INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 7 • November 2026

PREDICTING HOUSEHOLD ENERGY USE FROM SMART METER TIME SERIES: A STACKED XGBOOST–LSTM MODEL

Shahan Yamin Siddiqui[*1], Muzdalifa Bibi[2], Abdullah Dar[3], Unaiza Rehman[4], Shahzada Atif Naveed[5], Muhammad Farrukh Khan[6], Muhammad Waleed[7]

[*1,7] *Department of Computer Science, NASTP Institute of Information Technology, Lahore 54810, Pakistan*
[2] *Department of Computer Science, Minhaj University, Lahore, Pakistan*
[3] *Global Reporting and Analysis Department, Ibex Global Solution, Lahore, Pakistan*
[4] *Department of Software Engineering, NASTP Institute of Information Technology, Lahore 54810, Pakistan*
[5] *Department of Computer Science, Rashid Latif Khan University, Lahore, Pakistan*
[6] *Department of Artificial Intelligence, NASTP Institute of Information Technology, Lahore 54810, Pakistan*

[*1]drshahan@niit.edu.pk, [2]muzdalifabibi3@gmail.com, [3]abdullah.dar@ibex.co, [4]unaizarehman@niit.edu.pk, [5]shahzada.atif@rlku.edu.pk, [6]dr.farrukhkhan@niit.edu.pk, [7]muhammad.waleed@niit.edu.pk

DOI: https://doi.org/10.5281/zenodo.24131721

International Journal of Advanced Research

Keywords
Smart Meters, Energy Consumption Prediction, Long Short-Term Memory, XGBoost, Stacked Ensemble Learning, Time-Series Forecasting, Smart Grid.

Article History
Received: 05 August 2026
Accepted: 19 October 2026
Published: 11 November 2026

Copyright @Author
Corresponding Author: *
Shahan Yamin Siddiqui

Abstract

Smart meters record household electricity use at a fine temporal resolution, providing the data needed to forecast consumption at the level of the individual dwelling. Accurate short-term forecasts support demand-side management, peak shaving and the integration of distributed renewable generation, yet residential load is noisy, non-stationary and shaped by weather, calendar effects and occupant behaviour, which limits the accuracy of any single learner. This study presents a smart meter-based intelligent model that predicts household appliance energy consumption with a stacked ensemble of heterogeneous learners. The model is organised as a layered architecture covering data publishing, middleware preprocessing, structured storage, model training and prediction. Using 19,735 ten-minute records of appliance energy use, indoor temperature and humidity, and weather from a low-energy house, the readings were enriched with twelve lagged consumption values, rolling statistics, calendar features and outdoor conditions. An eXtreme Gradient Boosting (XGBoost) regressor tuned by grid search, a Long Short-Term Memory (LSTM) network and a Prophet model were trained on the first 80% of the series, and their predictions were combined by a linear meta-learner. On the final 20% of the series, the stacked model achieved a mean absolute error of 15.45 Wh, a root mean squared error (RMSE) of 27.37 Wh, a mean absolute percentage error of 16.11% and a coefficient of determination (R^2) of 0.909, compared with R^2 values of 0.892 for the tuned LSTM, 0.870 for XGBoost and 0.557 for a baseline LSTM; its RMSE was 8.2% lower than that of the tuned LSTM and 16.2% lower than that of XGBoost. The results indicate that combining boosted trees with recurrent networks improves household consumption prediction and can support smart home and smart grid energy management.

INTRODUCTION

Residential buildings account for a large and growing share of electricity demand. Urbanisation, longer indoor occupancy, rising expectations of thermal comfort and the spread of electrical appliances have all pushed household consumption upward, and they have also made it harder to anticipate. Monitoring that is based on repeated measurements over time, historical averages and some assumptions about constant load cannot capture the unpredictable and changing nature of consumption for individual users, and companies relying on this monitoring cannot organize generation or handle load losses or plan new capacity [1], [2].

Smart meters have changed how energy consumption can be monitored. Unlike electromechanical meters which record cumulative total of energy consumed only at the end of billing period, smart meters measure consumption every few minutes to hourly and deliver the information to utility and cloud services [2]. When smart meters are combined with Internet of Things (IoT) devices such as smart plugs, thermostats and occupancy sensors, the same data stream can drive automated appliance scheduling under price-based demand response programmes [3] and can even reveal departures from a resident's normal daily routine [4]. The volume and granularity of these records make household-level forecasting with machine learning feasible at scale [1].

Reliable forecasts are the basis on which most of these services operate. Demand response, time-of-use tariffs and peak shaving all require an estimate of upcoming load, and the integration of rooftop photovoltaic generation and battery storage depends on matching an intermittent supply to a predicted demand [2], [5]. Forecasting the consumption of a single household is, however, considerably more difficult than forecasting aggregate system load. Individual profiles are volatile and are driven by occupant behaviour, weather, day-of-week and seasonal effects, and by appliance-specific usage that averages out only when many homes are combined [6], [7]. Smart meter data also arrive with gaps, spikes and measurement errors caused by communication failures, sensor faults and outages, and these defects propagate into any model trained on them [8], [9].

