Research Article
2025, Vol. 01, Issue. 2, 48-61
doi.xxx.

Journal of Innovations in
Social and Applied Sciences

Deep Learning-Based Energy Demand Forecasting for Sustainable Smart Cities

Amr E. Elshora
Department of Computer Science and Information, Higher Institute of Management in Kafrelsheikh – Ministry of Higher Education, Cairo, Egypt
Corresponding Author: amr.alshura@himit-kfs.edu.eg

Abstract

Reliable short-term electricity forecasts have a direct role in smart-city operations, including storage scheduling, peak control, and demand response. In this paper, we examine a direct multi-horizon forecasting strategy using the public Individual Household Electric Power Consumption dataset. The original 2,075,259 minute-level measurements were aggregated to 34,589 hourly observations. For each forecasting sample, the model uses 72 hours of seven electrical variables to predict Global Active Power for the next 24 hours. The chronological order of the series was preserved by assigning 70% of the data to training, 15% to validation, and 15% to testing, which produced 24,117, 5,165, and 5,166 windows, respectively. The comparison includes a persistence baseline, a standard LSTM, a GRU, and a Compact Stacked LSTM. On the held-out test period, the GRU produced the lowest error, with an RMSE of 0.6181 kW and an MAE of 0.4687 kW. The standard LSTM followed closely with an RMSE of 0.6262 kW, while the stacked LSTM recorded 0.6509 kW. Persistence performed considerably worse, with an RMSE of 0.9177 kW. These results support recurrent modelling for this forecasting task, but they also show that simply adding recurrent depth does not necessarily improve household-demand forecasts.

Article History

Manuscript Received
June 12, 2025

First Revision
December 08, 2025

Final Acceptance
December 29, 2025

Keywords: energy demand forecasting; smart cities; deep learning; LSTM; smart grid; household electricity; time-series forecasting

1 Introduction

Smart-city energy systems generate much more operational data today than they did a decade ago, yet the value of these data depends on whether they lead to better decisions. Smart meters capture consumption changes at fine temporal resolution, while storage, distributed generation, demand response, electric vehicles, and flexible tariffs give operators a wider range of possible responses. Forecasting connects these observations with operational action. If near-future demand can be estimated reliably, a utility can prepare for peaks, schedule storage, and coordinate flexible loads before conditions become critical. Load forecasting, therefore, is no longer only a predictive-modelling exercise; it has become part of how the modern smart grid operates [1], [2].

Household electricity usage typically adheres to recognizable daily and weekly patterns, although these patterns are never entirely regular. Demand can alter quickly when someone is home, a heavy-duty appliance is used, climate conditions change, or an unusual event occurs. The value of cyclic structure and variant shocks can allegedly deduce the interest in recurrent models in residential forecasting. Kong et al. demonstrated the suitability of recurrent learning for short-term household load prediction, while Alhussein, Aurangzeb, and Haider proved that convolutional feature extraction can be paired with LSTM sequence modelling [3], [4]. Moreover, the authors track down an important limitation in their findings. Specifically, the results for an aggregated feeder cannot be equated with the household level. Energy forecasting has a substantial number of models due to deep learning. Gated state updates in LSTM networks enable information storage for extended periods. Sequence-to-sequence models discover correlations among overlapping temporal blocks. Convolutional layer-based models, attention-based models, graph-recurrent models, and enhanced CNN-LSTM models aim to identify local, long-range, and spatial-temporal structures [5]–[8]. More sophisticated architecture, however, does not necessarily produce a similar reduction in prediction error [5]. This trade-off is relevant particularly in smart-city, where computational requirements, maintenance, robustness, and ease of deployment may be as important as a slight improvement in accuracy.

The experiments in this paper operate within one clearly specified forecasting setup and all experimental values we reported are based on our own runs using public UCI data. To begin with, we aggregate the minute-level measurements into hourly observations. Each sample subsequently has a 72-hour multivariate history that will forecast Global Active Power directly for the next 24 hours. The context of earlier studies like [9] is adopted, although the performance values reported in them are not stated as results of the experiments.

This paper presents four contributions. This is a 24-hour residential forecasting task where seven smart-meter variables from the previous 72 hours are used as inputs. LSTM, GRU, Compact Stacked LSTM tested on same chronological partition of UCI series for second time and persistence. The RMSE, MAE, MAPE, and R^2 values reported here are based on predictions made in this study rather than from earlier studies. The experiments also address a practical question that can be easy to overlook: would further recurrent depth actually help at the household scale, or would a simpler recurrent model suffice?

