International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 6, Nov – Dec 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Energy Consumption Forecasting in Smart Grids Using LSTM: A Review of Methods, Trends and Open Challenges

[1st] **Sahil Raj**
MTech Scholar, Dept. Computer Science and Enginnering
Jai Narain College of Technology, Bhopal, India
Sahilr8742@gmail.com

[2nd] **Dr. Amit Kumar Sahu**
Professor, Dept. Computer Science and Enginnering
Jai Narain College of Technology, Bhopal, India Bhopal, India

[3rd] **Dr. Vivek Sharma**
HOD, Dept. Computer Science and Enginnering
Jai Narain College of Technology, Bhopal, India Bhopal, India

***Abstract*—**
The transition from monthly meter readings to minute-resolution advanced metering infrastructure has made data-driven forecasting of electricity demand practical at the level of the individual premises, where the smoothing effect of aggregation no longer applies. Long Short-Term Memory (LSTM) networks have become the dominant architecture in this setting because their additive cell state preserves gradient information across the twenty-four or more timesteps that a daily cycle spans. This paper reviews the development of short-term load forecasting from regression and autoregressive models through shallow machine learning to recurrent, hybrid and attention-based deep architectures, and examines what the accumulated evidence does and does not establish. Forty-three studies are surveyed and summarised comparatively. The review then presents a controlled implementation on the UCI Individual Household Electric Power Consumption dataset, in which a three-layer stacked LSTM of 137,633 parameters trained under a strictly chronological partition achieved a root mean squared error of 0.4625 kW, a mean absolute error of 0.3147 kW and a coefficient of determination of 0.5992, alongside a mean absolute percentage error of 39.79 per cent. A quantitative analysis demonstrates that this last figure is a property of the skewed demand distribution rather than of forecast quality, since 46.5 per cent of hourly observations fall below 0.5 kW. Four categories of open problem are identified: systematic under-prediction of peak demand under squared-error training, error accumulation in recursive multi-step forecasting, the absence of shared evaluation protocols that renders cross-study comparison unreliable, and the near-universal reporting of point forecasts without uncertainty. Eight directions for further work are proposed, of which probabilistic forecasting and standardised benchmarking are argued to be the most consequential.

***Index Terms*—**Attention mechanisms, deep learning, energy forecasting, load forecasting, Long Short-Term Memory, recurrent neural networks, smart grid, smart meters, time series analysis.

I. INTRODUCTION

A. Overview of the Topic

Electricity differs from most traded commodities in that it must be generated at the instant it is consumed. Grid-scale storage exists but remains expensive and limited relative to demand, so a system operator must know several hours in advance how much power the network will draw, and must know it accurately enough that generation can be committed, reserves sized and interchange scheduled without either shortfall or waste.

For most of the twentieth century this was a tractable problem. Demand at the level of a control area is the sum of hundreds of thousands of independent consumers, and that summation smooths away individual behaviour almost entirely. What remains is a curve driven by weather, calendar and slow economic trend, and a well-tuned regression against temperature and time of day could achieve percentage errors in the low single digits.

The rollout of advanced metering infrastructure has changed both the data available and the questions asked of it. Meters that once turned over monthly now report at fifteen-minute or one-minute intervals, and the resulting archives are large enough that data-driven methods become viable where physical modelling is not [1]. At the same time the operational problems have moved downward in the network hierarchy. Rooftop photovoltaics, domestic battery storage, electric-vehicle charging and time-of-use tariffs all create decisions that must be taken at the level of the individual premises or the local feeder rather than the control area [2], and forecasts of this kind feed directly into microgrid energy management, where predictions of demand and of renewable output are combined to schedule local resources [3].

This downward shift is what makes the problem difficult again. The statistical properties that made aggregate forecasting tractable are precisely the properties that disappear when the aggregation is removed.

B. Importance and Motivation

Consider what a single domestic load curve represents. At any given minute the meter records the sum of whichever appliances happen to be running. A refrigerator contributes a small, almost periodic baseline. Lighting or a television adds a modest step. An electric kettle, an oven or an immersion heater adds a step an order of magnitude larger, for a duration governed by human decisions to which the model has no access.

The resulting series is spiky, in that the ratio between the ninety-ninth percentile and the median is large; intermittent, with long stretches of near-baseline consumption punctuated by short bursts; and only weakly stationary, since the same household behaves differently in February and in July. Kong and colleagues quantified the consequence directly, showing that methods performing well on aggregated residential load degrade sharply when applied to individual customers, and that the degradation is attributable to volatility rather than to any deficiency in the method [4]. Shi, Xu and Li reached the same conclusion from the opposite direction, demonstrating that pooling many households into a shared training set recovers

 ©IJSRED: All Rights are Reserved

much of the lost accuracy precisely because it restores some of the averaging that disaggregation removed [5].

Two further considerations motivate a review at this point. First, the field has moved rapidly toward architectural elaboration — convolutional front-ends, attention layers, decomposition pre-processing — while the reporting of simpler configurations has become correspondingly thinner, so that it is now common to encounter a multi-component hybrid proposed without a fully specified single-architecture baseline against which its components can be judged. Second, percentage-based error metrics are reported almost universally in this literature, yet the interaction between those metrics and the highly skewed demand distribution of an individual household is rarely examined. A model can appear to fail on mean absolute percentage error while producing forecasts whose absolute error is small in the hours that actually matter for grid operation.

C. Objectives of the Review

This paper pursues five objectives.

1) To trace the development of short-term load forecasting from statistical models through shallow machine learning to recurrent and hybrid deep architectures, and to identify what each transition actually contributed.
2) To summarise comparatively the principal studies applying LSTM and related recurrent networks to load forecasting, with attention to the level of aggregation at which each operated.
3) To review current developments, including hybrid convolutional–recurrent stacks, attention mechanisms, and online and adaptive training.
4) To demonstrate, through a controlled implementation on a widely used public dataset, the specific ways in which conventional error metrics can misrepresent forecast quality at the individual-household level.
5) To set out the technical challenges and research gaps that remain, and to propose concrete directions for further work.

Section II reviews the literature and provides a comparative summary. Section III examines current trends and includes an illustrative implementation. Section IV identifies challenges and research gaps, and Section V proposes future directions. Section VI concludes.

Fig. 1 places the methods discussed in this paper on a common timeline, showing the progressive shift from hand-specified temporal features toward architectures that learn temporal structure directly.