Existing approaches fall broadly into three families. Statistical time-series models such as the autoregressive integrated moving average (ARIMA) are transparent and inexpensive, but they assume linear dynamics and stationarity after differencing, and they cope poorly with abrupt changes and with the influence of exogenous variables [2]. Classical machine learning regressors, including support vector machines (SVM), k-nearest neighbours, random forests and gradient-boosted trees, handle non-linear relationships among engineered features and have been applied extensively to smart meter and building data [10], [7], [11]. Deep learning models, in particular recurrent architectures such as Long Short-Term Memory (LSTM) networks and more recent designs such as N-BEATS, learn temporal dependencies directly from the consumption sequence [6], [1].

Each family has a characteristic weakness. Tree ensembles such as eXtreme Gradient Boosting (XGBoost) model feature interactions well but see the series only through the lag and calendar variables supplied by the analyst, so long-range temporal structure is easily lost. Recurrent networks capture that structure but make less effective use of heterogeneous tabular inputs such as temperature, weather condition and holiday indicators, and they require long, consistent training histories. Hybrid and ensemble schemes have been proposed to combine the strengths of different learners, for example gradient boosting with LSTM networks and Shapley-value explanations [12], or support vector regression tuned by a firefly algorithm and coupled with an adaptive neuro-fuzzy inference system [13]. These studies report gains over single models, but they are often computationally heavy, are evaluated on a single dataset, or describe data preparation only briefly, even though data quality strongly conditions forecasting accuracy [8], [5].

Three gaps therefore remain. First, few residential forecasting studies combine learners of different

kinds, such as boosted trees, recurrent networks and additive time-series models, so that each contributes the type of structure it captures best [12], [13]. Second, the steps that turn raw meter readings into model-ready data, including lagged, rolling, calendar and weather features, are rarely specified explicitly and reproducibly [9], [14]. Third, proposed combinations are seldom compared with their own components under a single experimental protocol, which makes it difficult to tell whether a reported gain comes from the combination itself [5].

The objectives of this work are to analyse and preprocess smart meter data for machine learning, to develop a forecasting model that exploits both temporal and contextual information, and to compare its accuracy with that of its constituent learners. To this end, we propose a smart meter-based intelligent model organised as a layered architecture that carries the data from acquisition, through middleware preprocessing and structured storage, to model training and prediction. The model is evaluated on the publicly available Appliances Energy Prediction dataset, which records the appliance energy use of a low-energy house every ten minutes together with indoor temperature and humidity and outdoor weather [15]. An XGBoost regressor [16], an LSTM network [17] and a Prophet model [18] are trained on the earlier part of the series, and their predictions are combined by a linear meta-learner following the principle of stacked generalisation [19]. The stacked model attains an R^2 of 0.909 and an RMSE of 27.37 Wh, improving on the best single learner, the tuned LSTM, by 8.2% in RMSE.

The latter part of this paper is divided as follows. The Related Work section discusses previous work in the area of energy consumption prediction using smart meters. The Proposed Methodology section is dedicated to the architecture of the system, the steps of preprocessing and feature engineering, as well as the machine learning models. The Experimental Setup section refers to the dataset involved in the research, the procedure of training and evaluation. The Results and Discussion section presents the results obtained by the models, compares them to the results described in other publications, and addresses the limitations of the conducted research, while the final section of the paper summarizes the findings of the paper and indicates future research directions.

The major contributions of this work are as follows:

- A layered smart meter-based forecasting architecture, comprising data publishing, middleware, data and data-seeking modules, that carries raw household meter readings through to consumption prediction and visualisation.
- A feature engineering procedure that augments ten-minute meter readings with twelve lagged consumption values, rolling statistics, calendar features and outdoor weather variables.
- A stacked ensemble in which the predictions of a grid-search-tuned XGBoost regressor, an LSTM network and a Prophet model are combined by a linear meta-learner.
- A comparison of the stacked model with its constituent learners and a baseline LSTM under the same chronological 80:20 protocol and five error metrics, in which the stacked model lowers RMSE by 8.2%, 16.2% and 54.6% relative to the tuned LSTM, XGBoost and the baseline LSTM, respectively.
- A comparison with previously published studies, including the original benchmark on the same dataset, together with a discussion of the limitations that must be addressed before deployment.

Related Work

The forecasting of loads achieved through smart metering data has been analyzed from different points of view. Dewangan and others [2] have reviewed smart meter data based forecasting methodologies and categorized them into statistical, machine learning, and hybrid techniques, emphasizing that the best methodology depends on the forecast horizon and data aggregation. Rashid et al. [5] have presented a broader view of machine learning applications in smart grids including the concepts of supervised, unsupervised, and ensemble methods for demand

forecasts and predictive maintenance, identifying bagging and boosting as well-working approaches for heterogeneous data inputs, for achieving better accuracy. In both reviews, the importance of quality of input data and treatment of exogenous input variables has been recognized as important as the learning algorithm.