The experimental question is the framework of the remainder of the paper. The most relevant forecasting literature is reviewed in Section 2, the dataset and preprocessing procedure is discussed in Section 3, and the Compact Stacked LSTM formulation is described in Section 4. The outcomes of the experiment are presented and discussed in Section 5, while the final section brings together the fundamental findings and presents the limitations that must be borne in mind.

2 Energy Demand Forecasting: Background and Recent Advances

2.1 Statistical and Machine-Learning Forecasting Approaches

Electric-load forecasting has a long history that predates the recent rise of deep learning. Regression, autoregressive models, exponential smoothing, support-vector methods, and conventional neural networks have all been applied in practice. Hong and Fan make an important distinction between point and probabilistic forecasting, showing that the appropriate formulation depends on factors such as forecast horizon, aggregation level, available explanatory variables, and the decision for which the forecast will be used [1]. More recent smart-grid forecasting reviews arrive at much the same conclusion: the choice of model needs to reflect both the scale and the purpose of the application [2].

A smart-city forecasting target may be a single building, a distribution feeder, or an entire urban area, and these levels do not behave in the same way. Household demand is especially volatile. Even a strong neural model should therefore be compared with simple baselines and tested on chronological data rather than randomly shuffled observations. Weather and calendar information can improve a forecast when those variables would genuinely be available at prediction time [10], [11]. Allowing future information to enter preprocessing or model inputs before it would be known in real operation produces optimistic accuracy and an unrealistic forecasting scenario.

Using a deep model does not make careful input design any less important. Subbiah and Chinnappan combined filter- and wrapper-based feature selection with LSTM and reported improved short-term forecasts on European load and weather data [12]. Jahani, Zare, and Mohammad Khanli followed another approach, pairing sequential-pattern mining with LSTM to represent variable-length relationships between demand and meteorological variables; their best microgrid case achieved an R^2 of 0.951 [13]. Although the methods differ, both studies point to the same practical issue: having informative predictors available at the right time can be just as important as the network used to process them.

2.2 Deep Learning for Short-Term Energy Demand Prediction

LSTM remains a widely used reference model for load forecasting because its gated memory offers a natural way to represent recurring temporal patterns. Kong et al. applied LSTM to short-term residential load prediction [3]. Using the same UCI household dataset examined in the present study, Hyeon et al. compared a conventional LSTM with sequence-to-sequence and attention-based alternatives [9]. Choi, Cho, and Kim also studied LSTM-based demand forecasting for energy-sustainability monitoring [14]. Looking more broadly across architectures, Gasparin, Lukovic, and Alippi showed why model claims are easier to interpret when competing approaches are tested under consistent data conditions [5].

Much of the more recent work combines several modelling ideas instead of relying on a single recurrent component. Eskandari, Imani, and Parsa Moghaddam combined convolutional processing with LSTM/GRU units [15]. Saeed, Paul, and Seo developed a CNN-LSTM model with channel communication [16], whereas Zhang et al. proposed a multi-task CNN-LSTM architecture for short- and medium-term forecasting [17]. Other studies have considered deep neural forecasting for sustainable energy management [18], wavelet decomposition with LSTM [19], attention-enhanced CNN-LSTM [7], hybrid N-BEATS-CNN modelling [20], and explicit extraction of temporal features [21].

More recent research has also moved toward automated architecture design and multi-scale representations. Lu et al. presented an automated Transformer-LSTM framework in 2026 [22], while Chen et al. combined graph convolution with LSTM to represent multiple influencing factors together with spatial-temporal dependence [8]. Chai et al. investigated multi-batch Bayesian optimization as a way to improve LSTM load forecasts [23], and Caicedo-Vivas and Alfonso-Morales reported an LSTM application for a grid operator [24]. These advances broaden the available modelling choices, but they also increase the importance of careful benchmarking. Performance observed on one grid, sampling interval, or aggregation level cannot be assumed to transfer directly to a different forecasting setting.

3 Dataset, Preprocessing and Forecasting Setup

These experiments utilize the Individual Household Electric Power Consumption dataset which is prepared by Georges Hebrail and Alice Berard. It is available at the UCI Machine Learning Repository. Between December, 2006, and November, 2010, a household in Sceaux, France, recorded 2,075,259 one-minute observations over 47 months. UCI contains nine variables; it states about 1.25% of its rows have missing measurement values. The data set can be publically identified through DOI 10.24432/C58K54 [25].