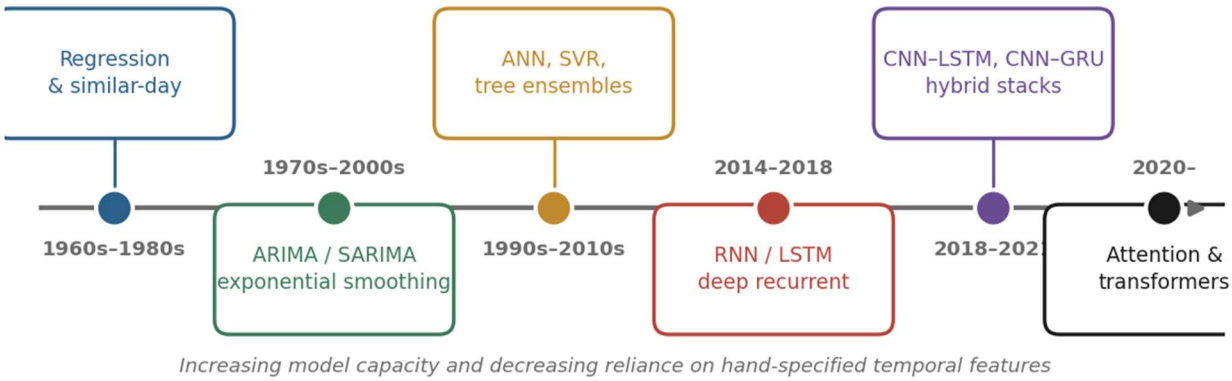

Fig. 1. Development of short-term load forecasting methods. Successive generations reduce reliance on hand-specified temporal features and increase the capacity of the model to learn dependence directly from data.

II. LITERATURE REVIEW

A. Taxonomy of the Forecasting Problem

Load forecasting is conventionally partitioned by horizon. Very short-term forecasting covers minutes to an hour and serves automatic generation control. Short-term forecasting spans an hour to a week and drives unit commitment and economic dispatch; it is by a wide margin the most heavily studied band and is the subject of this review. Medium-term forecasting addresses weeks to a year for maintenance scheduling and fuel contracting, while long-term forecasting supports capacity planning over years. Table I summarises the four bands.

TABLE I. Forecasting horizons and their operational uses

Band	Horizon	Principal operational use
Very short-term	Minutes to 1 hour	Automatic generation control, regulation reserve
Short-term	1 hour to 1 week	Unit commitment, economic dispatch, demand response
Medium-term	1 week to 1 year	Maintenance scheduling, fuel contracting
Long-term	Over 1 year	Capacity expansion and network planning

Hong and Fan provide the standard treatment of this taxonomy and, more usefully, of the distinction between point and probabilistic forecasting [2]. Their argument that operators increasingly require distributions rather than single numbers has aged well and forms part of the motivation for Section V. A second axis of organisation is the level of aggregation: system-level, feeder-level and individual-customer forecasting are different problems sharing a name, and Wang and colleagues note in their survey of smart meter analytics that the individual-customer band remains the least mature of the three [1], a judgement echoed in the comprehensive review of single and hybrid predictive models by Al Mamun and co-workers [6]. Fig. 2 organises the methods reviewed below by model family.

 ©IJSRED: All Rights are Reserved

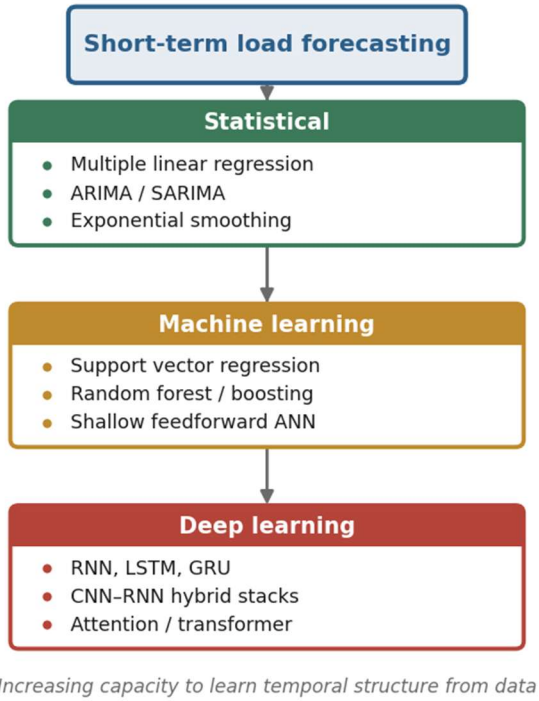

Fig. 2. Taxonomy of short-term load forecasting approaches by model family.

B. Statistical Foundations

The earliest systematic approaches were regression models relating demand to temperature, day type and hour. They remain competitive at aggregate level and are still deployed operationally, largely because they are interpretable, cheap to refit and degrade gracefully; the standard modern treatment of these and the methods that followed is given by Hyndman and Athanasopoulos [7].

The Box–Jenkins family of autoregressive integrated moving average models formalised the treatment of serial dependence, and seasonal variants handle the daily and weekly periodicity of electricity demand explicitly. Their weakness in the individual-household setting is structural rather than incidental. They assume a linear relationship between past and future values, and they assume that the series can be rendered stationary by differencing. A load curve dominated by appliance switching satisfies neither assumption comfortably, and exponential smoothing shares the limitation, extrapolating level and seasonality without any capacity to represent the conditional, threshold-like behaviour that switching produces.

C. Shallow Machine Learning

The 1990s and 2000s saw wide adoption of methods that relaxed the linearity assumption. Multilayer perceptrons, support vector regression and, later, tree ensembles were all applied to load forecasting with reported improvements over regression baselines. Raza and Khosravi survey this period thoroughly and observe a recurring pattern: nonlinear learners reliably outperform linear ones on aggregate data, but the margin narrows as the quality of the feature engineering improves, which suggests that much of the gain came from representation rather than from model capacity [8].

Tree-based ensembles deserve particular mention because they remain a strong and often underrated baseline. Moon and colleagues combined random forests with a multilayer perceptron and found the hybrid competitive with more elaborate architectures [9], while Johannesen and co-workers compared a range of regression tools for urban demand and found gradient boosting difficult to beat once lag features were supplied [10]. This is a useful corrective to the assumption that deep learning is automatically preferable.

The general limitation of shallow learners in this domain is that they consume a fixed-length feature vector. Temporal structure must be encoded by hand through lags and rolling statistics; the model cannot discover it. That is precisely the constraint recurrent architectures were designed to remove.

D. Recurrent Architectures

A recurrent network maintains a hidden state that is updated at every timestep by a single shared transition, so that the parameter count is independent of sequence length and the ordering of the input is intrinsic to the computation. In their basic form, however, such networks are difficult to train across long spans. Bengio, Simard and Frasconi showed that the gradient of an early input with respect to a late output is a product of Jacobians, and that this product tends either to vanish or to explode as the span grows [11]. Pascanu, Mikolov and Bengio later gave a geometric account and proposed gradient clipping, which controls the exploding case but does nothing for the vanishing one [12].

The Long Short-Term Memory architecture of Hochreiter and Schmidhuber was designed to break that product [13]. It introduces a cell state updated by addition rather than by repeated matrix multiplication, together with three multiplicative gates that determine what is discarded, what is admitted and what is exposed. The forget gate, added subsequently by Gers, Schmidhuber and Cummins [14], allows the cell to release information that has ceased to be relevant, which matters for continual prediction on streams that have no natural episode boundaries — exactly the setting of a load series.