Classical supervised learners are the most researched group. Reddy et al. [11] conducted a comparison of linear regression, KNN, XGBoost, random forest and neural networks based on a year of hourly consumption data from power company and found that the greatest performance of 90.92% happened for KNN. Gumz et al. [7] analyzed the data obtained in three-year timeframe from 470 smart meters and concluded that combination of household data and past consumption enhances forecasting, with best results shown by SVM, random forest regression and neural networks. Liu et al. [10] employed SVM for predicting and diagnosing energy consumption in public buildings in China and found that the model helps not only forecast energy consumption, but also detect any abnormalities. These studies are informative baselines; however, they treat consumption as a regression problem, meaning that temporal characteristics enter the model via lag features, little information is presented about hyperparameter optimization, and the evaluation of the results was carried out using one dataset only.

A second line of work uses smart meter data to understand consumers rather than to forecast their demand. Emrouznejad et al. [14] applied machine learning, including clustering of household energy profiles combined with physical, demographic and socio-economic attributes, to derive targeted energy-efficiency implications for residents, and found marked differences between high- and low-consumption groups; the usefulness of such segmentation, however, depends on the number of clusters chosen and on the behavioural variables available. Interpretability has also received attention. Amiri et al. [20] used explainable machine learning to expose the factors behind household transportation energy use, and Janjua et al. [12] attached Shapley additive explanations to a gradient boosting and LSTM forecaster so that the contribution of each input to a household prediction could be inspected. Explanations of this kind increase confidence in a model, although they do not in themselves improve its accuracy.

Deep sequence models address the temporal limitations of tabular learners directly. Shaikh et al. [6] applied neural basis expansion analysis for interpretable time series (N-BEATS) to consumption data from 169 customers and reported that it outperformed LSTM, blocked LSTM, gated recurrent unit and temporal convolutional network baselines, with the best results obtained for day-ahead forecasts that used the day of the week as a covariate. Abdulla et al. [1] trained LSTM forecasters locally at each household and aggregated them through federated learning, adding an adaptive mechanism to respond to drift in the data; compared with centralised training, the approach lowered forecast error by about 8% and reduced training time by roughly 80% while keeping raw readings on site. Such results confirm that recurrent and related architectures capture daily and weekly patterns well. They also reveal the costs: deep models need long and consistent training histories, exogenous variables such as weather are integrated less naturally than in tree-based learners, and the advantage they offer depends on how clean the available history is.

Hybrid and ensemble models attempt to combine complementary learners. Janjua et al. [12] integrated gradient boosting with LSTM networks for smart grid electricity prediction, extracting correlated features before training, and reported improved accuracy over single models. Souhe et al. [13] built a hybrid of support vector regression, the firefly algorithm and an adaptive neuro-fuzzy inference system on 24 years of consumption data from Cameroon and obtained lower RMSE, MAE and MAPE than LSTM and random forest benchmarks, at the price of a complex architecture and demanding training. The bagging and boosting ensembles reviewed in [5] improve stability, but they are usually built from learners of

the same kind and therefore add little temporal modelling capacity. Across this group, the hybrid is frequently tuned for one dataset, and comparisons against its individual constituents under the same data split are not always reported. Data quality and data fusion form the final strand. According to Dumitru et al. [8], energy contour forecasting is carried out using smart metering data, where errors are around 1% in typical instances but rise to about 15% during exceptional cases, especially if there are bad or incomplete readings supplied by the meters. Xiao et al. [9], however, performed an industrial study of a fusion technique combining a modified stacked denoising autoencoder and GANs to eliminate noise and limit overfitting leading to better power consumption predictions. In sensor networks, in general, Cao et al. [21] described the use of a bat algorithm-based extreme learning machine to fuse data coming from different nodes and limit excessive transmissions. In addition, according to Hernández et al. [4], the data can be used to identify anomalies in people's activities typical for them and, thus, create the phenomenon of prediction error itself. Overall, the studies prove that fusion or preprocessing need the same attention as forecasting methods.

In this sense, literature reveals one important gap. XGBoost and SVM, for instance, are examples of tabular learners that can utilize engineered contextual features, but they model the time aspect indirectly [11], [7]. On the other hand, recurrent and basis-function models can successfully model time, but their performance with respect to heterogeneous exogenous variables and noisy data is far from being satisfactory [6] [1]. Existing hybrids are either too complex and require separate optimization for every dataset in question [13] or do not provide any systematic comparison with the elements of their own approach [12], [8], [9]. What is needed is a household forecasting model that combines boosted trees and recurrent networks through stacking, is built on an explicit feature engineering procedure, and is evaluated against each of its components under one protocol on a public dataset. The model proposed in this paper is designed to meet these requirements.

