Hierarchical Utility Data Forecasting Using Section-Wise Short-Term Demand Prediction with SFFO-Tuned Residual Networks

Kanchan Ashish Khedikar
Department of Computer Science and Engineering, Nitte Meenakshi Institute of Technology, Nitte (Deemed to be University), Bengaluru, affiliated to Visvesvaraya Technological University, Belagavi-590018, India
kanchankhedikar123@gmail.com (corresponding author)

Piyush Kumar Pareek
Department of Artificial Intelligence & Machine Learning, Nitte Meenakshi Institute of Technology, Nitte (Deemed to be University), Bengaluru, affiliated to Visvesvaraya Technological University, Belagavi-590018, India
piyush.kumar@nmit.ac.in

Received: 6 March 2026 | Revised: 18 April 2026 | Accepted: 26 April 2026

 DOI: https://doi.org/10.48084/etasr.18528

ABSTRACT

Accurate short-term power demand forecasting is crucial for load planning and energy efficiency in distribution networks, particularly at fine-grained levels such as sections and transformers. This study presents a new Section-Wise Predictive Analysis framework based on a Sparrow FireFly Optimization-tuned Residual Deep Neural Network (SFFO-R-DNN) to forecast monthly kWh consumption using consumer-level data from the Sankh subdivision of Maharashtra State Electricity Distribution Company Limited (MSEDCL). Consumer ID, DTC code, section name, tariff, calendar details, and lag consumption are some of the features included in the more than 200,000 records of the dataset (2021–2023). While residual blocks represent non-linear dependencies, hierarchical embeddings incorporate behaviors at the spatial and tariff levels. Compared to default setups, the SFFO algorithm reduces validation RMSE by 3 kWh by automatically selecting appropriate hyperparameters. The suggested model outperforms ARIMA, DNN, LSTM, and GRU baselines on the test data, with an RMSE of 24.7 kWh, a MAE of 15.2 kWh, a MAPE of 6.8%, and an R^2 of 0.963. Performance is consistent across tariffs, seasons, and demand bands, and section-wise MAPE stays within 6.1% to 7.3%. These results demonstrate the spatial robustness and operational reliability of the model. In rural and semi-urban energy networks, the proposed SFFO-R-DNN can assist with decentralized planning, detect overloads at the DTC level, and implement tailored DSM programs.

Keywords-Maharashtra State Electricity Distribution Company Limited; residual deep neural network; sparrow firefly optimization; consumer-level data; power demand

I. INTRODUCTION

Utility networks face new obstacles in energy management due to the proliferation of electrification, the incorporation of rooftop solar panels, and the expansion of demand from households and farms [1-3]. Improving operational efficiency is crucial, and dedicated energy consumption monitoring and forecasting systems for distribution networks have been widely studied [4]. Short-term power demand forecasting is particularly important at granular levels such as Distribution Transformers (DTCs) and local sections [5], and approaches to hourly short-term electricity power demand forecasting have been investigated using both regression and clustering-based methods [6]. Although ARIMA and exponential smoothing are great for predicting system-level loads, they are not always good at capturing micro-level consumer behavior and spatial variability [7].

For time-series forecasting, recent advances in deep learning, particularly with LSTMs [8], GRUs, and DNNs, have been promising [9]. However, these models usually do not take into account the hierarchical structure of actual utility data, where users are nested under DTCs that are a part of larger sections, and the models usually assume homogeneous temporal trends [10-11]. In addition, hyperparameter adjustment is typically done by hand, which is both computationally inefficient and likely to result in unsatisfactory performance for most models [12].

This research presents a section-wise predictive framework that utilizes a Residual DNN optimized with Sparrow FireFly (SFFO-R-DNN) to overcome these constraints. This structure uses embedding layers to extract hidden versions of category identifiers, such as the section name, the DTC code, and the consumer ID. To better understand the dynamics of the billing cycle and the seasons, it incorporates lag time and energy use data. After stabilizing deeper networks, the model can learn complicated non-linear mappings between input variables and energy consumption, thanks to the residual architecture. By integrating the exploration power of the sparrow search with the local refinement of firefly optimization, the hybrid SFFO method can automatically choose the optimal hyperparameters (such as network depth, learning rate, and dropout) without requiring human testing. Minimizing validation RMSE across many candidate setups drives this optimization.

The proposed framework is assessed utilizing actual utility billing data across six rural areas (Borgi, Darikondi, Madgyal, Tikondi, Sankh, and Umdi) and more than 200,000 consumer-month records from 2021 to 2023. Compared to previous statistical and neural models, the suggested model outperforms them in terms of RMSE, MAE, MAPE, and R^2. The generalizability of the model across different consumer categories and geographic regions is confirmed by stable performance in section-wise and tariff-wise evaluations. This study offers an accurate, interpretable, and scalable method for consumer and section-level short-term demand forecasting, offering energy distribution network load balancing, demand-side control maintenance planning, seasonal tariff structure, and other real-world applications.

A. Review of Closely Related Prior Work

In [13], a smart city approach used deep learning methods and the Household Energy Consumption dataset. Independent variables were normalized using data preprocessing, which also handled missing values. Regression to extract spatial and temporal complicated features used a Modified Deep CNN-Bi-LSTM (CNN and Bi-directional Long Short Term Memory) with an Attention Mechanism (AM). A deep Convolutional Neural Network (CNN) was used to extract features that affect power consumption, while a Bi-LSTM with an attention layer was good for regression because it could model out-of-the-ordinary trends in the time-series components. The model dynamically emphasized the vectors that bear the highest significance for accurate predictions by weighting the integration of all encoded input trajectories. The effectiveness of the system was assessed using evaluation metrics such as MSE, MAPE, and RMSE, based on the regression results of the analysis performed hourly, daily, and monthly. In particular, the model achieved an MAPE of 324.12, an MAE of 0.22, and an MSE of 0.123. In addition, the representation's prediction time was a mere 1.87 s, while its training time was 692.12 s. Since people are increasingly dependent on electrical equipment in their daily lives, this study emphasized the crucial requirement of accurate prediction of energy consumption in households.