The gates are computed from the concatenation of the current input and the previous hidden state, as given in (1) to (3), where σ denotes the logistic sigmoid.

$$f_t = \sigma\left(W_f \bullet [\, h_t^{-1}, x_t \,] + b_f \right) \qquad (1)$$

$$i_t = \sigma\left(W_i \bullet [\, h_t^{-1}, x_t \,] + b_i \right) \qquad (2)$$

 ©IJSRED: All Rights are Reserved

$$o_t = \sigma(W_o \bullet [h_t^{-1}, x_t]\ b_o) \quad (3)$$

A candidate update is produced through a hyperbolic tangent, after which the cell and hidden states follow from (4) to (6), in which $\odot$ denotes elementwise multiplication.

$$\tilde{c}_t = \tanh(W_c \bullet [h_t^{-1}, x_t] + b_c) \quad (4)$$

$$c_t = f_t \odot c_t^{-1} + i_t \odot \tilde{c}_t \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

The significance of (5) lies in its additive form. Differentiating the cell state at time t with respect to the cell state at t−1 yields the forget gate activation directly, as in (7), with no multiplication by a weight matrix and no saturating derivative.

$$\frac{\partial c_t}{\partial c_t\hat{}(-1)} = f_t \quad (7)$$

Propagating across k steps therefore accumulates a product of forget gate values rather than a product of Jacobians. Where the network learns to hold the forget gate near unity for a given cell dimension, information in that dimension is carried forward essentially intact and the gradient flows backward along the same path with little attenuation. Fig. 3 renders the structure.

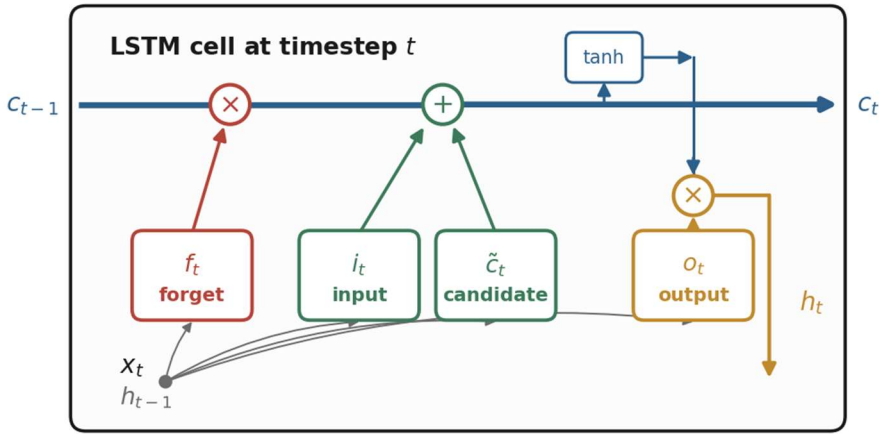

Fig. 3. Internal structure of an LSTM cell, showing the three gates of (1) to (3), the candidate update of (4), and the additive cell-state path of (5).

The gated recurrent unit of Cho and colleagues [15] achieves a comparable effect by merging the forget and input gates and dispensing with the separate cell state, yielding roughly three-quarters of the parameters. Chung and colleagues compared the two empirically and found neither consistently superior across tasks [16], which is why both appear in the literature surveyed below.

E. Deep Learning Applied to Load Forecasting

Mocanu and colleagues were among the first to apply deep architectures to building energy estimation, using restricted Boltzmann machine variants and reporting gains over shallow alternatives [17]. Marino, Amarasinghe and Manic then applied LSTM networks directly to building load data, comparing a standard LSTM against a sequence-to-sequence formulation [18]. Their finding is worth restating because it has shaped subsequent practice: the standard LSTM performed well on hourly data but failed at one-minute resolution, whereas the sequence-to-sequence variant handled both.

That result is a direct argument for temporal aggregation. At one-minute resolution the dependencies a model must span to cover a daily cycle number 1,440 timesteps, which is beyond what gradient descent through a recurrent network reliably learns. Aggregating to hourly means reduces the span to twenty-four.

Zheng and colleagues applied LSTM networks to smart grid load specifically and reported improvements over both seasonal ARIMA and support vector regression baselines [19]. Bouktif and co-workers wrapped a genetic algorithm around feature and lag selection, arguing that the input window is itself a hyperparameter worth optimising rather than fixing by convention [20].

At the individual-customer level the two most directly relevant studies are those of Kong and colleagues [4] and Shi, Xu and Li [5]. Kong's group benchmarked LSTM networks against a broad set of alternatives on residential smart meter data; the headline result was that the LSTM led the comparison, but the more instructive finding was the size of the gap between individual-level and aggregate-level accuracy under identical methodology. A companion paper by the same group incorporated resident behaviour learning as an additional signal and improved matters further, confirming that the residual unpredictability is behavioural rather than purely statistical [21].

Fekri and colleagues extended the recurrent approach to an online setting, updating the network continuously as new readings arrive rather than retraining in batch [22]. Chitalia and co-workers evaluated several recurrent variants on commercial buildings and emphasised robustness across building types rather than peak accuracy on any one [23]. Li and colleagues worked at apartment level with a modified network using selected autoregressive features, again finding that careful lag selection contributed as much as architectural sophistication [24].

F. Comparative Summary

Table II summarises the principal studies reviewed, organised by architecture and by the level of aggregation at which each operated. The final column records the finding most relevant to the present discussion rather than the headline claim.

TABLE II. Comparative summary of representative studies in short-term load forecasting

 ©IJSRED: All Rights are Reserved

Study	Architecture	Aggregation level	Data resolution	Principal finding
Mocanu et al. [17]	Deep belief network / RBM	Building	Hourly	Deep architectures outperform shallow alternatives on building energy estimation
Marino et al. [18]	LSTM and sequence-to-sequence	Building	Minute and hourly	Standard LSTM fails at minute resolution but succeeds at hourly resolution
Zheng et al. [19]	LSTM	Grid	Hourly	Recurrent model improves on seasonal ARIMA and SVR baselines
Kong et al. [4]	LSTM	Individual customer	Half-hourly	LSTM leads the benchmarks, but individual-level error far exceeds aggregate-level error
Shi et al. [5]	Pooled deep RNN	Individual customer	Half-hourly	Pooling customers increases data diversity and suppresses overfitting
Kong et al. [21]	LSTM with behaviour learning	Individual customer	Half-hourly	Residual unpredictability is behavioural rather than purely statistical
Bouktif et al. [20]	LSTM with genetic search	Aggregate	Hourly	Lag and feature selection materially affect accuracy
Moon et al. [9]	Random forest and MLP	Building complex	Hourly	Tree ensembles remain competitive with deeper models
Kim and Cho [25]	CNN–LSTM	Household	Minute to hourly	Convolutional front-end improves on LSTM alone
Alhussein et al. [26]	CNN–LSTM	Individual household	Hourly	Hybrid suppresses spurious peaks produced by recurrent-only models
Sajjad et al. [27]	CNN–GRU	Residential	Hourly	Comparable accuracy at reduced parameter count
Shao and Kim [28]	Multi-channel LSTM	Residential	Hourly	Explicit time–location encoding captures customer behaviour
Sehovac et al. [29]	Sequence-to-sequence with attention	Building	Hourly	Attention improves accuracy over multi-step horizons
Fekri et al. [22]	Online adaptive RNN	Household	Hourly	Continuous adaptation outperforms periodic batch retraining
Chitalia et al. [23]	Deep recurrent variants	Commercial building	Hourly	Robustness across building types matters more than peak accuracy
Li et al. [24]	Modified NN with AR features	Apartment	Hourly	Lag selection contributes as much as architectural change
Gunawan and Huang [30]	Dynamic time warping with LSTM	Aggregate	Hourly	Explicit holiday handling addresses a known recurrent weakness
Sadaei et al. [31]	CNN with fuzzy time series	Aggregate	Hourly	Alternative input representations are viable

