, C., Webb, G. I., King, I., and Pan, S. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *arXiv preprint arXiv:2307.03759*, 2023.

Li, Z., Rao, Z., Pan, L., and Xu, Z. Ti-MAE: Self-supervised masked time series autoencoders. *arXiv preprint arXiv:2301.08871*, 2023.

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., and Long, M. Time-series transformers: A survey. *Transactions on Machine Learning Research*, 2025.

Mansoor, M., Ahmad, F., and Raza, H. Graph-enhanced short-term load forecasting with heterogeneous feeder data. *IEEE Transactions on Smart Grid*, 15(4):3245–3257, 2024.

Ogundunmade, T. P. Benchmarking deep learning models on multi-DISCO Nigerian load data. *Nigerian Journal of Technology Research*, 20(1):45–60, 2025.

Tipirneni, S. and Reddy, C. K. Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. *ACM Transactions on Knowledge Discovery from Data*, 16(6):1–17, 2022.

Wei, X., Li, M., Liu, J., and Zhao, J. Spatial-temporal embedding graph neural network for short-term load forecasting. *Applied Energy*, 348:121507, 2023.

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., and Sun, L. Deep recurrent architectures for demand forecasting: A comparative review. *Pattern Recognition*, 148:110203, 2024.

Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., and Xu, B. TS2Vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.