

Explainable AI-based Multi-Branch Hybrid Model for Short-term Energy Demand Forecasting

¹Akpan, E. J. ²Sako, D. J. S. ³Nwiabu, N. D.

Department of Computer Science,
Rivers State University, Port-Harcourt, Nigeria

¹eseabasi.akpan@ust.edu.ng, ²dum.sako@ust.edu.ng, ³nwiabu.nuka@ust.edu.ng

<https://doi.org/10.56201/wjimt.v10.no10.2026.pg127.139>

Abstract

Energy demand forecasting occupies a fairly central place in modern power system operation, and yet two problems keep resurfacing in the literature devoted to it. Weather variables, more often than not, get folded into forecasting models only partially, and the deep learning architectures that now dominate the field remain, for the most part, opaque to the people who must actually rely on their output. This paper presents an explainable multi-branch hybrid Convolutional Neural Network and Long Short-Term Memory (CNN-LSTM) model for energy demand forecasting in a smart grid system. Historical demand, weather and calendar information, and a set of engineered lag and rolling statistical features were each routed through a separate convolutional-recurrent branch before the three resulting representations were concatenated and passed through fully connected layers to produce a twenty-four-hour-ahead forecast. Interpretability was addressed through SHAP (Shapley Additive Explanations), applied by way of a GradientExplainer, so that predictions could be attributed to individual branches and features across the forecast horizon rather than at a single fixed point alone. The proposed model was benchmarked against three baselines trained under matched conditions, a standalone LSTM, CNN, and a sequential CNN-LSTM, using hourly data drawn from Panama's national electricity grid. On aggregate test-set metrics, the multi-branch model (RMSE 86.69, MAE 60.687, MAPE 5.169%, R^2 0.780) performed almost identically to, though not decisively better than, the considerably simpler LSTM baseline (RMSE 86.33, MAE 60.465, MAPE 5.173%, R^2 0.782). SHAP attribution showed that historical demand accounted for roughly 63.6% of overall importance, weather and calendar variables for 24.3%, and engineered lag features for the remaining 12.2%, with this ranking shifting toward cyclical hour-of-day encodings as the horizon lengthened.

Keywords: Grid system, energy demand, explainable AI, CNN, LSTM, electricity, forecasting

1. Introduction

The energy landscape is shifting, and shifting quickly. Rapid urbanisation, the spread of internet-connected devices, and the growing presence of decentralised renewable generation are pushing power systems into territory that their original designers, working with one-way supply frameworks, never really anticipated. Smart grids, capable of monitoring and adjusting themselves in something close to real time, are gradually taking the place of these older arrangements, and within that shift, energy demand forecasting has become one of the quietly indispensable tasks holding the whole enterprise together. It balances supply against demand, mitigates operational

risk, and supports the dispatch decisions that keep a grid stable (Zhang, Liu & Wang, 2021). None of this is conceptually new. What has changed is mostly the scale and the pace at which it now has to happen. Electricity demand does not sit still. It moves across the day, across the seasons, and from one region to another, driven by weather, industrial activity, calendar effects, and a handful of socioeconomic factors that do not always behave predictably. Because of this, utilities need some principled way of projecting future consumption if generation, transmission, and distribution are to be planned with any real confidence. Historically this task leaned on statistical and econometric tools, regression models, exponential smoothing, and autoregressive integrated moving average methods among them, and these approaches did reasonably well on stationary, largely linear series (Box, Jenkins & Reinsel, 2015). Where they tend to fall short, rather predictably, is with the nonlinear, fast-changing character of contemporary demand, and it is this shortfall that opened the door to machine learning and, later, to deep learning, both of which can capture temporal dependencies that classical tools tend to smooth over or simply miss (Zhang, Eddy Patuwo & Hu, 2018).

Exogenous variables, weather chief among them, shape short-term load in ways that are well documented but not always well modelled. Temperature swings drive heating and cooling demand quite directly, while calendar effects, weekday against weekend, public holidays, and so on, introduce their own periodic rhythm into the data (Taylor & McSharry, 2007). Long Short-Term Memory networks have offered a meaningful improvement here, given their capacity to retain long-range dependencies while avoiding the vanishing-gradient problem that hampers plain recurrent networks (Hochreiter & Schmidhuber, 1997), yet even these architectures do not naturally surface which inputs mattered, or why, particularly once the input space becomes heterogeneous. Add to this the sheer diversity of exogenous data sources, weather readings that are temporally and spatially inconsistent, socioeconomic indicators that introduce their own noise, and the challenge of folding all of it into one coherent model only deepens (Wang, Chen, Kang & Xia, 2019).