Table 1. Numerical characteristics of the UCI household power dataset [25]

Quantity	Value	Unit / coding
Measurements	2,075,259	rows
Recorded variables	9	variables
Sampling interval	1	minute
Collection duration	47	months
Approx. missing rows	1.25	%
Global active power unit	1	kW
Global reactive power unit	1	kW
Voltage unit	1	V
Global intensity unit	1	A
Sub-metering unit	1	Wh

Past household behaviour is represented by seven measured electrical quantities: global active power, global reactive power, voltage, global current intensity, and three sub-metering channels. The UCI documentation associates sub-metering 1 mainly with the kitchen, sub-metering 2 with the laundry room, and sub-metering 3 with the electric water heater and air-conditioning load [25]. Date and time are used to build the temporal index rather than being treated as two additional electrical variables. Global active power serves as the forecasting target. The other electrical channels are used only as historical covariates, with no assumption that their future values are known beforehand.

We first resample the raw one-minute series to hourly means while preserving its chronological order. Missing intervals are interpolated along the time index before the forecasting sequences are created. This produces 34,589 hourly observations. The earliest 70% of the series forms the training set, the following 15% is used for validation, and the final 15% is reserved for testing. Means and standard deviations are calculated from the training portion only and then applied unchanged to the later periods. With a 72-hour look-back and a 24-hour target, this procedure yields 24,117 training windows, 5,165 validation windows, and 5,166 test windows. All competing models are evaluated using the same forecast origins and target windows.

Table 2. Numerical configuration of the executed Compact Stacked LSTM model

Setting	Value	Unit
Hourly aggregation	60	minutes
Input window L	72	hours
Forecast horizon H	24	hours
Training share	70	%
Validation share	15	%
Test share	15	%
LSTM hidden units	16	units
LSTM layers	2	layers
Dropout	0.20	fraction
Batch size	2048	samples

Setting	Value	Unit
Training epochs	16	epochs
Initial learning rate	0.001	dimensionless

Let d denote the number of historical variables and L the length of the look-back window. At forecast origin t, the input to the model is the following matrix:

$$X_t = [x_{t-L+1}, \dots,) \in R^{L \times d} \quad (1)$$

Here, X_t contains the historical observations available at time t, x_t denotes the d-dimensional hourly feature vector, and L specifies how many preceding hours are included. Before training, every continuous variable is standardized using statistics calculated from the training period:

$$\tilde{x}_{t,j} = \frac{x_{t,j} - \mu_j}{\sigma_j} \quad (2)$$

In (2), μ_j and σ_j are the mean and standard deviation of feature j, both estimated only from the training period. The same values are then used to transform the validation and test data. Keeping these statistics fixed prevents information from later periods from entering the standardization process.

4 Proposed Deep Energy Forecasting Model

The proposed network is intentionally compact. Two LSTM layers, each containing 16 hidden units, encode the 72-hour sequence of seven electrical measurements. A dropout rate of 0.20 is applied between the recurrent layers and again to the final hidden representation. A linear output head then generates the 24 future Global Active Power values simultaneously. This compact configuration was chosen to keep the experiment practical on CPU hardware while preserving the main modelling idea: stacked temporal representation followed by direct multi-step forecasting. The comparison LSTM and GRU are shallower, each consisting of a single 32-unit recurrent layer with the same 24-value output. Throughout this paper, Compact Stacked LSTM refers specifically to the compact two-layer LSTM configuration evaluated in these experiments; it is not presented as a newly invented LSTM cell or a new recurrent-learning algorithm.