III. CURRENT TRENDS AND DEVELOPMENTS

A. Hybrid Convolutional–Recurrent Architectures

Recent work has moved decisively toward combinations of architectures. The dominant pattern places one or more convolutional layers before the recurrent stack, on the reasoning that convolution extracts local shape features from within the input window while the recurrent layers model dependence across it. Fig. 4 shows the canonical arrangement.

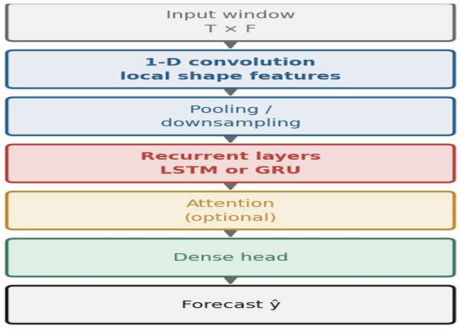

Fig. 4. Canonical hybrid pipeline. Convolutional layers extract local shape features from the input window; recurrent layers model dependence across it; an optional attention layer reweights timesteps before the dense head.

Kim and Cho applied this pattern to residential consumption and reported improvements over an LSTM-only configuration [25]. Alhussein, Aurangzeb and Haider developed a comparable hybrid for individual household forecasting and noted specifically that a recurrent-only model produced spurious peaks at instants where the actual load showed none [26]. Sajjad and colleagues substituted gated recurrent units for LSTM in a similar stack and obtained competitive accuracy with roughly a quarter fewer parameters [27].

Two observations follow from this body of work rather than from any single study. The gains reported for convolutional front-ends are real but consistently incremental; none represents a step change, which suggests that the recurrent component is performing most of the modelling and the convolution is refining rather than transforming the representation. And the direction of the residual error differs between studies, some reporting spurious peaks and others systematic under-response, which points to the loss function and the feature set rather than the architecture as the governing factor.

B. Attention Mechanisms and Transformers

Attention allows a model to weight timesteps in the input window according to their relevance to the current prediction rather than compressing the window into a fixed-size state. Sehovac, Nesen and Grolinger applied a sequence-to-sequence formulation with attention to building energy data and reported improvement that grew with the forecast horizon [29], which is

 ©IJSRED: All Rights are Reserved

the expected pattern: the longer the horizon, the more the fixed-size bottleneck of a plain recurrent encoder costs.

The transformer architecture dispenses with recurrence entirely and relies on self-attention throughout [32]. Its advantages in this domain are parallel training and unrestricted access to any position in the input window; its disadvantage is a data appetite that a single household's few years of hourly readings does not satisfy. The published evidence at individual-household scale remains thin, and the reasonable reading at present is that transformers are compelling where many customers or many years of data can be pooled and less so otherwise.

C. Online, Adaptive and Behaviour-Aware Training

A model trained once on a fixed archive degrades as occupancy, appliance stock and tariff structure change. Fekri and colleagues addressed this with an online adaptive recurrent network updated continuously as readings arrive, and reported that continuous adaptation outperforms periodic batch retraining [22]. Kong and colleagues approached the same underlying problem from the feature side, incorporating explicit resident behaviour learning [21].

A related strand applies metaheuristic search to the training of recurrent forecasters, as in the Jaya-optimised LSTM of Ashfaq and colleagues [33]. Gunawan and Huang addressed the specific and stubborn difficulty of holidays by combining dynamic time warping with an LSTM to identify analogous historical days [30], and Sadaei and colleagues took a different route entirely, converting the load series into fuzzy time series images before applying convolution [31]. What these have in common is that they add information or adaptivity rather than depth, which is consistent with the pattern noted in Section III-A.

D. An Illustrative Implementation

To ground the discussion that follows, this subsection reports a controlled implementation carried out by the author. The purpose is not to propose a new architecture but to provide a fully specified reference point against which the metric behaviour discussed in Section IV can be examined quantitatively.

The dataset is the UCI Individual Household Electric Power Consumption archive compiled by Hébrail and Bérard [34], comprising 2,075,259 minute-resolution measurements recorded in a dwelling near Paris between December 2006 and November 2010. Approximately 1.25 per cent of rows carry missing measurements; these were imputed by forward propagation followed by backward propagation, in preference to deletion, which would have broken the uniform spacing that windowing requires, and to linear interpolation, which would have manufactured smooth ramps in a series whose defining characteristic is that it moves in steps.

The series was resampled to hourly means and extended with twelve derived predictors: calendar fields for hour, day of week, month and weekend status; sine–cosine encodings of hour and month, computed as in (8), which remove the artificial discontinuity between hour 23 and hour 0; lagged demand at one and twenty-four hours; and six-hour rolling mean and standard deviation.

$$x_{sin} = \sin\left(\frac{2\pi k}{K}\right), \quad x_{cos} = \cos\left(\frac{2\pi k}{K}\right) \tag{8}$$

After removing rows with undefined lags, 34,565 hourly observations remained. These were divided at the eightieth percentile of the time index into 27,652 training and 6,913 test hours, without shuffling. This point deserves emphasis. A random split places test observations between training observations, and because consumption is strongly autocorrelated the model can then interpolate rather than forecast, producing reported accuracy that is unattainable in deployment. Separate min–max scalers for predictors and target were fitted on the training partition alone.

The network comprised three stacked LSTM layers of 128, 64 and 32 units with dropout [35] between them, trained with the Adam optimiser [36] against a mean squared error objective. It was implemented with TensorFlow and Keras [37], with scaling and metric computation provided by scikit-learn [38]. Overlapping windows of twenty-four timesteps were then constructed, each input block spanning indices i−T through i−1 and each target the demand at index i, which prevents any contemporaneous measurement from reaching the model. Table III records the resulting configuration.