Two problems, in particular, motivate this study. The first concerns the incomplete way weather variables are typically incorporated. Many forecasting models still lean mostly on historical consumption figures, treating temperature, humidity, precipitation, and wind speed as secondary inputs at best, and this tends to produce larger errors precisely when weather is most volatile (Hong & Fan, 2016). The relationship between weather and demand is, besides, highly nonlinear and dynamic, which makes it genuinely difficult for conventional methods to describe with any precision (Deb, Zhang, Yang, Lee & Shah, 2017). Deep learning is, in principle, well suited to learning such nonlinearity, yet a model that integrates weather variables properly and captures their nonlinear bearing on demand remains, arguably, more the exception than the rule (Lago, Marcjasz, De Schutter & Weron, 2021). The second problem is interpretability. Deep architectures such as LSTMs can reach high accuracy while offering very little transparency into how a given prediction actually came about (Zhang, Liu & Wang, 2021), and this matters because utility managers, engineers, and policymakers need models they can, to some extent, trust and defend, not merely ones that perform well on a held-out test set (Doshi Velez & Kim, 2017).

The aim of this study, accordingly, is to develop and evaluate an interpretable deep learning model for short-term energy demand forecasting capable of supporting smart grid operations with a reasonable degree of both accuracy and transparency. In pursuit of that aim, the study sets out to: design and implement a hybrid deep learning forecasting architecture combining convolutional and recurrent components that can accommodate historical demand, weather, calendar, and engineered temporal features jointly; evaluate and interpret the resulting model using explainable

artificial intelligence, specifically SHAP; and compare the proposed architecture against simpler statistical and deep learning baselines on both accuracy and computational grounds. The remainder of the article summarises related work, describes the proposed multi-branch architecture and the methodology used to evaluate it, reports the resulting evidence, and closes with conclusions.

2. Review of Related Work

A fairly large body of work has, by now, applied CNN, LSTM, and hybrid CNN-LSTM architectures to short-term load forecasting, and a recurring pattern emerges once these studies are read side by side. Weather is either omitted outright, folded in only partially, or included without much scrutiny of how it actually interacts with historical demand, and interpretability is addressed even less often. Kim and Cho (2019), for instance, proposed a CNN-LSTM hybrid for residential prediction that performed well against several recurrent baselines, yet the underlying dataset contained no weather variables at all, so the architecture was never really tested against weather-driven nonlinearity. Ullah et al. (2024) reported a strong hybrid CNN-LSTM result on Pakistani transmission data using seventeen engineered features, none of which were raw weather readings, despite the same paper's own literature review stressing climate as a demand driver. Qureshi, Arbab and Rehman (2024) compared six optimiser settings for a plain LSTM on hospital load data and reached 95% R^2 with Adam, but again without weather, occupancy, or any interpretability layer. Where weather is included, it tends to be treated somewhat gingerly. Shi, Miao and Ren (2023) built a CNN-BiLSTM model with attention and Bayesian hyperparameter search, and did run an ablation confirming that weather-inclusive variants performed modestly better across four seasonal subsets, though the attention mechanism itself was framed purely as an efficiency device rather than an explanatory one. Ariyo et al. (2025) achieved an R^2 of 0.87 with a hybrid LSTM-CNN on a Nigerian renewable microgrid using monthly weather-averaged data, outperforming ARIMA and exponential smoothing, but interpretability was not addressed at any point. Nabavi, Mohammadi, Motlagh, Tarkoma and Geyer (2024) went a little further, decomposing both load and weather signals with a discrete wavelet transform ahead of an LSTM, and did apply a post-hoc permutation-based feature importance analysis, one of comparatively few studies in this space to do so, though no SHAP or attention mechanism was applied to the core network itself.

A separate cluster of studies has looked specifically at interpretability, generally with mixed success in tying it back to weather. Shaikh et al. (2022) proposed an N-BEATS model that achieves interpretability structurally, through trend and seasonality basis functions, rather than via a post-hoc tool, though the study incorporated no weather covariate whatsoever, only a bare day indicator. Yang et al. (2025) fused Prophet-decomposed load components with SHAP-filtered weather variables ahead of a ConvLSTM-3D network and reported the lowest error in most tested conditions, yet SHAP in that study explains which weather variables mattered to the historical pattern rather than why the network produced a specific forecast at a specific time, leaving the core mechanism still fairly opaque. Sarker et al. (2024), working on the same Panama dataset used in the present study, combined an attention-based 1D-CNN-GRU with federated learning and pruning, and reported branch-level SHAP rankings that varied sensibly by dataset, though the added architectural complexity carries its own computational cost, one the paper does not dwell on at length. Karima et al. (2026) tuned an LSTM-RNN hybrid through hyperparameter optimisation and reported strong accuracy gains, though interpretability was not a stated objective of that work. Reviews of the field arrive at a broadly consistent verdict. Aguiar and Pérez-Juárez (2023), surveying roughly thirty studies across a six-year window, found CNN, LSTM, and CNN-LSTM hybrids to dominate the literature and discussed at length how temperature and humidity