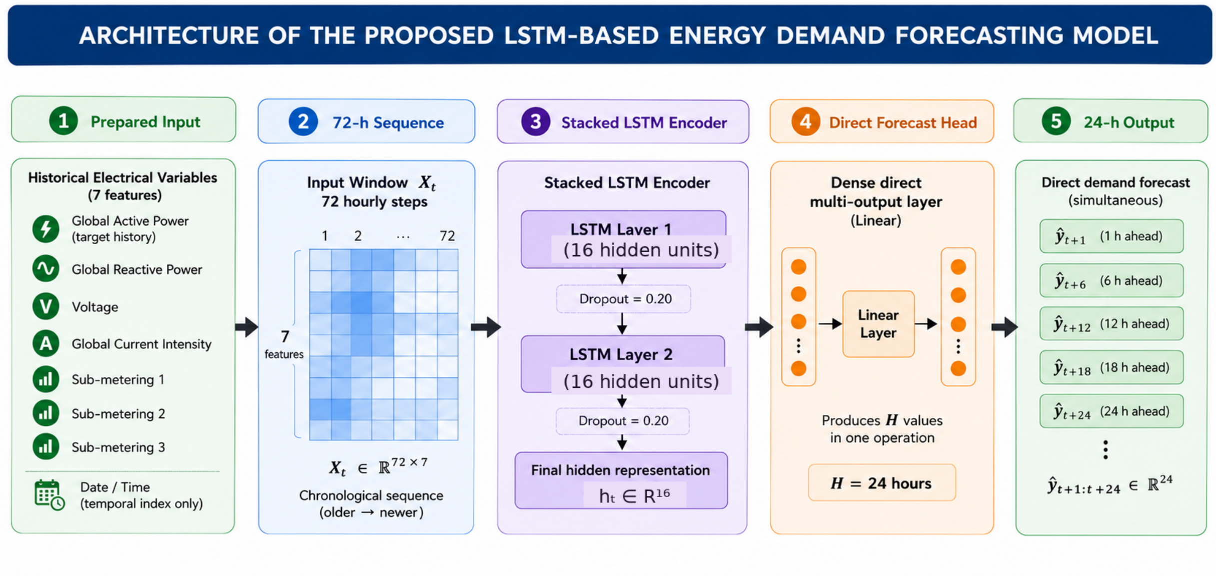

Figure 1. Architecture of the proposed LSTM-based energy-demand forecasting model.

At every time step, the normalized feature vector enters the LSTM. The forget gate determines how much of the previous memory is carried forward, while the input gate determines the contribution of the current observation to the updated state:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{3}$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{4}$$

In (3) and (4), f_t and i_t denote the forget and input gates. W and U are the trainable input and recurrent weight matrices, b is the associated bias vector, h_{t-1} is the previous hidden state, and σ denotes the logistic sigmoid function. The memory cell is then updated by:

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{5}$$

The operator ⊙ represents element-wise multiplication. The hyperbolic tangent term generates the candidate cell content, while c_t retains longer-term temporal information. The output gate and hidden state are given by:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), h_t = o_t \odot tanh(c_t) \tag{6}$$

Once the final observation in the input window has been processed, the resulting hidden state is passed to a dense layer, which produces all H future demand values in a single step:

$$\hat{y}_{t+1:t+H} = W_y h_t + b_y \tag{7}$$

The notation used in eq. (7) is as follows: W_γ and b_γ are the attributes of the prediction head and H refers to the number of future hours (e.g., for 1 hour ahead, H = 1). Instead of giving back every outcome to the network to predict the following outcome, the predictions all happen in one go. The direct multi-output approach prevents the gradual build-up of progressive recursive forecast errors during the 24-hour horizon. The model is trained to minimize a mean square error:

$$L_{MSE} = \frac{1}{NH}\sum_{i=1}^{N}\sum_{h=1}^{H}\left(y_{i,h} - \hat{y}_{i,h}\right)^2 \tag{8}$$

In (8), N denotes the number of training windows, $y_{i,h}$ is the observed demand for sample i at forecast step h, and $\hat{y}_{i,h}$ is the corresponding prediction. Adam is used to optimize the neural networks. For all recurrent models, the comparison keeps the 72-hour input history, seven historical electrical variables, 24-hour target construction, chronological partition, and held-out test windows unchanged. The persistence baseline simply repeats the final observed Global Active Power value for each of the 24 forecast steps. The standard LSTM and GRU each use one 32-unit recurrent layer, a batch size of 1024, and 8 training epochs. The Compact Stacked LSTM uses two 16-unit recurrent layers, dropout of 0.20, a batch size of 2048, and 16 epochs. Every neural model is trained with Adam at an initial learning rate of 0.001, and test predictions are generated from the state that achieved the lowest validation loss during the corresponding run.

5 Forecasting Results and Performance Analysis

5.1 Comparative Forecasting Performance

All results reported in this section were obtained from experiments performed for this study on the public UCI Individual Household Electric Power Consumption dataset. The values shown in Tables 3-5 and Figures 2-3 are not reproduced from Reference [9] or any other published benchmark. Once the data are aggregated hourly and the forecasting sequences are constructed, 5,166 windows remain in the final test period. Each window contains 24 future hourly values of Global Active Power, and the evaluation metrics are calculated over these held-out forecasts.

Table 3. Genuine held-out test results obtained from the executed UCI experiment

Model	RMSE (kW)	MAE (kW)	MAPE (%)	R^2
Persistence	0.9177	0.6608	95.55	-0.7127
Standard LSTM	0.6262	0.4737	69.60	0.2024
GRU	0.6181	0.4687	70.43	0.2230
Compact Stacked LSTM	0.6509	0.5161	81.18	0.1382

Table 3 shows a consistent pattern. Each recurrent model improves on persistence for both RMSE and MAE, although the differences among the recurrent models themselves are relatively modest.