TABLE III. Configuration of the illustrative implementation

Item	Value
Source dataset	UCI household power consumption [34]
Raw records	2,075,259 at 1-minute resolution
Modelling rows after aggregation	34,565 hourly observations
Predictors	18 (7 measured, 11 engineered)
Input window	24 timesteps
Training / test split	27,652 / 6,913, chronological
Training / test windows	27,628 / 6,889
Architecture	LSTM 128–64–32, dense 16, dense 1
Dropout	0.2 recurrent, 0.1 dense
Trainable parameters	137,633
Optimiser and loss	Adam (α = 0.001), mean squared error
Callbacks	Early stopping (patience 15), LR reduction on plateau
Epochs completed	37; best weights restored from epoch 22

The parameter count follows from the formulation of Section II-D. An LSTM layer of u units receiving d inputs holds four gate computations, each with an input matrix, a recurrent matrix and a bias, giving (9).

$$N = 4u(d + u + 1) \tag{9}$$

For the first layer this yields 75,264, for the second 49,408 and for the third 12,416; with the dense head the total is 137,633.

 ©IJSRED: All Rights are Reserved

Forecast accuracy was assessed with four metrics, defined in (10) to (13) and summarised in Table IV. All were computed after inverse transformation to kilowatts rather than in the scaled domain.

$$RMSE = \sqrt{\left(\frac{1}{n}\right)\Sigma\,(y_i - \hat{y}_i)^2} \tag{10}$$

$$MAE = \left(\frac{1}{n}\right)\Sigma\,|y_i - \hat{y}_i| \tag{11}$$

$$MAPE = (100/n)\,\Sigma\,|(y_i - \hat{y}_i)/y_i| \tag{12}$$

$$R^2 = 1 - \frac{\Sigma(y_i - \hat{y}_i)^2}{\Sigma(y_i - \bar{y})^2} \tag{13}$$

TABLE IV. Properties of the four reported error metrics

Metric	Units	Principal weakness
RMSE	kW	Dominated by a small number of large deviations
MAE	kW	Gives no indication of worst-case behaviour
MAPE	per cent	Grows without bound as the actual value approaches zero
R^2	dimensionless	Depends on the variance of the particular test partition

Table V reports the outcome on the held-out partition.

TABLE V. Accuracy on the held-out test partition (n = 6,889 hours)

Metric	Value	Reading
RMSE	0.4625 kW	Approximately 44 per cent of the mean demand of 1.086 kW
MAE	0.3147 kW	Typical error of roughly 315 W per hour
MAPE	39.79 %	Inflated by low-demand hours; see Section IV-A
R^2	0.5992	About 60 per cent of test-partition variance accounted for

The ratio of RMSE to MAE is 1.47, against approximately 1.25 for normally distributed errors, indicating a heavier-tailed error distribution: the model is usually accurate and occasionally wrong by a large margin. Fig. 5 shows predicted against actual demand over the first two hundred test hours.

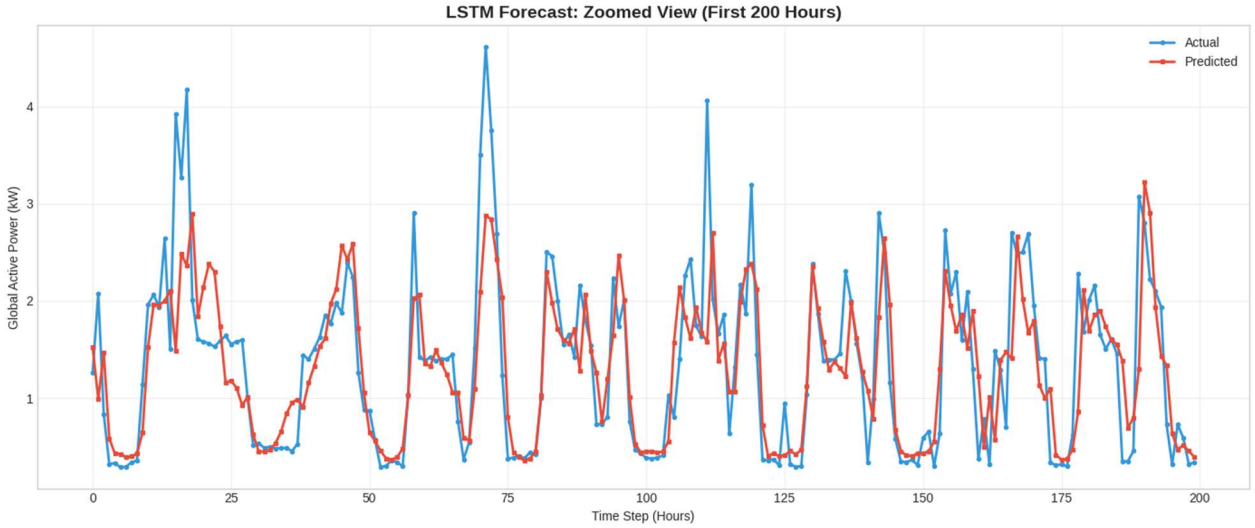

Fig. 5. Actual against predicted demand over the first two hundred hours of the held-out partition. The daily cycle is reproduced accurately in timing and shape, while the amplitude of evening peaks is consistently underestimated.

Two behaviours are visible. The diurnal cycle is reproduced accurately in both timing and general shape, confirming that the calendar and cyclical features are being used. The amplitude of the evening peaks, however, is consistently underestimated: where the actual series reaches four or five kilowatts, the prediction typically reaches three to four. Sharp transitions are also lagged, which is the failure mode the architecture predicts, since nothing in a window of twenty-four prior hours signals that an appliance is about to be switched on.

 ©IJSRED: All Rights are Reserved

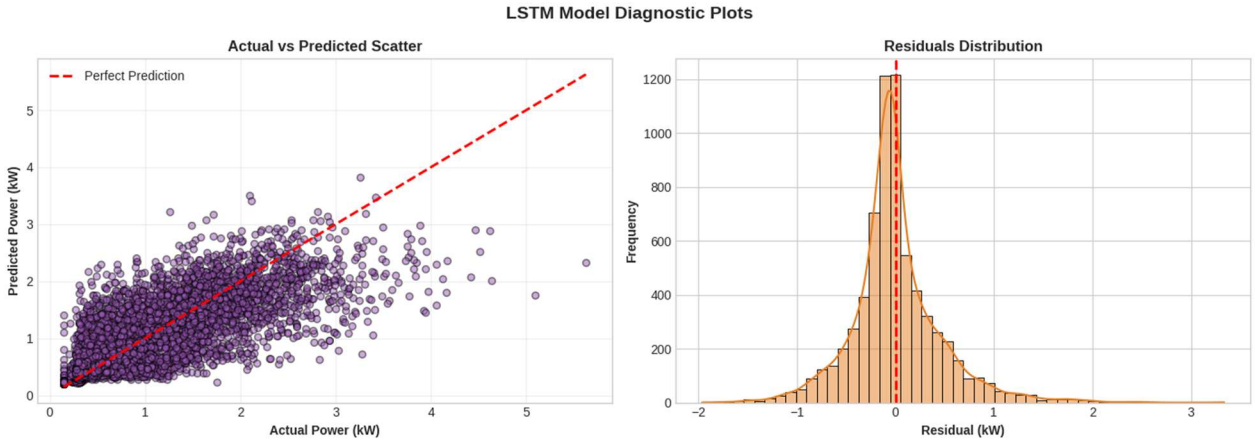

Fig. 6. Predicted against actual demand with the identity line (left) and the distribution of residuals (right). The cloud lies predominantly below the identity line at high demand; the residual distribution is approximately symmetric but markedly leptokurtic.