shape consumption, without, however, offering any empirical validation of their own. Eren and Küçükdemiral (2024), pooling over 190 studies, confirmed the same pattern at a larger scale: weather variables are inconsistently incorporated, many studies rely purely on historical load, and explainability receives comparatively little structured treatment, with no dedicated benchmarking of SHAP, LIME, or attention-based methods against one another. Ucar and Kaygusuz (2025) tested 264 architecture-optimiser-activation combinations on the PJM East dataset and reached an R^2 of 0.9976 with a tuned BiLSTM, all while relying purely on historical consumption and calendar-derived features, despite noting in their own introduction that short-term forecasting usually requires weather information. Abedin et al. (2025) compared LSTM and GRU with meteorological inputs on Bangladeshi substation data and found GRU consistently ahead, though the weather variables there appear concatenated rather than deeply fused, again with no interpretability mechanism attached. Abimbola (2025), reviewing forecasting challenges in smart cities more broadly, singled out poor data quality and the absence of standardised frameworks as persistent obstacles, a concern that sits alongside, rather than resolves, the weather and interpretability gaps identified above.

Taken together, this literature leaves two connected gaps that the present study tries to address directly. First, studies that incorporate weather meaningfully rarely pair that inclusion with a genuine interpretability mechanism, and studies that do offer interpretability tend to do so on architectures that exclude weather altogether, or include it only as an afterthought. Second, where feature importance is reported, it is almost always reported as a single, fixed ranking rather than as something that might reasonably shift across a multi-hour forecast horizon, even though there is little reason to expect the relative influence of, say, historical demand against calendar effects to remain constant from hour one to hour twenty-four of a forecast. It is this pair of gaps, weather integration on one hand and horizon-aware interpretability on the other, that motivates the three-branch architecture described in the next section.

2. Proposed System

Figure 1 presents this architecture of the proposed system: a high-level view of the major components and how data move through each stage, from acquisition and preparation through to forecasting, evaluation, interpretability, deployment, and visualization.

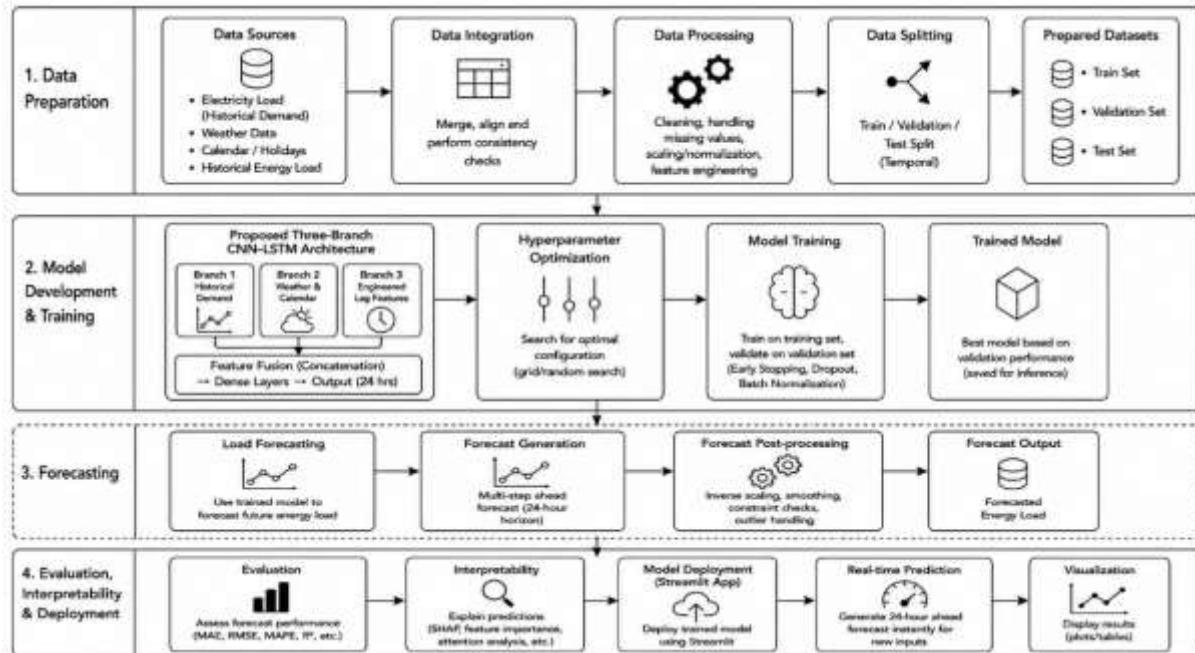


Figure 1: Architecture of the proposed system

It doubles as the implementation blueprint for the three-branch hybrid CNN-LSTM model, showing how its constituent modules are meant to collaborate toward accurate, interpretable forecasts. Four phases make up the architecture: Data Preparation, Model Development and Training, Forecasting, and finally Evaluation, Interpretability and Deployment. Each has its own function, and the output of one feeds directly into the next, which is what gives the overall workflow its systematic character.