The GRU performs best on the test set, reaching an RMSE of 0.6181 kW and an MAE of 0.4687 kW. The standard LSTM is close behind, with 0.6262 kW RMSE and 0.4737 kW MAE. The Compact Stacked LSTM records an RMSE of 0.6509 kW and an MAE of 0.5161 kW. Although this is still a substantial improvement over persistence, the additional recurrent layer does not reduce the error below that of the shallower alternatives in this run.

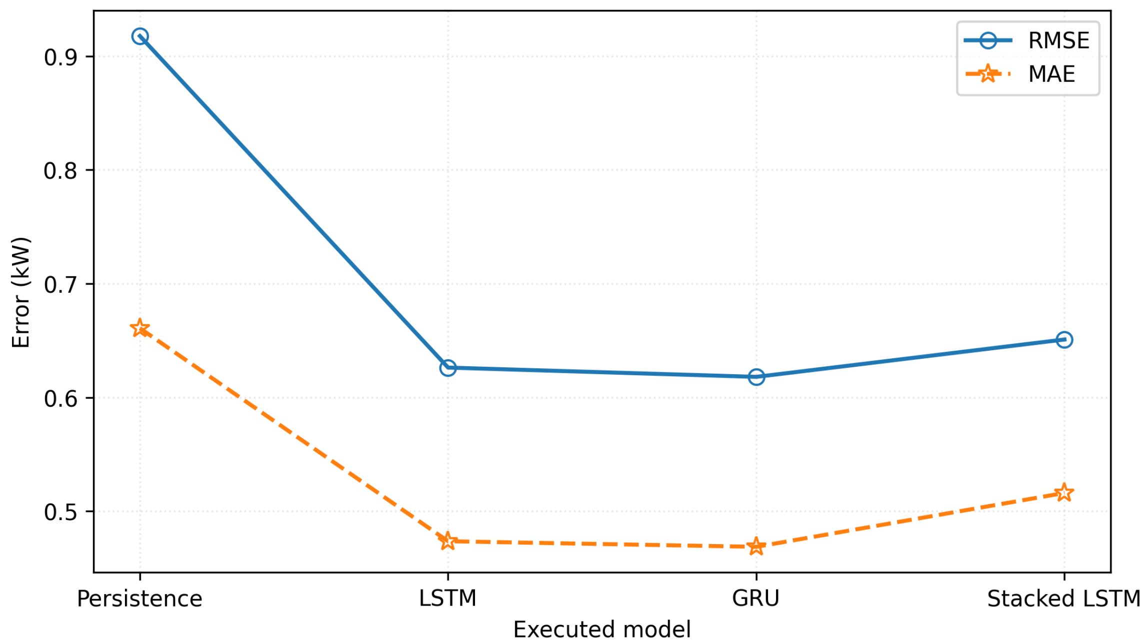

Figure 2. RMSE and MAE curves from the executed UCI test-set predictions.

Table 4. Numerical improvement over the persistence baseline

Model	RMSE reduction (%)	MAE reduction (%)	RMSE rank	R^2 rank
Persistence	0.00	0.00	4	4
Standard LSTM	31.76	28.33	2	2
GRU	32.64	29.07	1	1
Compact Stacked LSTM	29.07	21.90	3	3

Relative to persistence, the GRU reduces RMSE by 32.64%, the standard LSTM by 31.76%, and the Compact Stacked LSTM by 29.07%. MAE follows the same overall ordering. This comparison is important because it shows that network depth should not be treated as an advantage by itself. Under the settings used for this household series, the one-layer GRU is the strongest recurrent model among those tested.

5.2 Error Analysis and Model Comparison

The scale of the differences is easier to appreciate in Figure 2. Replacing persistence with any of the recurrent networks produces a clear reduction in both RMSE and MAE. The difference between the GRU and the standard LSTM, however, is small, and the stacked LSTM remains in a similar range despite having slightly higher error. Most of the improvement in this experiment therefore comes from learning temporal structure instead of simply extending the latest observation. Adding a second recurrent layer does not provide a further gain in this particular run.