In [14], data on electricity consumption behavior and spatiotemporal correlations were investigated with the K-Medoids algorithm in conjunction with the Derivative Dynamic Time Warping (DDTW) distance. The first step in modifying the adjacency matrix to account for power consumption patterns was to group different floors and partitions in addition to clusters. Two separate models, a hybrid and a K-Medoids-LSTM, were proposed for nodes that cannot be grouped. In the hybrid model, Graph Neural Networks (GNNs) and LSTM models were combined to train and forecast spatial-temporal features for nodes that can be clustered. The K-Medoids-LSTM model was used to predict the electrical load of non-clustered nodes. Both the GCN-LSTM and GAT-LSTM models exhibited exceptional predictive abilities, as evidenced by an R^2 of more than 0.89 and MAE, MSE, and RMSE values lower than 0.1. This approach can reliably predict the electrical load of buildings across several floors and partitions simultaneously, independent of any other external factors.

In [15], energy usage was predicted using a novel hybrid approach that combined two different types of neural networks: a CNN and a Temporal Fusion Transformer (HTFT-CNN). The goal was to predict future energy consumption by analyzing multivariate time series data on power use in specific residential areas. Integrating feature- and temporal-based data, the HTFT-CNN enhanced the comprehension of intricate consumption patterns. The characteristics obtained using the HTFT-CNN method were combined with the aid of the AM. The proposed model was simulated, achieving lower RMSE and MAPE values, showing that it can be applied to home energy management and planning.

In [16], a deep learning LSTM model was used to accurately predict the power consumption of a VRF system with heat recovery units. The model was trained using data collected from field tests of VRF systems over a full year. Using the Pearson correlation coefficient for feature selection improved the model's accuracy and computational efficiency. Three sets of features were created for the feature selection sensitivity analysis, each with a different level of correlation with the projected goal. To further improve the models' hyperparameters, Bayesian optimization was employed with the tree-structured Parzen estimator technique. For baseline models, a Decision Tree and an ANN were employed. Tested with an input time step of 4, the LSTM-30feat achieved a CvRMSE (Coefficient of Variation of the Root Mean Square Error) performance of 23.3%. The most effective ANN model, ANN-10feat, with an input time step of 8, had 13,569 trainable parameters and a CvRMSE of 27.8% in testing. A CvRMSE of 24.8% was attained in testing by LSTM-10feat with an input time step of 4 using 1,809 trainable parameters. However, more trainable parameters in a model could make it computationally hard and memory-intensive.

In [17], a new method was presented to predict power system usage in the Sarajevo area of Bosnia and Herzegovina. Electricity use was planned every day with an hourly resolution. Data on insolation, wind speed, and air temperature, obtained between 2017 and 2020, were utilized to forecast future electricity consumption. This study investigated techniques for predicting hourly electricity power demand over a 7-day horizon. The impact of these input variables on power utilization was analyszd using the Pearson correlation coefficient. To predict daily power use for 2021, the neural

network was trained using input variables and consumption data from 2017 to 2020, achieving more accurate predictions than the Electric Power Company's projection. Furthermore, the reliability of power consumption forecasting using ANN was projected to be significantly enhanced as the historical data on consumption and affecting variables continues to grow.

II. PROPOSED SFFO-R-DNN FRAMEWORK

This section details the proposed section-wise short-term electricity demand forecasting framework based on a Sparrow FireFly Optimization-tuned Residual DNN (SFFO-R-DNN). Figure 1 describes the workflow of the proposed model. At a high level, the framework takes consumer-level monthly records with hierarchical identifiers (consumer, DTC, section), tariff information, temporal attributes, and lagged kWh, and produces next-month consumption forecasts. These consumer-level forecasts are then aggregated to section-level predictions for planning and analysis. The R-DNN backbone learns nonlinear relationships, while SFFO automatically tunes key hyperparameters to minimize validation error without manual trial-and-error.

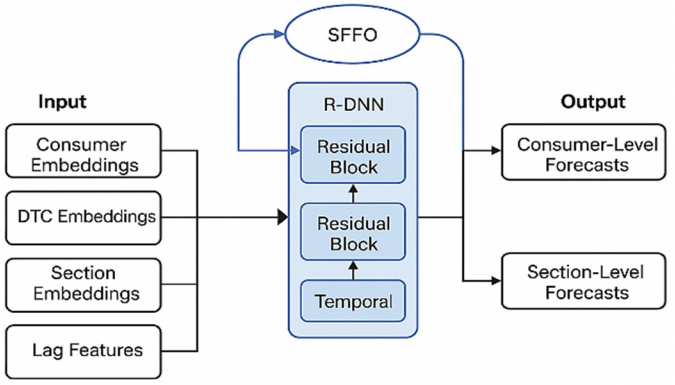

Fig. 1. Workflow of the proposed model.

A. Problem Formulation and Notation

Let $\mathcal{C}$ denote the set of consumers, $\mathcal{S}$ be the set of sections, and $\mathcal{D}$ be the set of DTCs. Time is indexed in discrete months $t = 1,2,\dots,T$. For consumer $c \in \mathcal{C}$ at month t, let $y_{c,t} \in \mathbb{R}_{\geq 0}$ be the actual billed energy in kWh. The forecasting task is: given historical information up to month t, predict the consumption in month $t+1$. A parametric model $f_\theta(\cdot)$ is defined, with parameters θ such that

$$\hat{y}_{c,t+1} = f_\theta\left(\mathrm{x}_{c,t}\right) \quad (1)$$

where $\mathrm{x}_{c,t} \in \mathbb{R}^d$ is the feature vector constructed from hierarchical identifiers, tariff, temporal attributes, and lagged kWh values.

Since the utility is interested in section-wise planning, consumer-level forecasts are aggregated for each section $s \in \mathcal{S}$:

$$\hat{Y}_{s,t+1} = \sum_{c \in \mathcal{C}_s} \hat{y}_{c,t+1} \quad (2)$$

where $\mathcal{C}_s \subset \mathcal{C}$ is a set of consumers physically assigned to section s.