IV. CHALLENGES AND LIMITATIONS

A. Distortion of Percentage-Based Error Metrics

A mean absolute percentage error approaching forty per cent invites the conclusion that a model has failed. That conclusion is frequently wrong at the individual-household level, and the mechanism is worth setting out because it distorts a substantial part of the reported literature.

Each term of (12) divides an absolute error by the corresponding actual value, so when demand is small the quotient becomes large irrespective of whether the absolute error carries any operational significance. In the dataset examined in Section III-D the minimum hourly demand is 0.076 kW and the median 0.598 kW, and 46.5 per cent of hourly observations fall below 0.5 kW. Fig. 7 plots the percentage error implied by a fixed absolute error across the demand range.

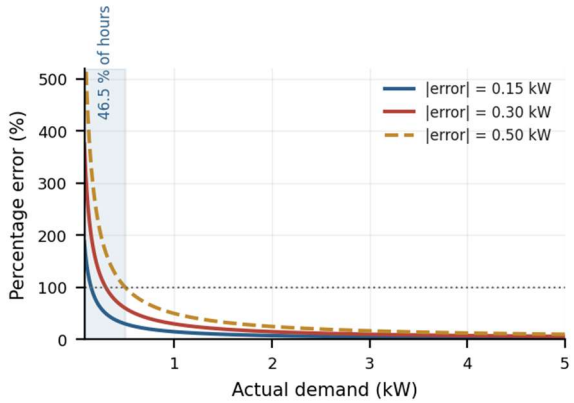

Fig. 7. Percentage error implied by a fixed absolute error as a function of actual demand. The shaded region contains 46.5 per cent of all hourly observations, within which even a modest absolute error registers as a very large relative one.

Consider an hour in which actual demand is 0.10 kW and the model predicts 0.25 kW. The absolute error of 150 W is roughly a household's lighting load and of no operational consequence, yet the percentage error is 150 per cent, contributing as much to the mean as one hundred and fifty hours in error by one per cent. Contrast an hour at 5.0 kW predicted as 4.0 kW: the absolute error is nearly seven times larger and genuinely significant for scheduling, but the percentage error is only 20 per cent. Table VI makes the arithmetic explicit for a fixed absolute error of 0.30 kW, close to the measured mean absolute error.

TABLE VI. Percentage error implied by a fixed 0.30 kW absolute error across demand bands

Demand band	Share of hours	Representative value	Implied error
0.0 – 0.5 kW	46.5 %	0.25 kW	120 %
0.5 – 1.0 kW	21.0 %	0.75 kW	40 %
1.0 – 2.0 kW	20.5 %	1.50 kW	20 %
2.0 – 3.0 kW	8.5 %	2.50 kW	12 %
Above 3.0 kW	3.5 %	4.00 kW	7.5 %

The weighted average of the final column is approximately 63 per cent, for a model whose absolute error is uniformly at its measured average. That the observed value is 39.79 per cent indicates that the model is in fact more accurate than its average in precisely the low-demand hours where the metric is most punishing. The percentage error is therefore substantially a statement about the demand distribution rather than about forecast quality.

B. Systematic Under-Prediction of Peaks

 ©IJSRED: All Rights are Reserved

Fig. 6 shows the predicted-versus-actual cloud lying predominantly below the identity line at high demand, with points above 4 kW on the horizontal axis rarely reaching the same height on the vertical axis. This is not a defect of implementation but a consequence of the objective. Under squared error the loss-minimising prediction is the conditional mean of the target, and when the conditional distribution of the next hour is wide that mean is necessarily less extreme than the realised value.

The consequence is that the model is least reliable in exactly the hours that matter most for distribution-network stress and for demand-response dispatch. The residual distribution in Fig. 6 remains approximately symmetric despite this, because high-demand hours are a small minority whose contribution is confined to the tails.

C. Error Accumulation in Multi-Step Forecasting

Extending a one-step model to a longer horizon is usually done recursively: the model predicts one step, the prediction is appended to the input window, the oldest timestep is discarded, and the process repeats. This is more fragile than it appears. Each prediction is a conditional mean and is therefore smoother than a realised value, so feeding it back makes the next window smoother still, and iterating drives the forecast progressively toward the unconditional mean.

The implementation difficulty is equally real. A recursive loop must write the prediction into the correct feature position, regenerate the lag and rolling-statistic features from the evolving history, advance the calendar features, and re-normalise the row with the correct scaler. In the implementation of Section III-D an indexing error in this loop caused each prediction to be written into an adjacent feature channel, leaving every demand-related feature frozen at its last observed value. The resulting forecast curve remained entirely plausible in shape, which is what makes this class of error dangerous: nothing in the output signals that anything is wrong. Direct multi-step formulations, in which the output layer emits all horizons from a single forward pass, eliminate the feedback mechanism altogether and are preferable where the horizon is known in advance [29].

D. Research Gaps

Beyond these technical issues, four gaps in the literature itself are apparent.

1) Absence of shared evaluation protocols: Studies using the same public dataset differ in temporal resolution, in the target variable predicted, in how partitions are drawn, in input window length and in forecast horizon. Any one of these can move a reported error metric by more than the difference between two architectures. A minority of published work does not state whether the split respects chronology, which matters greatly for the reasons given in Section III-D. Himeur and colleagues argue in their survey of building power consumption datasets that the absence of standard benchmarks is a substantive obstacle to progress [39], and the present review finds no basis on which to rank the studies of Table II numerically.

2) Thin baselines: Proposals of multi-component hybrids frequently omit a fully specified single-architecture comparison, and persistence forecasts are reported still less often. Where lag features are supplied to the model, a substantial share of achievable accuracy is available to a naive predictor, and without a baseline it cannot be established how much the architecture contributes.

3) Point forecasts without uncertainty: The overwhelming majority of studies report point forecasts. For a household-level application the interval is arguably more actionable than the point: knowing that demand will fall between 1 and 6 kW supports a different dispatch decision from a point estimate of 2.5 kW with no indication of its reliability.

4) Under-reporting of negative results: Defects discovered during analysis are typically corrected in silence, and configurations that failed are rarely described. This deprives readers of information about where these pipelines are fragile, and it is one reason the implementation defect described in Section IV-C is documented here rather than omitted.

V. FUTURE RESEARCH DIRECTIONS

Table VII maps each challenge identified in Section IV to the work that would address it. The directions are elaborated below in the order in which they should reasonably be attempted.