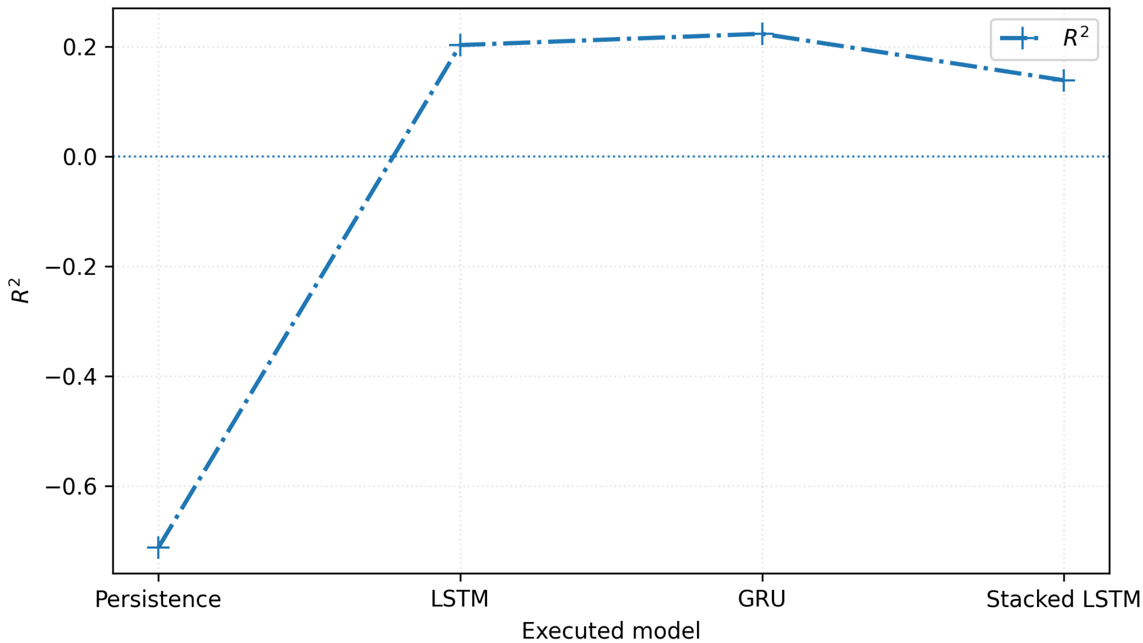

Figure 3. R² curve from the executed UCI test-set predictions.

Table 5. Numerical configuration of the executed model comparison

Model	Layers	Hidden units	Dropout	Batch size	Epochs	Outputs
Persistence	0	0	0.00	—	—	24
Standard LSTM	1	32	0.00	1024	8	24
GRU	1	32	0.00	1024	8	24
Compact Stacked LSTM	2	16	0.20	2048	16	24

Figure 3 presents the same comparison from the perspective of R^2. Persistence gives a negative R^2 of -0.7127, indicating that carrying the latest observed value through the full 24-hour horizon represents the variation in the test targets poorly. All three recurrent models raise R^2 above zero. The GRU achieves 0.2230, followed by the standard LSTM at 0.2024 and the stacked LSTM at 0.1382. Simultaneously, these scores indicate that the forecasting problem still poses challenges: while the best recurrent model captures only part of the variation in the household target at 24 hours. The comparison is specific to the evaluated configurations (as opposed to a controlled architectural ablation) because hidden sizes, batch sizes, and training durations differ across the networks. The focus is on RMSE and MAE since both values can be interpreted in the unit of Global Active Power.

The amount of variation captured by the models is evaluated by R^2. MAPE is also reported, although it is less informative for this dataset. When household demand is low, percentage errors can explode to be quite large. Hence, the ranking of models is not primarily based on MAPE.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{9}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(|y_i - \hat{y}_i|) \tag{10}$$

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}(\left|\frac{y_i - \hat{y}_i}{y_i + \epsilon}\right) \tag{11}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \acute{y})^2} \tag{12}$$

Where, n denotes the number of forecast values that are being evaluated, y_i is the observed Global Active Power value, $\hat{y}_i$ is the predicted Global Active Power value, and $\bar{y}$ is the mean observed demand, while ε is a small stabilizing constant that we use for MAPE in (9)-(12) above. The implementation uses $|y_i| + \varepsilon$ in the MAPE denominator due to the nonnegative nature of Global Active Power in this data. For this objective, that implementation numerically equates to the expression given in (11). All models are assessed on the same held-out test windows. Having a common evaluation set is essential as varying forecast horizon, target scale, or temporal split would render the RMSE comparisons non-informative. There are two observations from the experiment that are relevant for future work. The GRU beats the deeper LSTM configuration in this case, meaning a lighter recurrent model can remain competitive. The second reason is the R^2 values are relatively modest, indicating that there is still important information that is missing from the seven electrical inputs. Increasing the depth of the network may not be as useful as weather, calendar effects, occupancy proxies, peak-aware objectives and probabilistic forecasting. The complex architectures mentioned in [7], [8], [18]-[23] would be interesting to test too, if the same temporal partitioning and forecasting task is maintained.