Proposed Methodology

The proposed smart meter-based intelligent model is a modular forecasting system that transforms household meter readings into consumption predictions. Its architecture, shown in Figure 1, consists of four layers that correspond to the stages of data collection, preprocessing, training and testing, and prediction and visualisation. Each layer has a single responsibility, so that the system can be extended with new data sources or learners without changing the remaining layers. The learning layer is designed to host several forecasting algorithms; in the experiments reported here it contains an XGBoost regressor, an LSTM network and a Prophet model whose outputs are combined by stacking.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH
International Peer-Reviewed Research Journal
ISSN (Online): 3106-7980 | ISSN (Print): 3106-7972
Volume 3 • Issue 7 • November 2026

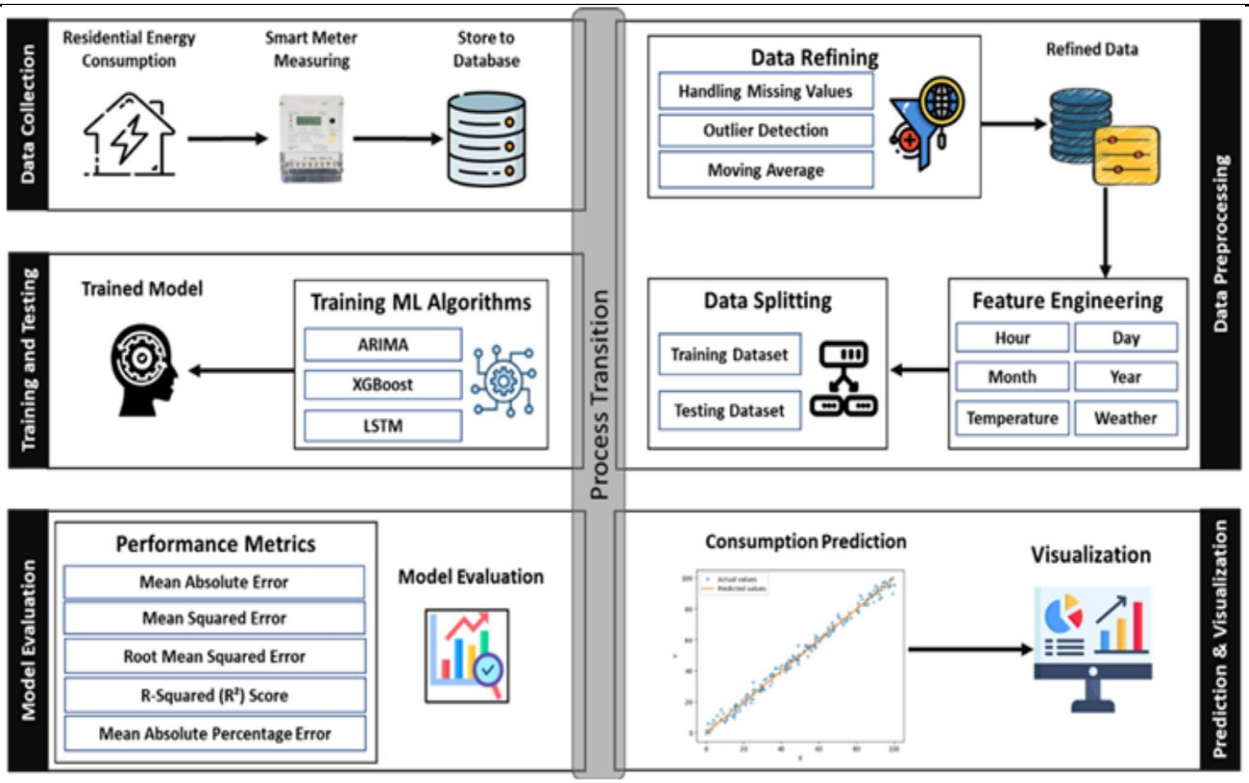


Figure 1: Proposed Smart Meter-Based Energy Consumption Prediction Model

A. Data Publishing Module

The data publishing module corresponds to data collection. Smart meters and IoT sensors record appliance energy use at fixed intervals, together with environmental measurements, and transmit the timestamped readings to a central repository. This fine temporal granularity is needed to reveal short-term consumption patterns and the time-based dependencies on which accurate prediction relies, and errors introduced at this stage propagate through the rest of the pipeline.

B. Middleware Layer

The middleware layer converts raw readings into model-ready features. Records are indexed by timestamp and sorted chronologically, and the consumption series is enriched with three groups of engineered features. Lagged values of consumption at the twelve preceding ten-minute steps capture short-term autocorrelation over the previous two hours. The rolling statistics consist of the average and standard deviation of usage over a seven-step period, which serve as an indicator of recent usage and its local volatility. The time-based parameters that are used are time of day, day of week, and month, which capture the daily, weekly, and seasonal cycles. The temperature and humidity are also incorporated in the analysis as external drivers of usage. All missing data from the lag and rolling-window exercises are removed.