TABLE VII. Mapping of identified gaps to proposed research directions

Gap or challenge	Proposed direction	Expected benefit
Thin baselines (IV-D)	Train persistence, SARIMA and gradient boosting under an identical protocol	Isolates the contribution of the recurrent architecture
Metric distortion (IV-A)	Report weighted percentage error and error by demand band	Prevents low-demand hours from dominating the headline figure
Peak under-prediction (IV-B)	Quantile or asymmetric loss functions	Improves accuracy where operational cost is highest
Error accumulation (IV-C)	Direct multi-step output layer or sequence-to-sequence decoding	Removes the feedback loop entirely
Feature attribution unclear	Systematic ablation of calendar, cyclical, lag and rolling groups	Quantifies how much accuracy is learned rather than supplied
No exogenous inputs	Incorporate temperature and a public-holiday calendar	Addresses weather-driven and anomalous-day error
No uncertainty (IV-D)	Pinball loss, Bayesian or ensemble methods	Produces intervals that support dispatch decisions
No shared protocol (IV-D)	Community benchmark with fixed splits and metrics	Makes cross-study comparison meaningful

A. Controlled Baseline Comparison

This is the most urgent gap. A persistence forecaster, a seasonal ARIMA model and a gradient-boosted tree should be trained on identical windows under an identical chronological split and evaluated on the same held-out hours. The evidence that gradient boosting is difficult to beat once lag features are supplied [10] makes the outcome genuinely uncertain, which is precisely why the comparison is worth running.

 ©IJSRED: All Rights are Reserved

B. Distribution-Aware Error Reporting

A weighted percentage error, in which the sum of absolute errors is divided by the sum of actual values, weights each hour by its energy significance rather than inflating trivial hours; it is computed as in (14).

$$wMAPE = 100 \bullet \Sigma | y_i - \hat{y}_i | / \Sigma | y_i | \quad (14)$$

Error should additionally be reported separately by demand band, so that peak-hour performance is visible rather than averaged away.

C. Probabilistic Forecasting

Replacing the point output with quantile regression under a pinball loss, as in (15) for quantile τ, converts the forecast into a distribution at modest additional cost.

$$L_\tau = max(\tau(y - \hat{y}), (\tau - 1)(y - \hat{y})) \quad (15)$$

Hong and Fan argue that this is what operators increasingly require [2], and Quan and colleagues survey the computational-intelligence methods available for quantifying such uncertainty in power system applications [40]. Together with standardised benchmarking, this is the direction with the greatest prospective value.

D. Architectural Comparison Under a Fixed Protocol

A gated recurrent unit stack [15], a convolutional–recurrent hybrid following Kim and Cho [25], and a transformer encoder [32] should each be trained with the data pipeline held constant and only the architecture varied. This is the comparison the literature summarised in Table II largely lacks, and it is the only way to separate architectural contribution from pipeline differences.

E. Multi-Household and Transfer Learning

The pooling strategy of Shi, Xu and Li [5] should be applied more widely, training a shared model across many customers and fine-tuning per household. Given that dataset breadth appears to matter at least as much as architectural sophistication at this level of aggregation, this may yield larger gains than any modelling change. Federated variants, in which model updates rather than raw readings leave the premises, would address the privacy concerns that smart meter data raises [1].

F. Adaptive Deployment

Occupancy, appliance stock and tariffs all change, and a model trained once will drift. The online adaptive approach of Fekri and colleagues [22] deserves wider evaluation, particularly on the question of how quickly a recurrent model can be updated without catastrophic forgetting of seasonal structure it has already learned. The computational budget makes this practical: the network of Section III-D required approximately seventeen minutes of CPU time to train, which places nightly retraining within reach of edge hardware at a substation or gateway.

VI. CONCLUSION

This paper has reviewed the development of short-term load forecasting from regression and autoregressive models through shallow machine learning to recurrent, hybrid and attention-based deep architectures, with particular attention to the individual-household setting in which the smoothing effect of aggregation is absent.

Four conclusions emerge from the surveyed evidence and the illustrative implementation. First, LSTM networks are a sound default for this problem, and the mechanism responsible — the additive cell state of (5), whose derivative in (7) is the forget gate rather than a product of Jacobians — is well understood. Second, the gains reported for convolutional front-ends, attention layers and other elaborations are real but consistently incremental, which suggests that data breadth, feature quality and evaluation protocol now constrain progress more than model capacity does. Third, percentage-based error metrics systematically misrepresent forecast quality at the individual-household level: the analysis of Section IV-A shows that a mean absolute percentage error near forty per cent is compatible with a mean absolute error of roughly 315 W, because 46.5 per cent of hours fall below 0.5 kW. Fourth, squared-error training produces systematic under-prediction of peaks, which places the least reliable forecasts in the hours of greatest operational consequence.

Three recommendations follow for future work in this area. Studies at individual-customer level should report a weighted percentage error and error disaggregated by demand band alongside any conventional metric, and should state explicitly whether their partition respects chronology. Proposals of hybrid architectures should include a fully specified single-architecture baseline and a persistence forecast, so that the contribution of each component can be judged. And the field would benefit substantially from a community benchmark on a public dataset with fixed splits, a fixed horizon and a fixed metric set, without which the comparative summary of Table II cannot be converted into a ranking.

Household-level forecasting will not reach the accuracy attainable at aggregate level, because a substantial part of what it attempts to predict is a sequence of human decisions that leaves no trace in the data until it has already occurred. The productive question is not whether the error can be driven toward zero but how much of the remainder is irreducible, and answering it requires the baselines, ablations and uncertainty estimates set out in Section V.

REFERENCES

[1] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," IEEE Trans. Smart Grid, vol. 10, no. 3, pp. 3125–3148, May 2019, doi: https://doi.org/10.1109/TSG.2018.2818167

[2] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," Int. J. Forecast., vol. 32, no. 3, pp. 914–938, Jul. 2016, doi: https://doi.org/10.1016/j.ijforecast.2015.11.011

[3] M. Alhussein, S. I. Haider, and K. Aurangzeb, "Microgrid-level energy management approach based on short-term forecasting of wind speed and solar irradiance," Energies, vol. 12, no. 8, p. 1487, Apr. 2019, doi: https://doi.org/10.3390/en12081487

[4] 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 Trans. Smart Grid, vol. 10, no. 1, pp. 841–851, Jan. 2019, doi: https://doi.org/10.1109/TSG.2017.2753802

[5] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting — A novel pooling deep RNN," IEEE Trans. Smart Grid, vol. 9, no. 5, pp. 5271–5280, Sep. 2018, doi: https://doi.org/10.1109/TSG.2017.2686012

[6] A. Al Mamun, M. Sohel, N. Mohammad, M. S. H. Sunny, D. R. Dipta, and E. Hossain, "A comprehensive review of the load forecasting techniques using single and hybrid predictive models," IEEE Access, vol. 8, pp. 134911–134939, 2020, doi: https://doi.org/10.1109/ACCESS.2020.3010702

[7] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice, 3rd ed. Melbourne, Australia: OTexts, 2021. [Online]. Available: https://otexts.com/fpp3/

[8] M. Q. Raza and A. Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," Renew. Sustain. Energy Rev., vol. 50, pp. 1352–1372, Oct. 2015, doi: https://doi.org/10.1016/j.rser.2015.04.065