6 Conclusion, Limitations and Future Research

This study aims to conduct direct 24-hour household electricity forecasting by using the publicly available UCI Individual Household Electric Power Consumption dataset. Every model has seven electrical variables from the last 72 hours as input and models the Global Active Power for the same duration as the test period. The persistence baseline appears to be less effective in this experiment for recurrent forecasting. GRU gives the lowest error with an RMSE value of 0.6181 kW and an MAE of 0.4687 kW, whereas the standard LSTM gives an RMSE of 0.6262 kW. The

Compact Stacked LSTM Displays 0.6509 Total RMSE in kW It follows then that increasing the depth of the LSTM does not improve the resulting forecasting accuracy in the configuration tested here.

A deeper network will not necessarily perform better than a shallower one, the main finding shows. It is evident that learning the temporal ordering is beneficial, as every recurrent model decreases the error significantly relative to the 24-hour persistence forecast. Yet more complexity does not automatically mean more efficacy. The GRU has a 32.64% decrease in RMSE, standard LSTM a 31.76% decrease, Compact Stacked LSTM a 29.07% decrease against persistence. In cases where the computational cost is also a practical issue. These results support evaluating compact recurrent alternatives before moving to deeper designs.

The results should nevertheless be interpreted within the boundaries of this experiment. The dataset represents one household in France and cannot stand in for feeder-level or city-wide demand. Several potentially important influences, including weather, occupancy, tariffs, holidays, distributed generation, electric-vehicle charging, and socioeconomic differences, are not fully captured. MAPE is also sensitive to low-demand hours, and the modest R^2 values show that a substantial share of the 24-hour variation remains unexplained. The preprocessing introduces another limitation: missing hourly values were interpolated along the continuous time index before the chronological split. Although the original ordering was preserved and standardization relied only on training statistics, a stricter replication should use causal or split-wise imputation so that a later observation cannot influence an earlier missing value. Finally, because hidden sizes, batch sizes, and training durations differ among the models, the comparison is configuration-specific and should not be interpreted as a capacity-matched ablation study.

A natural next step would be to repeat the experiment across multiple households and, where data are available, at feeder level. Weather and calendar information could then be added as external variables, with probabilistic forecasts evaluated alongside point predictions. Longer training and systematic hyperparameter selection may also benefit the recurrent networks, provided that the held-out test period remains untouched. Transformer, N-BEATS, CNN-LSTM, and graph-recurrent approaches are reasonable additional baselines, but a meaningful comparison requires them to use the same temporal split, input information, forecast horizon, and evaluation procedure.

7 References

[1] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," International Journal of Forecasting, vol. 32, no. 3, pp. 914–938, 2016. doi: 10.1016/j.ijforecast.2015.11.011.

[2] H. Habbak, M. Mahmoud, K. Metwally, M. M. Fouda, and M. I. Ibrahem, "Load Forecasting Techniques and Their Applications in Smart Grids," Energies, vol. 16, no. 3, 1480, 2023. doi: 10.3390/en16031480.

[3] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network," IEEE Transactions on Smart Grid, vol. 10, no. 1, pp. 841–851, 2019. doi: 10.1109/TSG.2017.2753802.

[4] M. Alhussein, K. Aurangzeb, and S. I. Haider, "Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting," IEEE Access, vol. 8, pp. 180544–180557, 2020. doi: 10.1109/ACCESS.2020.3028281.

[5] A. Gasparin, S. Lukovic, and C. Alippi, "Deep learning for time series forecasting: The electric load case," CAAI Transactions on Intelligence Technology, vol. 7, no. 1, pp. 1–25, 2022. doi: 10.1049/cit2.12060.

[6] J. Ran, Y. Gong, Y. Hu, and J. Cai, "EV load forecasting using a refined CNN-LSTM-AM," Electric Power Systems Research, vol. 238, 111091, 2025. doi: 10.1016/j.epsr.2024.111091.

[7] S. M. Hasanat, M. Haris, K. Ullah, S. Z. Shah, U. Abid, and Z. Ullah, "Hybrid CNN–LSTM Model With Soft Attention Mechanism for Short-Term Load Forecasting in Smart Grid," Engineering Reports, vol. 7, no. 5, e70163, 2025. doi: 10.1002/eng2.70163.

[8] H. Chen, M. Zhu, X. Hu, J. Wang, Y. Sun, and J. Yang, "Research on short-term load forecasting of new-type power system based on GCN-LSTM considering multiple influencing factors," Energy Reports, vol. 9, pp. 1022–1031, 2023. doi: 10.1016/j.egyr.2023.05.048.