C. Data Layer

In the data layer, the engineered dataset is held in a structured format consisting of the input features and the target energy consumption value in watt-hours for the appliances. The training and testing datasets are separate from each other, and the entire history is stored so that the models can be retrained and back-tested using new data to ensure reproducibility of experiments and version of model improvements.

D. Data-Seeking Module and Learning Algorithms

The data-gathering component of the algorithm instructs the prediction models and measures their

effectiveness. The algorithm employs three different models possessing additional advantages. XGBoost generates an additive mix of regression trees whereby each tree is fitted on the residuals of the current mix, thereby reducing the value of the regulated objective function [16]

$$Obj = \Sigma_i\, l(y_i, \hat{y}_i) + \Sigma_k\, \Omega(f_k), \quad \Omega(f) = \gamma T + \tfrac{1}{2}\,\lambda\, \Sigma_j\, w_j^2$$

where l is the squared-error loss, f_k is the k-th tree, T is the number of leaves and w_j are the leaf weights. Its regularisation and shrinkage limit overfitting, and it captures non-linear interactions among lagged, calendar and weather features.

The LSTM network is a recurrent architecture whose memory cell is controlled by input, forget and output gates [17]. At each step t, the cell state c_t and hidden state h_t are updated as

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \quad h_t = o_t \odot \tanh(c_t)$$

where f_t, i_t and o_t are the forget, input and output gates and $\tilde{c}_t$ is the candidate state. The gating mechanism allows the network to retain relevant information and discard irrelevant inputs, which makes it suitable for consumption series that depend on recurring human activity.

Prophet decomposes a time series into trend, seasonal and holiday components, $y(t) = g(t) + s(t) + h(t) + \varepsilon_t$, and fits them as an additive regression model [18]. It is included as a third, structurally different base learner that models consumption directly as a function of time.

E. Stacked Ensemble

The predictions of the three base learners are combined by stacked generalisation [19]. A linear regression meta-learner receives the base predictions as inputs and estimates the final consumption as

$$\hat{y}_stack = \beta_0 + \beta_1\, \hat{y}_XGB + \beta_2\, \hat{y}_LSTM + \beta_3\, \hat{y}_Prophet$$

where the coefficients β_k weight each base learner according to its contribution to reducing the squared error. Because the meta-learner assigns weights on the basis of the observed performance of the base learners, a learner whose predictions are uninformative receives little weight, while complementary learners are combined to reduce both bias and variance.

Experimental Setup

A. Dataset

The experiments use the Appliances Energy Prediction dataset [15], which contains 19,735 records sampled every ten minutes over about four and a half months in a low-energy house in Belgium. The target is the energy consumed by appliances in watt-hours. The predictors comprise the energy used by lighting fixtures, temperature and humidity in nine indoor locations measured by a wireless sensor network, and outdoor temperature, pressure, humidity, wind speed, visibility and dew point from a nearby weather station. After the lag and rolling-window features were created and incomplete rows removed, 19,717 records remained.

B. Training Protocol and Implementation

The series was split chronologically, with the first 80% of the records (15,773) used for training and the final 20% (3,944) for testing, so that every test observation lies after the training period. XGBoost was tuned by grid search with three-fold cross-validation over the learning rate (0.01, 0.05, 0.1), maximum depth (3, 5, 7), number of trees (50, 100, 200), row subsampling (0.7, 0.8, 0.9) and column subsampling (0.7, 0.8, 0.9). The LSTM received the min–max-scaled feature vector of each time step and consisted of one LSTM layer with 100 units, dropout of 0.3 and a dense output unit, trained with the Adam optimiser and a mean squared error loss for 50 epochs with a batch size of 32. A baseline LSTM with 50 units, dropout of 0.2 and 20 training epochs was also trained to quantify the effect of tuning. Prophet was fitted to the consumption series of the training period. The models were implemented in Python with Pandas, scikit-learn, XGBoost, Keras and Prophet.

C. Evaluation Metrics

Prediction accuracy was measured with five standard metrics. For n test samples with observed values y_i, predictions $\hat{y}_i$ and mean observed value $\bar{y}$:

$$MAE = (1/n)\, \Sigma_i\, |y_i - \hat{y}_i|$$

$$MSE = (1/n)\, \Sigma_i\, (y_i - \hat{y}_i)^2, \quad RMSE = \sqrt{MSE}$$

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

$$R^2 = 1 - \Sigma_i (y_i - \hat{y}_i)^2 / \Sigma_i (y_i - \bar{y})^2$$

MAE and RMSE are expressed in watt-hours; RMSE penalises large errors such as missed consumption peaks more heavily, MAPE expresses the error relative to the observed load, and R^2 gives the proportion of the variance in consumption explained by the model.