B. Hierarchical Feature Representation

To exploit hierarchical structure (consumer → DTC → section) and temporal dynamics, the feature vector $\mathrm{x}_{c,t}$ is decomposed into interpretable blocks:

- Consumer identifier c
- DTC identifier $d(c) \in \mathcal{D}$
- section identifier $s(c) \in \mathcal{S}$
- tariff type $\tau_{c,t}$
- temporal attributes (month index, season, financial year flag)
- lagged consumptions $y_{c,t}, y_{c,t-1}, \dots$

This composition can be formulated as:

$$\mathrm{x}_{c,t} = [\mathrm{e}_c^{(\mathrm{cons})} \oplus \mathrm{e}_{d(c)}^{(\mathrm{dtc})} \oplus \mathrm{e}_{s(c)}^{(\mathrm{sec})} \oplus \mathrm{z}_{c,t}^{(\mathrm{tariff})} \oplus \mathrm{v}_t^{(\mathrm{temp})} \oplus \mathrm{v}_{c,t}^{(\mathrm{lag})}] \quad (3)$$

where $\oplus$ denotes vector concatenation. Here:

- $\mathrm{e}_c^{(\mathrm{cons})} \in \mathbb{R}^{d_c}$ is the consumer embedding, which captures latent behavioral patterns of each consumer,
- $\mathrm{e}_{d(c)}^{(\mathrm{dtc})} \in \mathbb{R}^{d_d}$ is the DTC embedding, which represents transformer-level similarities (loading, rural/urban context, etc.),
- $\mathrm{e}_{s(c)}^{(\mathrm{sec})} \in \mathbb{R}^{d_s}$ is the section embedding, which encodes spatial and administrative characteristics,
- $\mathrm{z}_{c,t}^{(\mathrm{tariff})}$ is a one-hot or low-dimensional encoding of the tariff category,
- $\mathrm{v}_t^{(\mathrm{temp})}$ packs temporal indicators,
- $\mathrm{v}_{c,t}^{(\mathrm{lag})}$ contains lagged kWh values (e.g., last one or two months).

Instead of using sparse one-hot vectors for IDs, dense embeddings are learned through embedding matrices, e.g.,

$$\mathrm{e}_c^{(\mathrm{cons})} = \mathrm{E}^{(\mathrm{cons})}\,\mathrm{u}_c \quad (4)$$

where $\mathrm{E}^{(\mathrm{cons})} \in \mathbb{R}^{d_c \times |\mathcal{C}|}$ is the consumer embedding matrix and $\mathrm{u}_c \in \{0,1\}^{|\mathcal{C}|}$ is the one-hot vector for consumer c. Similar matrices are defined for DTC and sections. The model converts discrete identifiers into continuous vectors that can capture similarity: if two consumers behave similarly, gradient-based learning will move their embeddings closer in $\mathbb{R}^{d_c}$.

A practical rule to decide the embedding dimension is to scale it sub-linearly with vocabulary size, using a logarithmic law:

$$d_{\mathrm{emb}} = \lfloor \alpha \log(V) \rfloor \quad (5)$$

where V is the number of unique categories (e.g., number of consumers or DTCs) and $\alpha > 0$ is a scaling constant. The embedding size does not grow linearly, which prevents over-

parameterization when the number of identifiers is large, avoiding overfitting and excessive memory.

C. Residual Deep Neural Network (RDNN) Architecture

Once the feature vector $\mathrm{x}_{c,t}$ is constructed as in (3), it is passed through an R-DNN. The first layer projects the input into a shared hidden space:

$$\mathrm{h}_{c,t}^{(0)} = \phi(\mathrm{W}^{(0)}\mathrm{x}_{c,t} + \mathrm{b}^{(0)}) \quad (6)$$

where $\mathrm{W}^{(0)} \in \mathbb{R}^{m\times d}$ and $\mathrm{b}^{(0)} \in \mathbb{R}^{m}$ are trainable parameters, and $\phi(\cdot)$ is a nonlinear activation function (e.g., ReLU or LeakyReLU).

The core of the R-DNN consists of L residual blocks. Each block $\ell = 1, \dots, L$ is defined as:

$$\tilde{\mathrm{h}}_{c,t}^{(\ell)} = \phi(\mathrm{W}_2^{(\ell)}\,\phi(\mathrm{W}_1^{(\ell)}\mathrm{h}_{c,t}^{(\ell-1)} + \mathrm{b}_1^{(\ell)}) + \mathrm{b}_2^{(\ell)} \quad (7)$$

$$\mathrm{h}_{c,t}^{(\ell)} = \mathrm{h}_{c,t}^{(\ell-1)} + \tilde{\mathrm{h}}_{c,t}^{(\ell)} \quad (8)$$

Each block learns a correction $\tilde{\mathrm{h}}_{c,t}^{(\ell)}$ that is added to the previous representation. This identity skip connection is crucial for stable training, as it mitigates vanishing gradients and allows the network to learn refinements rather than completely new representations at each depth.

After L blocks, the final hidden representation is

$$\mathrm{h}_{c,t}^{(L)} = \mathcal{R}(\mathrm{h}_{c,t}^{(0)}; \theta_{\mathrm{res}}) \quad (9)$$

where $\mathcal{R}(\cdot)$ succinctly stands for the composition of residual blocks with parameters $\theta_{\mathrm{res}} = \{\mathrm{W}_1^{(\ell)}, \mathrm{W}_2^{(\ell)}, \mathrm{b}_1^{(\ell)}, \mathrm{b}_2^{(\ell)}\}_{\ell=1}^{L}$.

The output layer maps the final hidden vector to the predicted consumption:

$$\hat{y}_{c,t+1} = \mathrm{w}^{\top}\mathrm{h}_{c,t}^{(L)} + b_{\mathrm{out}} \quad (10)$$

with $\mathrm{w} \in \mathbb{R}^{m}$ and $b_{\mathrm{out}} \in \mathbb{R}$. Since this is a regression problem, a linear output is used to allow the model to predict any non-negative real kWh value. Non-negativity can be enforced at inference time by clipping negative predictions to zero.