The architecture proposed here is a three-branch hybrid CNN-LSTM model for short-term electricity demand forecasting as shown in Figure 2. It aims to learn the nonlinear relationships among historical demand, weather, calendar information, and engineered temporal features somewhat more effectively than a single, undivided pathway could. The overall system can be written as

$$\hat{F}_{t+h} = F(P_{t-n+1}, W_{t-n+1}, C_{t-n+1}, L_{t-n+1}) \quad (1)$$

where P denotes historical electricity demand, W the weather variables, C the calendar and time variables, and L the engineered lag and rolling features; n is the length of the historical input window, fixed at 168 hours, h is the forecasting horizon, and F is the proposed CNN-LSTM forecasting function producing the predicted demand.

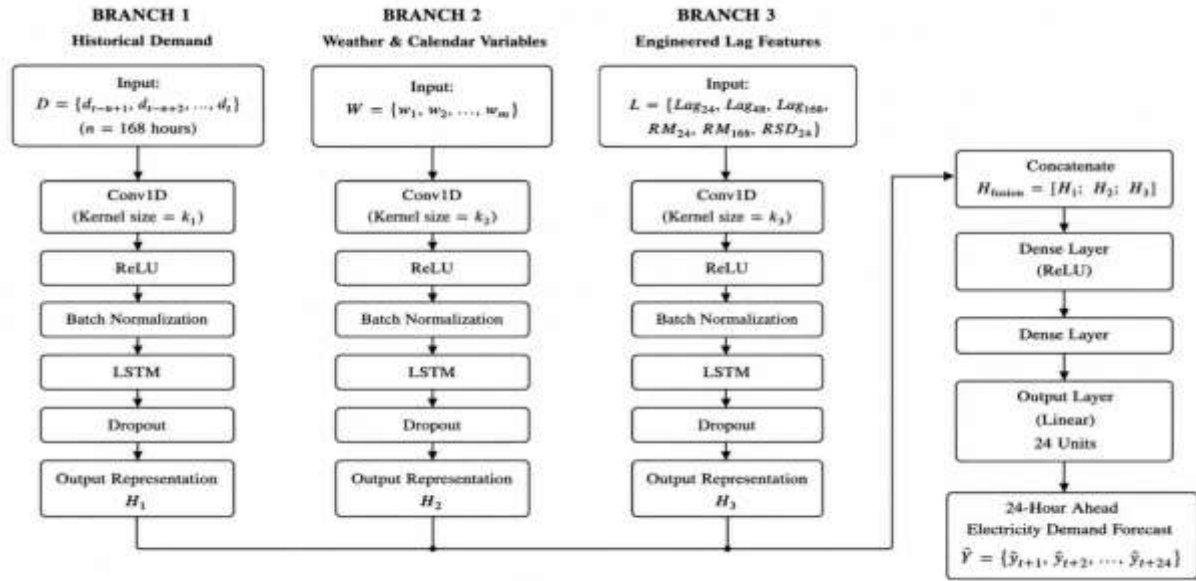


Figure 2: Proposed three branch Hybrid CNN-LSTM model

Branch 1 learns short-term temporal variation directly from the historical demand sequence. A one-dimensional convolutional layer first extracts local patterns, sudden fluctuations, peak periods, and short-term consumption behaviour, without these needing to be hand specified, and the resulting feature maps then pass to an LSTM layer that captures longer dependencies:

$$F_1 = \sigma(K_1 * D + b_1), \quad H_1 = \text{LSTM}(F_1) \quad (2)$$

Branch 2 handles the weather and calendar variables. A convolutional layer learns local relationships among the environmental inputs, after which an LSTM models how these evolve across the input window:

$$F_2 = \sigma(K_2 * W + b_2), \quad H_2 = \text{LSTM}(F_2) \quad (3)$$

Branch 3 receives the engineered lag and rolling statistics, information that summarises consumption history at several temporal scale. It follows the same convolution-then-LSTM pattern as the other two branches:

$$F_3 = \sigma(K_3 * L + b_3), \quad H_3 = \text{LSTM}(F_3) \quad (4)$$

where K_1 , K_2 , and K_3 are the convolutional kernels and b_1 , b_2 , and b_3 the associated bias terms for the three branches, and $\sigma(\cdot)$ denotes the Rectified Linear Unit activation. Each branch, independently produced a high-level representation of its respective input category, and these three outputs are merged in a fusion layer through simple concatenation:

$$H(\text{fusion}) = \text{Concat}(H_1, H_2, H_3) \quad (5)$$

The fused vector then passes through a fully connected dense layer that learns whatever nonlinear relationships remain among the extracted features, before a final output layer generates the predicted electricity demand across the full twenty-four-hour horizon:

$$Z = \mathcal{O}(W_f \cdot H(\text{fusion}) + b_f), \quad \hat{Y} = W_o Z + b_o \quad (6)$$

where W_f and b_f are the weight matrix and bias of the dense layer, $\mathcal{O}(\cdot)$ is the nonlinear activation applied there, and $\hat{Y}$ is the resulting sequence of predicted values spanning the twenty-four-hour-ahead demand forecast. Training minimises Mean Squared Error between predicted and actual demand, with the Adam optimiser updating network weights iteratively across epochs while a held-out validation split guards against overfitting.

Interpretability was layered onto the trained model through SHAP, applied by way of a GradientExplainer capable of attributing a prediction back to the three branches and, within each branch, to the individual engineered inputs.

4. Experiments and Discussion

4.1 Experimental Setup

All models (proposed and baselines) were implemented in Python using TensorFlow and Keras, with scikit-learn used for preprocessing and evaluation and SHAP used for interpretability.

4.2 Dataset and Preprocessing

The study draws on the publicly available Panama short-term electricity load dataset, an hourly series assembled from Panama's national grid operator together with supporting meteorological and calendar sources, and spanning January 2015 to June 2020. Retaining only one weather station, rather than the three represented in the original source, was a deliberate simplification intended to keep the pipeline transferable to single-location datasets in future work. Taken together, the merged table used here runs to 47,880 hourly rows across eight features. Historical demand was reshaped into a 168-hour input window per sample, weather and calendar variables were aligned to the same window, and a further six engineered features, lag-24, lag-48, lag-168, and rolling mean and standard deviation statistics at 24- and 168-hour windows, were computed to summarise consumption history at several temporal horizons. We used 80% of the dataset as the training set and 20% as the test set. The validation set is 20% of the training set.

4.3 Model Training and Evaluation

The proposed multi-branch model was benchmarked against three baselines trained under matched conditions: a standalone CNN, a standalone LSTM, and a sequential CNN-LSTM in which a single convolutional stage feeds a single LSTM stage rather than three parallel branches. Each model was trained on the same 168-hour input window, the same train-test split, and the same random seed, so that any performance difference could reasonably be attributed to architecture rather than to incidental variation in the experimental setup. Table 1 shows the model configurations and training.

Table 1: Shared global configuration used by all models

Hyperparameter	Value
Input window (lookback)	168 hours
Forecast horizon	24 hours
Batch size (training)	64
Epochs (maximum)	100
Optimizer	Adam
Learning rate	0.0005
Loss function	Huber loss
Evaluation metric	MAE
Conv1D filters	64
LSTM units	64
Dense units	128
Dropout rate	0.30
Kernel size (short/local convolution)	3

Daily kernel size	24
Early stopping patience	20 epochs (monitors val_loss, restores best weights)
Reduce-LR-on-plateau patience	8 epochs
Reduce-LR factor	0.5
Minimum learning rate	1e-6

Performance was assessed using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2), computed across the full 24-hour forecast horizon and all test windows, together with training time and per-sample inference latency, since a model's practical value depends on more than its error figures alone.

4.4. Results and Discussion

In this section, we describe the experimentally obtained results to demonstrate the effectiveness of our proposed model

4.4.1 Aggregate Predictive Performance

Table 2 places the four models side by side on aggregate test-set metrics, and the pattern that emerges is not, on first inspection, entirely the one this study perhaps set out to demonstrate. The proposed multi-branch CNN-LSTM recorded an RMSE of 86.69, an MAE of 60.687, a MAPE of 5.169%, and an R^2 of 0.780, figures that sit almost exactly alongside those of the considerably simpler LSTM baseline, whose RMSE of 86.33, MAE of 60.465, identical MAPE of 5.173%, and marginally higher R^2 of 0.782 suggest that added architectural complexity did not translate into a clearly superior aggregate result. The gap between the two is, in practical terms, vanishingly small, smaller certainly than one might expect given that the proposed architecture carries close to four times as many trainable parameters, 246,424 against 65,688. Both hybrid and pure-recurrent designs comfortably outperformed the standalone CNN, which lagged on every measure, and outperformed, if only modestly, the simpler sequential CNN-LSTM baseline.

Table 2: Aggregate test-set performance of the four forecasting models.