Results and Discussion

Table 1 reports the performance of all models on the test period, and Figure 2 compares them visually. The stacked model achieved the best result on every metric, with an MAE of 15.45 Wh, an RMSE of 27.37 Wh, a MAPE of 16.11% and an R^2 of 0.909.

Table 1: Performance of the Models on the Test Period

Model	MAE (Wh)	MSE	RMSE (Wh)	MAPE (%)	R^2
Baseline LSTM	26.92	3636.88	60.31	24.19	0.557
XGBoost (tuned)	17.38	1067.91	32.68	18.40	0.870
LSTM (tuned)	16.42	889.64	29.83	17.82	0.892
Stacked model	15.45	749.33	27.37	16.11	0.909

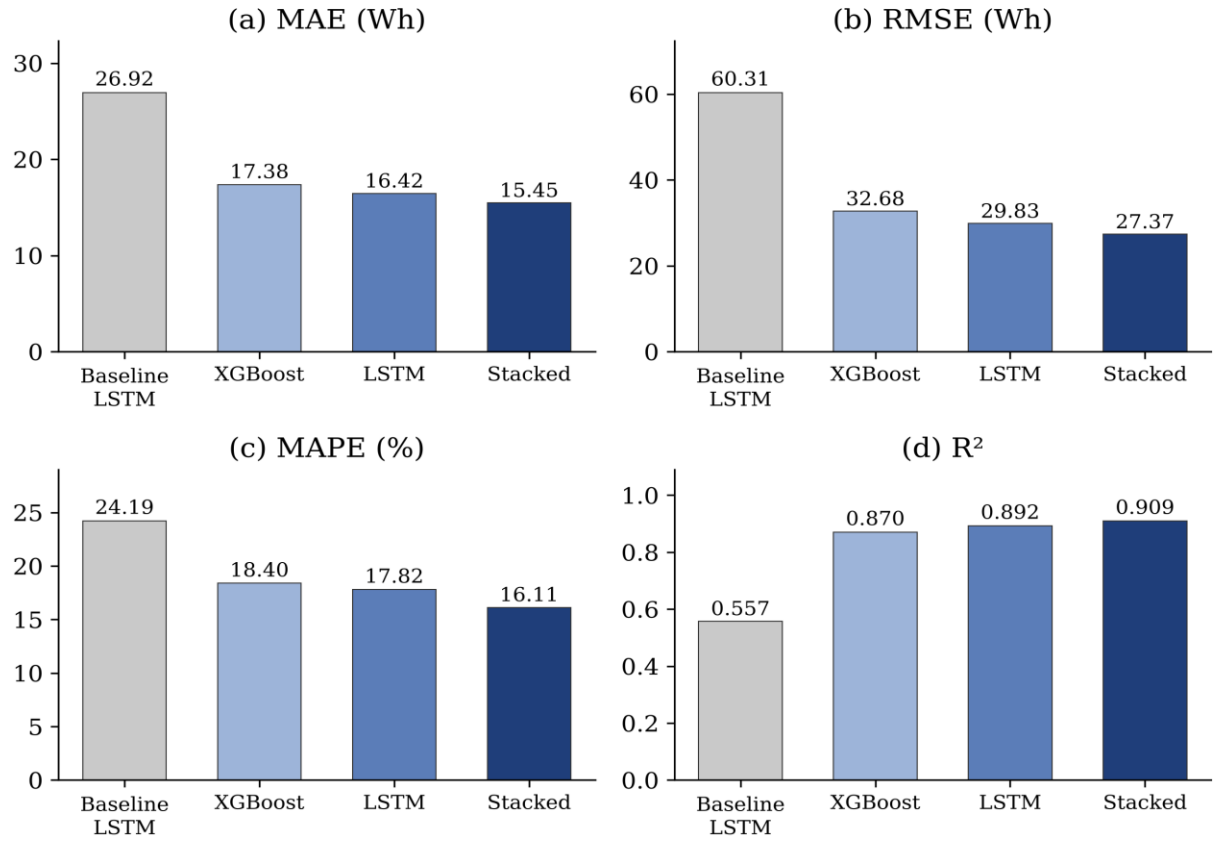


Figure 2: MAE, RMSE, MAPE and R^2 of the Models on the Test Period

Among the single learners, the tuned LSTM performed best, with an RMSE of 29.83 Wh and an R^2 of 0.892, followed by XGBoost with an RMSE of 32.68 Wh and an R^2 of 0.870. The two learners reached similar MAPE values (17.82% and 18.40%), but the lower RMSE of the LSTM indicates that it tracked consumption peaks more closely, which is consistent with its ability to retain information about recent activity through its gated memory [17]. XGBoost gained from grid search when tuned as opposed with untuned model configuration which resulted in RMSE of 34.30 Wh and R^2 of 0.857 and using tuning reduced RMSE by 4.7%. Prophet that predicts consumption in the form of a continuous function could not produce useful predictions when applied to the series due to the negative value of R^2 because the series is based on short irregularities instead of a uniform trend which is obtained with no valid results.