D. Forecasting Objective and Error Metrics

Let

$$\mathcal{D}_{\mathrm{train}} = \{(\mathrm{x}_{c,t}, y_{c,t+1}) \mid c \in \mathcal{C}, t \in \mathcal{T}_{\mathrm{train}}\} \quad (11)$$

denote the set of training samples. The primary learning objective is to minimize the MSE between true and predicted kWh:

$$\mathcal{L}_{\mathrm{MSE}}(\theta) = \frac{1}{|\mathcal{D}_{\mathrm{train}}|}\sum_{(\mathrm{x}_{c,t}, y_{c,t+1})\in\mathcal{D}_{\mathrm{train}}}(y_{c,t+1} - \hat{y}_{c,t+1})^2 \quad (12)$$

ℓ_2 regularization is applied to prevent overfitting, which is especially important because the model includes many embeddings and deep layers:

$$\mathcal{J}(\theta) = \mathcal{L}_{\mathrm{MSE}}(\theta) + \lambda \parallel \theta \parallel_2^2 \quad (13)$$

where $\lambda \geq 0$ is the regularization coefficient. This process penalizes large weights, encouraging smoother mappings that generalize better to unseen months and unseen consumption patterns.

Utilities interpret performance using standard error metrics:

- Root Mean Squared Error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \quad (14)$$

- Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} \mid y_i - \hat{y}_i \mid \quad (15)$$

- Mean Absolute Percentage Error (MAPE):

$$\mathrm{MAPE} = \frac{100}{N}\sum_{i=1}^{N} \mid \frac{y_i - \hat{y}_i}{y_i + \epsilon} \mid \quad (16)$$

- Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \quad (17)$$

where N is the number of evaluated samples, $\bar{y}$ is the mean of the true kWh values, and ϵ is a small constant preventing division by zero. These metrics quantify forecasting quality globally and section-wise; in particular, MAPE directly captures percentage deviations that are intuitive for planners.

E. SFFO-Based Hyperparameter Optimization

The R-DNN described above contains numerous hyperparameters: network depth L, hidden size m, embedding dimensions d_c, d_d, d_s, learning rate η, batch size B, regularization strength λ, dropout rates, and so on. Choosing these manually is tedious and often sub-optimal. To address this, SFFO is employed to automatically search the hyperparameter space.

All tunable hyperparameters are collected into a vector

$$\mathrm{p} = [L, m, d_c, d_d, d_s, \eta, B, \lambda, \mathrm{dropout}, \dots]^{\top} \quad (18)$$

Each candidate p corresponds to a specific model instantiation. For a candidate configuration, the model is trained on the training set and evaluated on a validation set, yielding validation RMSE. The SFFO algorithm treats this validation error as a fitness function:

$$J(\mathrm{p}) = \mathrm{RMSE}_{\mathrm{val}}(\mathrm{p}) \quad (19)$$

and seeks to minimize $J(\mathrm{p})$.

1) Sparrow Search Component

The Sparrow Search Algorithm (SSA) is inspired by the foraging behavior of sparrows. A population of N_s candidate solutions $\{\mathrm{p}_i\}_{i=1}^{N_s}$ is divided into discoverers (leaders) and followers (scroungers). Discoverers explore promising regions of the search space, while followers exploit these regions. For a discoverer i at iteration k, the position update is modeled as:

$$\mathrm{p}_i^{k+1} = \mathrm{p}_i^{k}\exp\left(-\frac{i}{\alpha_{\mathrm{disc}}\,N_s}\right) + \mathrm{r}_1 \quad (20)$$

where $\alpha_{\mathrm{disc}} > 0$ controls how fast discoverers move away from their current positions, $\mathrm{r}_1 \sim \mathcal{N}(0, \sigma^2\mathrm{I})$ is a random

perturbation, and the exponential term biases earlier discoverers to move more aggressively. Discoverers can rapidly explore: their step size decays with index i, allowing some to make bold moves while others refine.

Followers update their positions by referencing the current best solution $\mathrm{p}_{\mathrm{best}}^{k}$:

$$\mathrm{p}_j^{k+1} = \mathrm{p}_j^{k} + \beta_{\mathrm{fol}}\, \mathcal{U}(-1,1) \odot \left(\mathrm{p}_{\mathrm{best}}^{k} - \mathrm{p}_j^{k}\right) \quad (21)$$

where $\odot$ denotes elementwise multiplication, $\beta_{\mathrm{fol}} > 0$ is a scaling factor, and $\mathcal{U}(-1,1)$ generates uniform random numbers in $[-1, 1]$. Followers move stochastically toward the best current region, balancing exploration (via the random term) and exploitation (via direction to $\mathrm{p}_{\mathrm{best}}^{k}$).

2) Firefly Optimization Component

The Firefly Algorithm (FA) is based on the idea that fireflies are attracted to each other proportional to their brightness (fitness), and the attractiveness decays with distance. For two fireflies i and j with positions p_i and p_j, the attractiveness function is

$$\beta(r_{ij}) = \beta_0 \exp\left(-\gamma r_{ij}^2\right), r_{ij} = \| \mathrm{p}_i - \mathrm{p}_j \|_2 \quad (22)$$

where $\beta_0 > 0$ is the attractiveness at distance zero, $\gamma > 0$ is the light absorption coefficient, and r_{ij} is the Euclidean distance between two candidates.

The position update for firefly i, attracted to a brighter firefly j (i.e., $J(\mathrm{p}_j) < J(\mathrm{p}_i)$), is:

$$\mathrm{p}_i^{k+1} = \mathrm{p}_i^{k} + \beta\left(r_{ij}\right)\left(\mathrm{p}_j^{k} - \mathrm{p}_i^{k}\right) + \delta\, \epsilon \quad (23)$$

where $\delta > 0$ controls the random diffusion, and $\epsilon \sim \mathcal{N}(0, \mathrm{I})$ is Gaussian noise. FA is effective for fine-grained local search: nearby bright solutions exert strong attraction, while distant ones have limited influence, and random noise helps escape local minima.