[9] J. Moon, Y. Kim, M. Son, and E. Hwang, "Hybrid short-term load forecasting scheme using random forest and multilayer perceptron," Energies, vol. 11, no. 12, p. 3283, Nov. 2018, doi: https://doi.org/10.3390/en11123283

[10] N. J. Johannesen, M. Kolhe, and M. Goodwin, "Relative evaluation of regression tools for urban area electrical energy demand forecasting," J. Clean. Prod., vol. 218, pp. 555–564, May 2019, doi: https://doi.org/10.1016/j.jclepro.2019.01.108

[11] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE Trans. Neural Netw., vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: https://doi.org/10.1109/72.279181

[12] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in Proc. 30th Int. Conf. Mach. Learn. (ICML),

 ©IJSRED: All Rights are Reserved

Atlanta, GA, USA, 2013, pp. 1310–1318. [Online]. Available: https://arxiv.org/abs/1211.5063

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: https://doi.org/10.1162/neco.1997.9.8.1735

[14] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," Neural Comput., vol. 12, no. 10, pp. 2451–2471, Oct. 2000, doi: https://doi.org/10.1162/089976600300015015

[15] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in Proc. EMNLP, Doha, Qatar, 2014, pp. 1724–1734. [Online]. Available: https://arxiv.org/abs/1406.1078

[16] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv:1412.3555, Dec. 2014. [Online]. Available: https://arxiv.org/abs/1412.3555

[17] E. Mocanu, P. H. Nguyen, M. Gibescu, and W. L. Kling, "Deep learning for estimating building energy consumption," Sustain. Energy Grids Netw., vol. 6, pp. 91–99, Jun. 2016, doi: https://doi.org/10.1016/j.segan.2016.02.005

[18] D. L. Marino, K. Amarasinghe, and M. Manic, "Building energy load forecasting using deep neural networks," in Proc. IECON 2016, Florence, Italy, 2016, pp. 7046–7051, doi: https://doi.org/10.1109/IECON.2016.7793413

[19] J. Zheng, C. Xu, Z. Zhang, and X. Li, "Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network," in Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS), Baltimore, MD, USA, 2017, pp. 1–6, doi: https://doi.org/10.1109/CISS.2017.7926112

[20] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm," Energies, vol. 11, no. 7, p. 1636, Jun. 2018, doi: https://doi.org/10.3390/en11071636

[21] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, "Short-term residential load forecasting based on resident behaviour learning," IEEE Trans. Power Syst., vol. 33, no. 1, pp. 1087–1088, Jan. 2018, doi: https://doi.org/10.1109/TPWRS.2017.2688178

[22] M. N. Fekri, H. Patel, K. Grolinger, and V. Sharma, "Deep learning for load forecasting with smart meter data: Online adaptive recurrent neural network," Appl. Energy, vol. 282, p. 116177, Jan. 2021, doi: https://doi.org/10.1016/j.apenergy.2020.116177

[23] G. Chitalia, M. Pipattanasomporn, V. Garg, and S. Rahman, "Robust short-term electrical load forecasting framework for commercial buildings using deep recurrent neural networks," Appl. Energy, vol. 278, p. 115410, Nov. 2020, doi: https://doi.org/10.1016/j.apenergy.2020.115410

[24] L. Li, C. J. Meinrenken, V. Modi, and P. J. Culligan, "Short-term apartment-level load forecasting using a modified neural network with selected auto-regressive features," Appl. Energy, vol. 287, p. 116509, Apr. 2021, doi: https://doi.org/10.1016/j.apenergy.2021.116509

[25] T.-Y. Kim and S.-B. Cho, "Predicting residential energy consumption using CNN-LSTM neural networks," Energy, vol. 182, pp. 72–81, Sep. 2019, doi: https://doi.org/10.1016/j.energy.2019.05.230

[26] 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: https://doi.org/10.1109/ACCESS.2020.3028281

[27] M. Sajjad, Z. A. Khan, A. Ullah, T. Hussain, W. Ullah, M. Y. Lee, and S. W. Baik, "A novel CNN-GRU-based hybrid approach for short-term residential load forecasting," IEEE Access, vol. 8, pp. 143759–143768, 2020, doi: https://doi.org/10.1109/ACCESS.2020.3009537

[28] X. Shao and C.-S. Kim, "Multi-step short-term power consumption forecasting using multi-channel LSTM with time location considering customer behavior," IEEE Access, vol. 8, pp. 125263–125273, 2020, doi: https://doi.org/10.1109/ACCESS.2020.3007163

[29] L. Sehovac, C. Nesen, and K. Grolinger, "Forecasting building energy consumption with deep learning: A sequence to sequence approach," in Proc. IEEE Int. Congr. Internet of Things (ICIOT), Milan, Italy, 2019, pp. 108–116, doi: https://doi.org/10.1109/ICIOT.2019.00029

[30] J. Gunawan and C.-Y. Huang, "An extensible framework for short-term holiday load forecasting combining dynamic time warping and LSTM network," IEEE Access, vol. 9, pp. 106885–106894, 2021, doi: https://doi.org/10.1109/ACCESS.2021.3099981

[31] H. J. Sadaei, P. C. de Lima e Silva, F. G. Guimarães, and M. H. Lee, "Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series," Energy, vol. 175, pp. 365–377, May 2019, doi: https://doi.org/10.1016/j.energy.2019.03.081

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Proc. Adv. Neural Inf. Process. Syst. (NIPS), Long Beach, CA, USA, 2017, pp. 5998–6008. [Online]. Available: https://arxiv.org/abs/1706.03762

[33] T. Ashfaq et al., "Electricity load and price forecasting using Jaya-long short term memory (JLSTM) in smart grids," Entropy, vol. 22, no. 1, p. 10, Dec. 2019, doi: https://doi.org/10.3390/e22010010

[34] G. Hébrail and A. Bérard, "Individual household electric power consumption," UCI Machine Learning Repository, 2006, doi: https://doi.org/10.24432/C58K54

[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learn. Res., vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: https://jmlr.org/papers/v15/srivastava14a.html

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. 3rd Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, 2015. [Online]. Available: https://arxiv.org/abs/1412.6980

[37] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," arXiv:1603.04467, Mar. 2016. [Online]. Available: https://arxiv.org/abs/1603.04467

[38] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://jmlr.org/papers/v12/pedregosa11a.html

[39] Y. Himeur, A. Alsalemi, F. Bensaali, and A. Amira, "Building power consumption datasets: Survey, taxonomy and future directions," Energy Build., vol. 227, p. 110404, Nov. 2020, doi: https://doi.org/10.1016/j.enbuild.2020.110404

[40] H. Quan, A. Khosravi, D. Yang, and D. Srinivasan, "A survey of computational intelligence techniques for wind power uncertainty quantification in smart grids," IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 11, pp. 4582–4599, Nov. 2020, doi: https://doi.org/10.1109/TNNLS.2019.2956195

 ©IJSRED: All Rights are Reserved