For instance, the standard LSTM model had an R^2 equal to 0.557 and RMSE of 60.31 Wh. The results changed for the better when the number of units in the model was increased to 100, dropout was raised from 0.20 to 0.30, and the number of the epochs was increased to 50. The final results demonstrated less RMSE 29.83 indicating the importance of the configuration of the recurrent neural networks during its use for predicting performance compared to other models.

Utilizing the base estimators led to an additional and stable enhancement. In comparison to the optimized LSTM, the stacked model has decreased MAE by 5.9%, RMSE by 8.2%, MAPE from 17.82% to 16.11%, and has increased R^2 from 0.892 to 0.909; the comparison to the XGBoost shows a 16.2% reduction of RMSE. These improvements suggest that errors of boosted trees are related to the errors of the recurrent network at least partially, which allows weighted averaging to lessen part of their errors. The MSE of the stacked model (749.33) is lower by 15.8% than the one of LSTM, which means that the combination is particularly effective in dealing with large errors caused by consumption spikes.

Table 2 places these results in the context of previously published studies. The comparison is indicative, because the studies differ in their datasets, forecasting horizons and inputs.

Table 2: Comparison with Previously Published Studies

Study	Dataset	Method	Reported Result
Candanedo et al. [15]	Appliances Energy Prediction (same dataset)	GBM, RF, SVM, linear regression; no lagged consumption	GBM: test R^2 0.57
Reddy et al. [11]	One year of hourly utility data	LR, KNN, XGBoost, RF, ANN	KNN: accuracy 90.92%
Shaikh et al. [6]	169 customers	N-BEATS vs. LSTM, GRU, TCN	N-BEATS best; day-ahead horizon
Abdulla et al. [1]	Household smart meters	Federated LSTM with adaptive learning	8% lower error than centralised training
This study	Appliances Energy Prediction	Stacked XGBoost, LSTM and Prophet	R^2 0.909, RMSE 27.37 Wh, MAPE 16.11%

The same dataset was used by Candanedo et al. (15), who reported a test R^2 of 0.57 when performing predictive modelling using sensors and weather variables only. The significantly improved R^2 values obtained in this study can be attributed mainly to the application of lagged consumption values to turn the exercise into a one-step-ahead forecasting problem, where the

latest load is available for the prediction process, and to using different but complementary learners together. The studies in Table 2 are based on other datasets and forecasting time horizons; thus, the estimated values cannot be compared directly, but they nevertheless demonstrate that recurrent and ensemble models tend to outperform traditional single classical approaches in the analysis of smart meter data [6], [1], [11].

Various limitations impose qualifications on these findings. First, the assessment is done on one household across four-and-a-half months; therefore, the generalization of findings to other dwellings, climates, and seasons has to be demonstrated. Second, the rolled mean and standard deviation were computed over a window including the current ten-minute observation, thus allowing the response variable to enter two of the input features-they must have free input variables to actually call their reported accuracy a forecast. Third, the coefficients of the linear meta-learner were estimated on the predictions for the test period, so the stacked result is an in-sample combination estimate; fitting the meta-learner on a separate validation period would give an unbiased out-of-sample measure. Finally, the LSTM received one time step at a time and therefore relied on the engineered lag features rather than on learned sequence memory, and the XGBoost grid search used non-temporal cross-validation folds. Addressing these points with a time-ordered validation design is the main priority for confirming the results.

Conclusion and Future Work

This study presented a smart meter-based intelligent model for predicting household energy consumption. The model organises meter readings into a layered architecture of data publishing, middleware preprocessing, structured storage, model training and prediction, enriches the readings with lagged, rolling, calendar and weather features, and combines an XGBoost regressor, an LSTM network and a Prophet model through a linear meta-learner.

On ten-minute appliance consumption data from a low-energy house, the stacked model achieved an MAE of 15.45 Wh, an RMSE of 27.37 Wh, a MAPE of 16.11% and an R^2 of 0.909, outperforming the tuned LSTM (R^2 0.892), the tuned XGBoost regressor (R^2 0.870) and a baseline LSTM (R^2 0.557). The results show that combining boosted trees with recurrent networks yields more accurate household consumption predictions than either learner alone, and that adequate configuration of the recurrent network is essential.

Further research will validate and build on these results. The pipeline will be revisited by incorporating strictly lagged rolling features, a meta-learner that learns from a different validation period, and time-ordered cross-validation, in addition to implementing multi-step forecasting for day-ahead prediction. The application of sequence-dependent inputs for the LSTM, Transformer and graph neural network structures will be investigated, as well as testing the model using multiple households and different seasons. Non-intrusive load monitoring for appliance level disaggregation, occupancy data and rooftop solar will be combined to facilitate demand response, and edge computing with adaptive retraining will enable dynamic tracking of consumption patterns.