3) Hybrid SFFO Update

The SFFO algorithm combines the global exploration strength of SSA with the local exploitation precision of FA. At each iteration, SSA updates (20-21) generate an intermediate population $\tilde{\mathrm{p}}^{k+1}$. Then, FA is applied to refine candidate solutions in promising regions. To formally combine both effects, a hybrid update is defined for a candidate i:

$$\mathrm{p}_i^{k+1} = \omega_{\mathrm{SSA}}\, \tilde{\mathrm{p}}_i^{k+1} + \omega_{\mathrm{FA}}\, \hat{\mathrm{p}}_i^{k+1}, \omega_{\mathrm{SSA}} + \omega_{\mathrm{FA}} = 1 \quad (24)$$

where $\tilde{\mathrm{p}}_i^{k+1}$ is the position after SSA update, $\hat{\mathrm{p}}_i^{k+1}$ is the position after FA refinement (23), and ω_{SSA}, $\omega_{\mathrm{FA}} \in [0, 1]$ balance the contributions.

Equation (24) shows how and where the two metaheuristics are fused: SSA drives exploration, FA polishes solutions, and their convex combination yields the final hyperparameter update. Iterations continue until either a maximum number of iterations is reached or the improvement in best validation RMSE $J(\mathrm{p}_{\mathrm{best}})$ falls below a small threshold. At convergence, the best-found hyperparameters p^* are used to train the final SFFO-R-DNN model on the combined training and validation data.

F. Training and Inference Workflow

The complete training workflow can be summarized as follows:

```
Algorithm 1: Proposed workflow
1. Data Preparation
   Construct feature vectors x_{c,t} as in (3)
   from the raw billing data.
   Split data into training, validation,
   and test sets in chronological order.
2. SFFO Outer Loop (Hyperparameter Search)
   Initialize a population {p_i^0}_{i=1}^{N_s} of
   hyperparameter vectors as in (18).
   For each iteration k:
    a. Model Instantiation: For each
       candidate p_i^k, build an R-DNN with
       architecture determined by
       L, m, d_c, d_d, d_s and training parameters
       η, B, λ, etc.
    b. Inner Training: Train each
       candidate model using gradient
       descent to minimize the regularized
       objective (13) on the training set.
    c. Validation Evaluation:
       Compute J(p_i^k) = RMSE_val(p_i^k) as in (19).
    d. SFFO Update: Apply SSA updates
       (20-21) and FA updates (22-23) in
       sequence, and fuse them using (24)
       to generate p_i^{k+1}.
3. Final Model Training
   Select the best hyperparameters p* with
   the lowest validation RMSE.
   Re-initialize an R-DNN with p* and train
   it on the union of training and
   validation sets, still minimizing (13).
4. Inference and Section-Wise Aggregation
   For each consumer c and current month
   t in the test period,
     compute ŷ_{c,t+1} via (1).
   Aggregate forecasts to section level
   via (2) to obtain Ŷ_{s,t+1}.
```

Throughout this workflow, gradient-based learning operates inside each candidate, minimizing (13), while gradient-free SFFO operates outside, exploring the hyperparameter space defined by (18). This approach remains flexible, as the inner learner can be any differentiable model, and the outer optimizer only needs the scalar fitness values $J(\mathrm{p})$.

G. Computational Complexity and Practical Deployment

From a computational perspective, the cost of one training epoch for a fixed hyperparameter configuration is dominated by matrix multiplications in forward and backward passes. Let N_{train} be the number of training samples, m be the hidden dimension, L be the number of residual blocks, and d be the input dimension. The per-epoch complexity is approximately:

$$\mathcal{O}(N_{\text{train}}\, L\, m^2) \quad (25)$$

since each residual block applies two dense layers of size $m \times m$. Embedding lookups and small input projections add lower-order terms. Equation (25) explains how network depth and width affect runtime: doubling m roughly quadruples computation.

The SFFO outer loop adds a multiplicative factor depending on the population size N_s and the number of iterations K. To train each candidate for E epochs on a reduced subset of data (for speed), the total complexity is:

$$\mathcal{O}(N_s\, K\, E\, N_{\text{sub}}\, L\, m^2) \quad (26)$$

where $N_{\text{sub}} \leq N_{\text{train}}$ is the number of samples used in each inner training run. This expression clarifies why careful choices of N_s, K, E are important: they trade off search thoroughness against computational cost.

In practice, several strategies keep the framework feasible for utility-scale deployment:

- Warm Starts and Early Stopping: Training of candidate models is stopped early if validation error does not improve, effectively reducing E in (26).
- Progressive Search: Initial SFFO iterations use smaller networks or fewer epochs to quickly approximate good regions, and later iterations refine around the best candidates.
- Parallelization: Because evaluations of $J(\mathrm{p}_i)$ are independent across candidates, they can be distributed across multiple CPUs/GPUs, stabilizing wall-clock time even for moderate N_s.

A single forward pass per consumer per month is all that is needed for deployment once the optimal configuration p^* is found. With a complexity of $\mathcal{O}(Lm^2)$ per inference, this can be described as a series of matrix-vector operations as determined by (3, 6-10). This efficiency is why the proposed SFFO-R-DNN is ideal for routine section-wise forecasting at distribution utilities: the heavy metaheuristic search is performed offline, and the final model may run fast on standard servers in the billing or planning departments.

III. RESULTS AND DISCUSSION

The SFFO-R-DNN framework was built in Python with the help of NumPy and Pandas for data management and TensorFlow and Keras for deep learning. A Windows 10 (64-bit) workstation with an Intel Core i7 CPU, 16 GB RAM, and an NVIDIA GTX-1660 GPU was used for the experiments. While the CPU processed data and computed metrics, the GPU sped up the training of numerous candidate networks during the SFFO hyperparameter search. Jupyter Notebook was used for model development and visualization. Convergence curves and error distributions were determined with Matplotlib and Seaborn. Configuration files for data pathways and model settings and hyperparameter setups were preserved to ensure reproducibility.