[9] J. Hyeon, H. Lee, B. Ko, and H.-J. Choi, "Deep learning-based household electric energy consumption forecasting," The Journal of Engineering, vol. 2020, no. 13, pp. 639–642, 2020. doi: 10.1049/joe.2019.1219.

[10] Y. Wang, N. Zhang, and X. Chen, "A Short-Term Residential Load Forecasting Model Based on LSTM Recurrent Neural Network Considering Weather Features," Energies, vol. 14, no. 10, 2737, 2021. doi: 10.3390/en14102737.

[11] W. Jiang, "Deep learning based short-term load forecasting incorporating calendar and weather information," Internet Technology Letters, vol. 5, no. 4, e383, 2022. doi: 10.1002/itl2.383.

[12] S. S. Subbiah and J. Chinnappan, "Deep learning based short term load forecasting with hybrid feature selection," Electric Power Systems Research, vol. 210, 108065, 2022. doi: 10.1016/j.epsr.2022.108065.

[13] A. Jahani, K. Zare, and L. Mohammad Khanli, "Short-term load forecasting for microgrid energy management system using hybrid SPM-LSTM," Sustainable Cities and Society, vol. 98, 104775, 2023. doi: 10.1016/j.scs.2023.104775.

[14] E. Choi, S. Cho, and D. K. Kim, "Power Demand Forecasting using Long Short-Term Memory (LSTM) Deep-Learning Model for Monitoring Energy Sustainability," Sustainability, vol. 12, no. 3, 1109, 2020. doi: 10.3390/su12031109.

[15] H. Eskandari, M. Imani, and M. P. Moghaddam, "Convolutional and recurrent neural network based model for short-term load forecasting," Electric Power Systems Research, vol. 195, 107173, 2021. doi: 10.1016/j.epsr.2021.107173.

[16] F. Saeed, A. Paul, and H. Seo, "A Hybrid Channel-Communication-Enabled CNN-LSTM Model for Electricity Load Forecasting," Energies, vol. 15, no. 6, 2263, 2022. doi: 10.3390/en15062263.

[17] S. Zhang, R. Chen, J. Cao, and J. Tan, "A CNN and LSTM-based multi-task learning architecture for short and medium-term electricity load forecasting," Electric Power Systems Research, vol. 222, 109507, 2023. doi: 10.1016/j.epsr.2023.109507.

[18] W. Waheed and Q. Xu, "Data-driven short term load forecasting with deep neural networks: Unlocking insights for sustainable energy management," Electric Power Systems Research, vol. 232, 110376, 2024. doi: 10.1016/j.epsr.2024.110376.

[19] S. A. Nabavi, S. Mohammadi, N. H. Motlagh, S. Tarkoma, and P. Geyer, "Deep learning modeling in electricity load forecasting: Improved accuracy by combining DWT and LSTM," Energy Reports, vol. 12, pp. 2873–2900, 2024. doi: 10.1016/j.egyr.2024.08.070.
[20] P. Nedić, I. Djurović, M. Ćalasan, S. Kovačević, and K. Pavlović, "Electrical energy load forecasting using a hybrid N-BEATS-CNN Approach: Case study Montenegro," Electric Power Systems Research, vol. 247, 111749, 2025. doi: 10.1016/j.epsr.2025.111749.
[21] Y. Ji, A. An, L. Zhang, P. He, and X. Liu, "Short-term load forecasting based on temporal importance analysis and feature extraction," Electric Power Systems Research, vol. 244, 111551, 2025. doi: 10.1016/j.epsr.2025.111551.
[22] K.-D. Lu, B.-X. Zhang, Y. Xu, Y. Shen, and Z.-G. Wu, "MoADL-TLSTM: Multiobjective Automated Deep Learning-Based Transformer-LSTM for Load Forecasting," IEEE Transactions on Neural Networks and Learning Systems, advance online publication, 2026. doi: 10.1109/TNNLS.2026.3700860.
[23] B. Chai, K. Qian, S. Shen, X. Tan, and Y. Hu, "Enhancing long short-term memory for electric load forecasting with multi-batch Bayesian optimization," Electric Power Systems Research, vol. 252, 112436, 2026. doi: 10.1016/j.epsr.2025.112436.
[24] J. S. Caicedo-Vivas and W. Alfonso-Morales, "Short-Term Load Forecasting Using an LSTM Neural Network for a Grid Operator," Energies, vol. 16, no. 23, 7878, 2023. doi: 10.3390/en16237878.
[25] G. Hebrail and A. Berard, "Individual Household Electric Power Consumption," UCI Machine Learning Repository, 2006. doi: 10.24432/C58K54.