REFERENCES

[1] N. Abdulla, M. Demirci, and S. Özdemir, "Smart meter-based energy consumption forecasting for smart cities using adaptive federated learning," Sustainable Energy, Grids and Networks, vol. 38, art. 101342, 2024.

[2] F. Dewangan, A. Y. Abdelaziz, and M. Biswal, "Load forecasting models in smart grid using smart meter information: A review," Energies, vol. 16, no. 3, art. 1404, 2023.

[3] G. Hafeez, Z. Wadud, I. U. Khan, I. Khan, Z. Shafiq, M. Usman, and M. U. A. Khan, "Efficient energy management of IoT-enabled smart homes under price-based demand response program in smart grid," Sensors, vol. 20, no. 11, art. 3155, 2020.

[4] Á. Hernández, R. Nieto, L. de Diego-Otón, M. C. Pérez-Rubio, J. M. Villadangos-Carrizo, D. Pizarro, and J. Ureña, "Detection of anomalies in daily activities using data from smart meters," Sensors, vol. 24, no. 2, art. 515, 2024.

[5] A. Rashid, P. Biswas, A. A. Masum, M. A. A. Nasim, and K. D. Gupta, "Power plays: Unleashing machine learning magic in smart grids," arXiv preprint arXiv:2410.15423, 2024.

[6] A. K. Shaikh, A. Nazir, I. Khan, and A. S. Shah, "Short term energy consumption forecasting using neural basis expansion analysis for interpretable time series," Scientific Reports, vol. 12, 2022, doi: 10.1038/s41598-022-26499-y.

[7] J. Gumz, D. C. Fettermann, E. M. Frazzon, and M. Kück, "Using Industry 4.0's big data and IoT to perform feature-based and past data-based energy consumption predictions," Sustainability, vol. 14, no. 20, art. 13642, 2022.

[8] C.-D. Dumitru, A. Gligor, I. Vlasa, A. Simo, and S. Dzitac, "Energy contour forecasting optimization with smart metering in distribution power networks," Sensors, vol. 23, no. 3, art. 1490, 2023.

[9] X. Xiao, Y. Xiao, Y. Zhang, J. Qiu, J. Zhang, and T. Yildirim, "A fusion data preprocessing method and its application in complex industrial power consumption prediction," Mechatronics, vol. 77, art. 102520, 2021.

[10] Y. Liu, H. Chen, L. Zhang, X. Wu, and X.-J. Wang, "Energy consumption prediction and diagnosis of public buildings based on support vector machine learning: A case study in China," Journal of Cleaner Production, vol. 272, art. 122542, 2020.

[11] G. V. Reddy, L. J. Aitha, Ch. Poojitha, A. N. Shreya, D. K. Reddy, and G. S. Meghana, "Electricity consumption prediction using machine learning," E3S Web of Conferences, vol. 391, art. 01048, 2023.

[12] J. I. Janjua, R. Ahmad, S. Abbas, A. S. Mohammed, M. S. Khan, A. Daud, T. Abbas, and M. A. Khan, "Enhancing smart grid electricity prediction with the fusion of intelligent modeling and XAI integration," International Journal of Advanced and Applied Sciences, vol. 11, no. 5, pp. 230–248, 2024.

[13] F. G. Y. Souhe, C. F. Mbey, A. T. Boum, P. Ele, and V. J. Foba Kakeu, "A hybrid model for forecasting the consumption of electrical energy in a smart grid," The Journal of Engineering, vol. 2022, no. 6, pp. 629–643, 2022.

[14] A. Emrouznejad, V. Panchmatia, R. Gholami, C. M. Rigsbee, and H. B. Kartal, "Analysis of smart meter data with machine learning for implications targeted towards residents," International Journal of Urban Planning and Smart Cities, 2023, doi: 10.4018/IJUPSC.318337.

[15] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," Energy and Buildings, vol. 140, pp. 81–97, 2017.

[16] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] S. J. Taylor and B. Letham, "Forecasting at scale," The American Statistician, vol. 72, no. 1, pp. 37–45, 2018.

[19] D. H. Wolpert, "Stacked generalization," Neural Networks, vol. 5, no. 2, pp. 241–259, 1992.

[20] S. S. Amiri, S. Mottahedi, E. R. Lee, and S. Hoque, "Peeking inside the black-box: Explainable machine learning applied to household transportation energy consumption," Computers, Environment and Urban Systems, vol. 88, art. 101647, 2021.

[21] L. Cao, Y. Cai, Y. Yue, S. Cai, and B. Hang, "A novel data fusion strategy based on extreme learning machine optimized by bat algorithm for mobile heterogeneous wireless sensor networks," IEEE Access, vol. 8, pp. 16057–16072, 2020.

International Journal of Advanced Research