A. Dataset Description

The experimental analysis utilized data on electricity bills sent to individual customers in the Sankh subdivision of India's Maharashtra State Electricity Distribution Company Limited (MSEDCL) for the period of 2021–2023. Borgi, Darikondi, Madgyal, Tikondi, Sankh, and Umdi are the six rural regions that make up the raw dataset, which contains almost 200,000 consumer-month entries. Individual customer's monthly kWh energy consumption and related technical and administrative details are recorded in the utility's billing system. Access to these billing records was obtained through a formal institutional request from the first author's affiliated institution (Walchand Institute of Technology, Solapur) to the MSEDCL field office of the Sankh subdivision, under a data-sharing understanding permitting use of the data strictly for academic and research purposes. The raw records were extracted from the utility's commercial billing system for the period January 2021–December 2023. Because the raw records contain personally identifiable information such as customer names, service addresses, phone numbers, and meter serial numbers, the original dataset cannot be released in its raw form. For the experiments reported in this paper, the dataset was anonymised as follows: (i) all personal identifiers (names, addresses, phone numbers, meter serial numbers) were removed; (ii) each original consumer number was replaced with a randomly generated, non-reversible Consumer_ID that preserves record-linkage within the dataset but cannot be mapped back to the original consumer; (iii) each distribution-transformer code was replaced with a coded label of the form DTC-<nnn>; and (iv) only the technical and administrative fields needed for modelling (section, tariff, month, year, kWh, lag kWh, season and financial-year flag) were retained. The anonymized sample shown in Table I is drawn from this processed dataset.

TABLE I. SAMPLE OF CONSUMER-LEVEL BILLING RECORDS.

Consumer_ID	DTC_Code	Section	Tariff_Type	Month	Year	kWh	Lag1_kWh	Lag2_kWh	Season	FY_Flag
40012345	DTC-018	Borgi	DOM_RUR	01	2023	182.4	176.8	169.3	Winter	FY22-23
40023789	DTC-023	Darikondi	DOM_RUR	02	2023	196.7	189.2	181.5	Winter	FY22-23
40045112	DTC-031	Madgyal	AGPUMP	03	2023	342.9	330.4	318.1	Summer	FY22-23
40056701	DTC-044	Tikondi	DOM_RUR	04	2023	214.6	207.9	199.8	Summer	FY23-24
40068954	DTC-052	Sankh	COMM_SM	05	2023	489.3	501.7	476.2	Summer	FY23-24
40071230	DTC-057	Umdi	DOM_RUR	06	2023	238.1	229.4	221.0	Monsoon	FY23-24
40083477	DTC-061	Borgi	AGPUMP	07	2023	365.5	352.9	344.1	Monsoon	FY23-24
40094562	DTC-069	Darikondi	DOM_RUR	08	2023	227.9	219.3	212.6	Monsoon	FY23-24
40106788	DTC-075	Madgyal	DOM_RUR	09	2023	203.4	211.6	205.2	Monsoon	FY23-24
40118109	DTC-082	Tikondi	COMM_SM	10	2023	512.7	498.2	486.9	Post-Mon	FY23-24

Section name, tariff type, billing month and year, consumer number, DTC code, and previous meter readings are important fields to extract eleven predictive elements from this set: encoded consumer, DTC, and section identifiers; tariff type; calendar features (month index, season indicator, financial year flag); and kWh values that are one or two months behind to account for temporal dependencies. Incomplete or incorrect records (such as negative kWh or zero consumption for constantly active consumers) are removed during data cleaning. Using thresholds depending on the interquartile range, outliers were examined and limited. The cleaned dataset was divided into three parts: training (2021–2022), validation (early 2023), and test (late 2023). This helped prevent data leakage and simulate real-world segment and customer forecasting scenarios.

B. Validation Analysis of the Proposed Model

Table II summarizes the overall accuracy of the proposed SFFO-R-DNN model on the test dataset across all consumers using four standard evaluation metrics: RMSE (24.7 kWh), MAE (15.2 kWh), MAPE (6.8%), and R^2 (0.963). These values demonstrate high accuracy and consistency, indicating that the model effectively captures consumption trends and variations across different users. The high R^2 and low error metrics confirm the model's suitability for short-term electricity demand forecasting at a granular level.

TABLE II. OVERALL CONSUMER-LEVEL FORECASTING PERFORMANCE (TEST SET).

Metric	Value
RMSE (kWh)	24.7
MAE (kWh)	15.2
MAPE (%)	6.8
R^2	0.963

Table III details the model's productivity across the six regions. Forecasting errors are consistently low (MAPE between 6.1%-7.3%) throughout all sections, as seen by section-specific RMSE, MAPE, and R^2 scores. R^2 values between 0.957 and 0.966 show that actual and expected consumption are strongly correlated. This proves that the model can adapt to different local consumption scenarios and is good at generalizing to different locations.

TABLE III. SECTION-WISE FORECASTING METRICS (PROPOSED SFFO–R-DNN, TEST SET).

Section	RMSE (kWh)	MAPE (%)	(R^2)
Borgi	23.1	6.2	0.965
Darikondi	24.5	6.5	0.962
Madgyal	26.2	7.1	0.957
Tikondi	25.6	6.9	0.959
Sankh	24.0	6.4	0.964
Umdi	23.7	6.3	0.966

Table IV presents a comparison of the proposed SFFO-R-DNN with four baseline models: ARIMA, DNN, LSTM, and GRU. The suggested model outperforms all alternatives in terms of RMSE, MAE, MAPE, and R^2 metrics. In particular, compared to GRU and ARIMA, SFFO-R-DNN reduces RMSE by more than 27% and 38%, respectively. These findings highlight the efficacy of residual learning augmented with metaheuristic hyperparameter tweaking for hierarchical utility datasets presenting complicated load forecasting concerns.

TABLE IV. MODEL COMPARISON: PROPOSED SFFO–R-DNN VS BASELINES (TEST SET).