Model	RMSE	MAE	MAPE (%)	R^2	Parameters
LSTM (baseline)	86.33	60.465	5.173	0.782	65,688
Proposed Multi-Branch CNN-LSTM	86.69	60.687	5.169	0.780	246,424
CNN-LSTM (baseline)	89.90	63.661	5.429	0.763	81,240
CNN (baseline)	99.75	70.706	6.037	0.708	27,544

4.4.2 Horizon-Dependent Behaviour

Aggregate figures, it turns out, conceal a good deal of variation once the twenty-four-hour horizon is examined hour by hour. At the very start of the horizon as shown in figure 4, the proposed multi-branch model is clearly the strongest of the four, recording an RMSE of 36.96 at hour one against 50.23 for the sequential CNN-LSTM and over 69 for both single-pathway baselines, an advantage that persists, gradually narrowing, through roughly the first five hours. From around the sixth or

seventh hour onward, though, something shifts. The LSTM baseline, whose error climbs more slowly through the middle stretch of the horizon, overtakes the proposed model somewhere between hours eight and ten, and by hour twelve the gap is fairly pronounced, an RMSE of 91.53 for the LSTM baseline against 97.72 for the proposed model. By the final hours of the horizon the proposed model recovers some ground, closing at the best RMSE of the four models at hour twenty-four. This suggests a horizon-dependent trade-off that the aggregate metrics in Table 3 mask almost entirely: the separated demand branch appears to exploit strong short-lag autocorrelation well near the start of the horizon and the weather branch perhaps regains relevance toward its end, while in the middle stretch, where recent history has decayed in relevance but the next daily cycle has not yet come back into view, the plain LSTM baseline seems to cope somewhat better, possibly because its undivided input allows it to blend signals more flexibly than a fixed three-way branch structure permits.

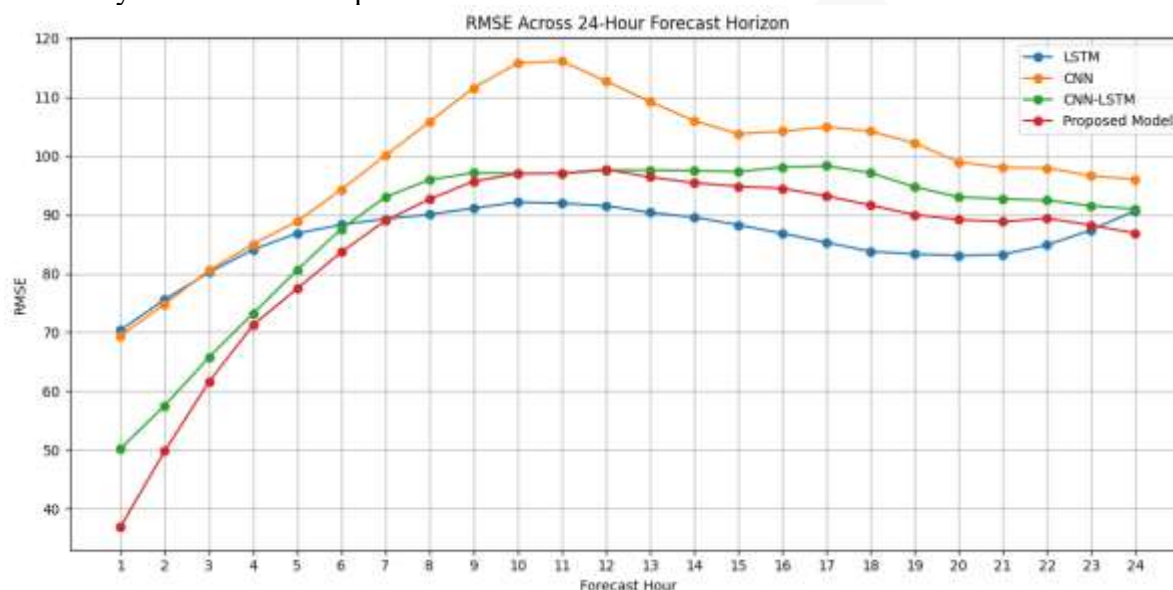


Figure 4: RMSE across the 24-hour forecast horizon for all four models

4.4.3. Computational Cost

Table 3 reports training time and inference latency alongside parameter counts, a dimension of the results that pure accuracy metrics tend to leave out and one that matters more than is sometimes acknowledged in the forecasting literature. The standalone CNN was, unsurprisingly, the cheapest across every measure, training in 143.47 seconds and achieving the fastest inference latency of the group. The proposed multi-branch model, given its three parallel branches, was by some distance the most expensive to train, requiring 774.07 seconds, roughly two and a half times the training time of the LSTM baseline and nearly four times its parameter count. Inference latency followed a broadly similar, if less dramatic, ordering, with the proposed model's 0.657 milliseconds per sample still comfortably within the bounds of practical real-time deployment, even if it is roughly double that of the standalone CNN.

Table 3: Training time, inference latency, and parameter count for the four models.

Model	Training Time (s)	Inference Latency (ms/sample)	Trainable Parameters
CNN	143.47	0.195	27,544
LSTM	297.28	0.347	65,688
CNN-LSTM	416.07	0.350	81,240
Proposed Multi-Branch CNN-LSTM	774.07	0.657	246,424

4.4 SHAP-Based Interpretability

The SHAP-based analysis reported in Table 4 speaks fairly directly to one of the two problems motivating this study, namely the limited attention paid to model transparency in much of the existing short-term load forecasting literature. Historical demand accounted for the largest share of attributed importance, at 63.56%, with the weather and calendar branch contributing 24.29% and the lag and rolling statistics branch the remaining 12.16%. It is worth noting that the weather and calendar branch here draws on a single-location weather representation rather than the multi-location profile available in the original source data, a simplification retained for the sake of transferability, and this may partly explain why that branch's contribution did not exceed that of historical demand.

Table 4: Branch-level SHAP contribution to the proposed model's forecasts.

Branch	Mean SHAP Value	Relative Contribution (%)
Historical Demand	0.000374	63.56
Weather and Calendar	0.000143	24.29
Lag and Rolling Statistics	0.000071	12.16

A further, and arguably more revealing, layer of the analysis examined how feature importance shifts across the forecast horizon rather than remaining fixed. Historical demand remained the single most influential predictor at the shortest horizons examined, hours one and three, before giving way to the cyclical hour-of-day encodings at every longer horizon sampled, six, nine, twelve, fifteen, eighteen, twenty-one, and twenty-four. Rather than a single static ranking, then, the dominant driver of the forecast appears to change in a fairly interpretable way as the prediction moves further from the present, a nuance that a global, horizon-agnostic importance summary would have missed entirely.

5. Conclusion

In this paper, we presented multi-branch hybrid CNN-LSTM model for short-term electricity demand forecasting, one built specifically to address the twin problems of incomplete weather integration and limited model interpretability identified in the existing literature. A three-branch architecture was proposed, routing historical demand, weather and calendar variables, and engineered lag features through separate convolutional-recurrent pathways before fusing them for a twenty-four-hour-ahead forecast, and SHAP-based attribution was layered onto the trained model to expose both branch-level and horizon-dependent feature importance. Evaluated against

a standalone CNN, LSTM, and a sequential CNN-LSTM on Panama's national demand data, the proposed model achieved aggregate accuracy essentially on par with the considerably simpler LSTM baseline, while offering a clear early-horizon advantage, a somewhat weaker middle-horizon showing, and a meaningfully higher computational cost. The interpretability results, for their part, showed that historical demand dominates attributed importance overall, though the balance shifts toward cyclical time-of-day features as the forecast horizon lengthens, a finding that a single, static feature-importance summary would not have surfaced. On the whole, then, the proposed architecture does not offer a decisive accuracy improvement over a simpler recurrent baseline, and this is probably worth stating candidly rather than understating. What it does offer is a somewhat more transparent account of how a forecast was reached, together with a modest early-horizon edge, and whether that trade-off is worthwhile will likely depend on the deployment context in question.

A few limitations are worth stating plainly. The dataset, though genuinely representative of a national grid, is a single grid, and the horizon-dependent behaviour reported above may or may not generalise to demand series with different autocorrelation structure or weather sensitivity. The weather representation used here was deliberately narrowed to a single station, which likely constrains the ceiling on the weather branch's attributed importance. Finally, the SHAP analysis treats the three branches as contributing independently, which is a reasonable first approximation but probably understates whatever interaction exists between, say, weather and historical demand in reality.

A few recommendations follow reasonably directly from these findings. First, given that the middle portion of the forecast horizon is where the proposed architecture is weakest relative to the LSTM baseline, an adaptive fusion mechanism, an attention-based or gated combination of branch outputs rather than fixed concatenation, seems a fruitful direction for future work. Second, the SHAP findings suggest that further gains in transparency, rather than in raw accuracy, may be the more productive avenue for this line of research, and future studies might usefully extend the interpretability pipeline to capture interaction effects between branches rather than treating them as independent contributors. Third, organisations considering deployment in genuinely resource-constrained settings might reasonably favour the LSTM baseline as a lighter-weight alternative, reserving the fuller multi-branch model for contexts where the interpretability benefit specifically justifies the additional training and inference overhead. Finally, it would be worth testing the proposed architecture on grids of varying size and climate, since the present evaluation, like a good deal of the literature reviewed, rests on a single dataset, and the horizon-dependent behaviour observed here may not generalise beyond it.