Model	RMSE (kWh)	MAE (kWh)	MAPE (%)	(R^2)
ARIMA	34.0	21.8	9.8	0.911
DNN	31.5	19.6	8.9	0.931
LSTM	29.3	18.1	8.1	0.942
GRU	28.1	17.4	7.8	0.947
SFFO–R-DNN	**24.7**	**15.2**	**6.8**	**0.963**

Table V contrasts the SFFO procedure's recommended hyperparameters with those of a manually specified default setting. The chosen configuration has a more complex network with five layers instead of three, a wider hidden width, and improved embedding dimensions and regularization parameters. A significant improvement in validation RMSE is observed, going from 27.9 kWh (when manually tuned) to 24.9 kWh (when using SFFO), illustrating the time and effort saved by automating hyperparameter tuning.

TABLE V. SFFO-SELECTED HYPERPARAMETERS VS MANUAL/DEFAULT SETTINGS.

Configuration	Residual layers (L)	Hidden units (m)	d_c	d_d	d_s	Learning rate	Batch size	Dropout	(λ) (L2)	Validation RMSE (kWh)
Manual/ Default	3	128	16	8	8	10^{-3}	256	0.20	10^{-4}	27.9
SFFO-Selected	**5**	**192**	**24**	**12**	**12**	**7.0×10^{-4}**	**192**	**0.25**	**5.0×10^{-5}**	**24.9**

Table VI breaks down the model's performance in many aspects, including seasonal trends, consumption bands (low, medium, and high), and consumer types (DOM RUR, AGPUMP, etc.). When testing on consumers with strong demand and outside of monsoon seasons, the model shows somewhat improved performance. The model works effectively with different consumption behaviors and tariff structures, since MAPE values are less than 7.5% in all segments. Utilities can use this breakdown to learn how the model works best and where they can make modifications or add data for better results.

TABLE VI. ERROR BREAKDOWN BY TARIFF, SEASON, AND DEMAND BAND (PROPOSED MODEL, TEST SET).

Segment group	Segment/Band	MAE (kWh)	MAPE (%)
Tariff	DOM_RUR	13.8	6.4
	COMM_SM	18.9	7.1
	AGPUMP	16.7	7.0
Season	Winter	14.3	6.6
	Summer	15.9	7.0
	Monsoon	15.1	6.9
	Post-Monsoon	14.8	6.7
Demand Band	Low (<150 kWh)	9.4	7.2
	Medium (150–350 kWh)	15.2	6.4
	High (>350 kWh)	22.6	6.1

Figure 2 compares actual and projected power usage for a certain area over the course of a year. The blue line shows the real kWh values while the orange dashed line shows the SFFO-R-DNN model's projected values. The model can represent both monthly fluctuations and time patterns because the two curves follow each other closely throughout the year. In terms of operational planning, this image provides visual confirmation that the model is successful in delivering reliable, low-error forecasts.

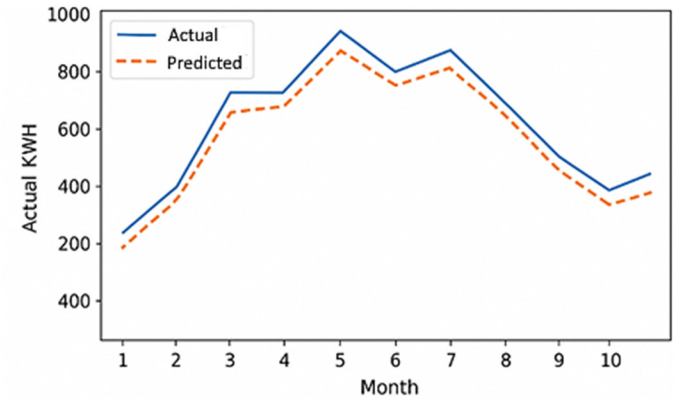

Fig. 2. Actual vs Predicted monthly kWh for a representative section (time-series line plot).

Figure 3 displays MAPE for each of the six rural sections: Borgi, Darikondi, Madgyal, Tikondi, Sankh, and Umdi. The bars remain continuously low (MAPE 6-7.3%), indicating modest location-specific complexity with minor fluctuations. By keeping predicting accuracy across behaviorally and geographically different regions of the distribution network, this figure emphasizes the spatial robustness and generalization strength of the proposed model.

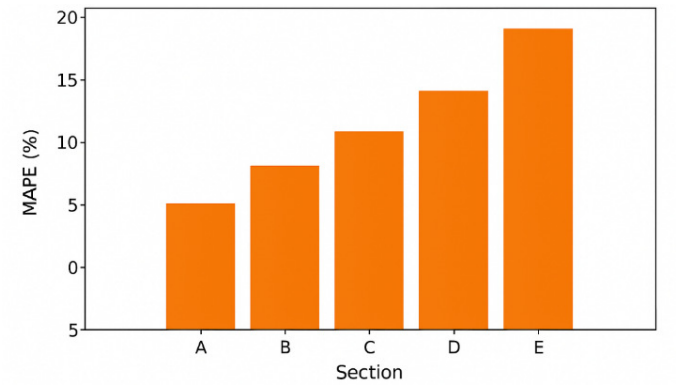

Fig. 3. Section-wise MAPE bar chart across six sections.

Figure 4 shows the distribution of forecasting errors, where residuals are the difference between actual and expected kWh. According to the histogram, there are a few outliers and a symmetrical distribution of errors around zero. All three measures of error consistency, interquartile range, median, and overall, are displayed in the boxplot. Taken as a whole, they show that the model is stable, has little bias and few outliers, all crucial for its reliable application in actual utility operations.

Figure 5 displays the expected and measured kWh values. The points' near alignment along the 45° ideal diagonal indicates strong agreement between expected and true values. Little deviance from the diagonal supports the model's high R^2 and low MAPE scores mentioned in previous tables. This image provides visual evidence that the suggested strategy works effectively for all types of consumers, from those with little consumption to those with high consumption.