References

- Abedin, J., Ikram, A. I., Islam, M. R., Islam, M. J., Ahemad, F., & Elias, M. (2025). Deep learning model for electrical load forecasting considering meteorological effect. Paper presented at the 2023 International Conference on Next-Generation Computing, IoT and Machine Learning (NCIM 2025), Gazipur-1707, Bangladesh.
- Abimbola, O. (2025). Energy demand forecasting with deep learning for smart cities. *International Journal of Research Publication and Reviews*, 6(4), 5950-5959.
- Aguiar-Pérez, J. M., & Pérez-Juárez, M. Á. (2023). An insight of deep learning based demand forecasting in smart grids. *Sensors*, 23(3), 1467.
- Ariyo, B. O., Adesina, L. M., Ogunbiyi, O., Musa, A., Ojuolape, B. J., & Balogun, M. O. (2025). AI-driven demand forecasting for enhanced energy management in renewable microgrids: A hybrid LSTM-CNN approach. *Journal of Engineering Research and Reviews*, 2(1), 1-11.
- Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2015). *Time series analysis: Forecasting and control* (5th ed.). Wiley.
- Deb, C., Zhang, F., Yang, J., Lee, S. E., & Shah, K. W. (2017). A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 74, 902-924.
- Doshi Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.
- Eren, Y., & Küçükdemiral, İ. (2024). A comprehensive review on deep learning approaches for short-term load forecasting. *Renewable and Sustainable Energy Reviews*, 189, 114031.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.
- Hong, T., & Fan, S. (2016). Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3), 914-938.
- Karima, N. N., Ahmad, S., Islam, A., Huda, A. S. N., Awal, L. J., Halim, S. A., Mokhlis, H., Ali, M. S., & Mubarak, S. (2026). Enhancing short-term load forecasting using hyperparameter-optimized deep learning approaches. *Energies*, 19(3), 705.
- Kim, T.-Y., & Cho, S.-B. (2019). Predicting residential energy consumption using CNN-LSTM neural networks. *Energy*, 182, 72-81.
- Lago, J., Marcjasz, G., De Schutter, B., & Weron, R. (2021). Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293, 116983.
- Nabavi, S. A., Mohammadi, S., Motlagh, N. H., Tarkoma, S., & Geyer, P. (2024). Deep learning modeling in electricity load forecasting: Improved accuracy by combining DWT and LSTM. *Energy Reports*, 12, 2873-2900.
- Qureshi, M., Arbab, M. A., & Rehman, S. ur. (2024). Deep learning-based forecasting of electricity consumption. *Scientific Reports*, 14, 6489.
- Sarker, M. A. A., Shanmugam, B., Azam, S., & Thennadil, S. (2024). Enhancing smart grid load forecasting: An attention-based deep learning model integrated with federated learning and XAI for security and interpretability. *Intelligent Systems with Applications*, 23, Article 200422.
- Shaikh, A. K., Nazir, A., Khan, I., & Shah, A. S. (2022). Short term energy consumption forecasting using neural basis expansion analysis for interpretable time series. *Scientific Reports*, 12, 22562.

- Shi, H., Miao, K., & Ren, X. (2023). Short-term load forecasting based on CNN-BiLSTM with Bayesian optimization and attention mechanism. *Concurrency and Computation: Practice and Experience*, 35(17), e6676.
- Taylor, J. W., & McSharry, P. E. (2007). Short-term load forecasting methods: An evaluation based on European data. *IEEE Transactions on Power Systems*, 22(4), 2213-2219.
- Ucar, M. T., & Kaygusuz, A. (2025). Short-term energy consumption forecasting analysis using different optimization and activation functions with deep learning models. *Applied Sciences*, 15(12), 6839.
- Ullah, K., Ahsan, M., Hasanat, S. M., Haris, M., Yousaf, H., Raza, S. F., Tandon, R., Abid, S., & Ullah, Z. (2024). Short-term load forecasting: A comprehensive review and simulation study with CNN-LSTM hybrids approach. *IEEE Access*, 12, 111858-111881.
- Wang, Y., Chen, Q., Kang, C., & Xia, Q. (2019). Clustering of electricity consumption behavior dynamics toward big data applications. *IEEE Transactions on Smart Grid*, 7(5), 2437-2447.
- Yang, X., Zhao, S., Li, K., Chen, W., Zhang, S., & Chen, J. (2025). An optimized method for short-term load forecasting based on feature fusion and ConvLSTM-3D neural network. *Frontiers in Energy Research*, 12, Article 1501963.
- Zhang, G., Eddy Patuwo, B., & Hu, M. Y. (2018). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14(1), 35-62.
- Zhang, Y., Liu, Z., & Wang, J. (2021). Deep learning for smart grid energy forecasting: A comparative study of LSTM and Transformer models. *Renewable and Sustainable Energy Reviews*, 144, 111024.