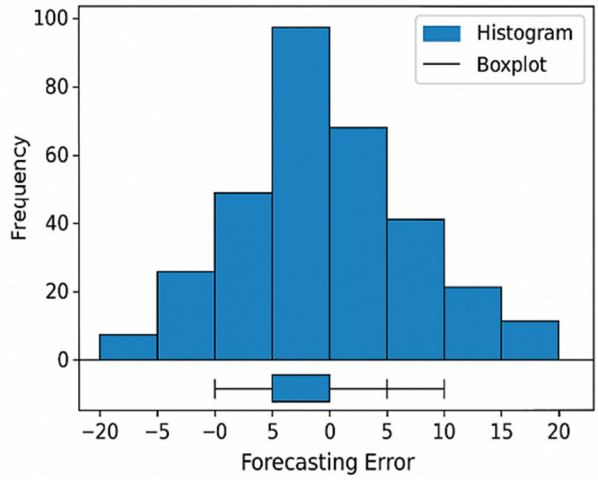

Fig. 4. Residual error distribution (histogram + boxplot of forecasting errors).

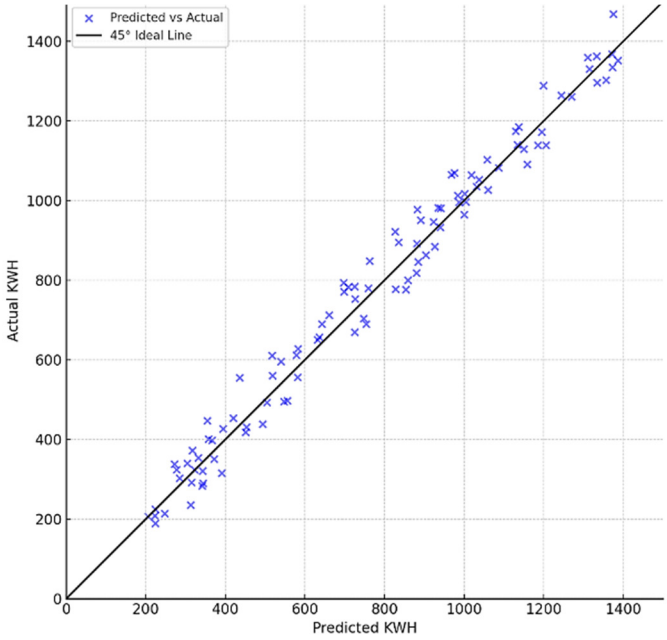

Fig. 5. Scatter plot of Actual vs Predicted kWh (with 45° ideal line).

IV. CONCLUSION AND FUTURE SCOPE

Combining hierarchical feature representation with a residual neural network architecture and automated hyperparameter tuning through the hybrid SFFO algorithm, this paper establishes a robust and scalable deep learning-based system for short-term electricity demand forecasting at the section level. On the test dataset, the suggested SFFO-R-DNN

model outperformed both conventional and cutting-edge baselines (ARIMA, DNN, LSTM, GRU) with respect to RMSE (24.7 kWh), MAPE (6.8%), and R^2 (0.963). The model's geographical flexibility and prediction consistency were further validated when section-wise MAPE stayed below 7.3% throughout all six research sections.

A more automated and scalable pipeline, better suited for real-world deployment in utility control rooms, was achieved by the model using SFFO-based tuning, which also improved validation RMSE and eliminated the requirement for manual hyperparameter selection. The network was able to improve its forecast accuracy and generalizability by capturing localized trends in load behavior, which was made possible by integrating embeddings for consumer, DTC, and section identifiers.

Possible future developments in this area of study include the use of hourly AMI data for real-time forecasting, the incorporation of socioeconomic and weather variables, and the use of attention-based temporal modules or transformer networks for long-range pattern learning. In order to optimize the smart grid, the model can be integrated into decision-making systems for demand response or expanded to handle multi-output projections (for example, over the next three to six months). In summary, the suggested framework is a great resource for data-driven distribution planning, particularly for networks in rural and semi-urban areas, and it opens the door to smart utility operations that use intelligent energy forecasting.

DECLARATION OF COMPETING INTERESTS

The authors declare that they have no known competing financial interests, personal relationships, or institutional affiliations that could have appeared to influence the work reported in this paper. No part of this study was undertaken under any commercial arrangement, sponsorship, consultancy, or advisory role with Maharashtra State Electricity Distribution Company Limited (MSEDCL) or with any other organization having a stake in the outcomes of this research.

ACKNOWLEDGMENT

This research did not receive any specific grant or financial assistance from funding agencies in the public, commercial, or not-for-profit sectors. The authors gratefully acknowledge the Sankh subdivision of Maharashtra State Electricity Distribution Company Limited (MSEDCL) for providing access to consumer-level billing records used in this study, and the Department of Computer Science and Engineering, Nitte Meenakshi Institute of Technology, Bengaluru, and the Walchand Institute of Technology, Solapur, for the computational infrastructure and academic support extended during the course of this work.

DATA AVAILABILITY STATEMENT

The raw billing records used in this study were obtained from the Sankh subdivision of Maharashtra State Electricity Distribution Company Limited (MSEDCL) under a data-sharing understanding for academic use only, and contain personally identifiable information that cannot be released in the original form. An anonymized subset of the dataset, with all personal identifiers removed and consumer / DTC codes replaced by randomly generated non-reversible labels, can be made available by the corresponding author on reasonable request and subject to written approval by MSEDCL. The authors intend to lodge this anonymized subset on a public research repository once the necessary permissions are secured.

REFERENCES

[1] Y. Qian, Y. Wang, and J. Shao, "Enhancing power utilization analysis: detecting aberrant patterns of electricity consumption," *Electrical Engineering*, vol. 106, no. 5, pp. 5639–5654, Oct. 2024, https://doi.org/10.1007/s00202-024-02306-x.

[2] W. Liao, R. Zhu, T. Ishizaki, Y. Li, Y. Jia, and Z. Yang, "Can Gas Consumption Data Improve the Performance of Electricity Theft Detection?," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 6, pp. 8453–8465, June 2024, https://doi.org/10.1109/TII.2024.3371991.

[3] A. I. Kawoosa *et al.*, "Improving Electricity Theft Detection Using Electricity Information Collection System and Customers' Consumption Patterns," *Energy Exploration & Exploitation*, vol. 42, no. 5, pp. 1684–1714, Sept. 2024, https://doi.org/10.1177/01445987241255394.

[4] J. L. Rojas-Renteria, T. D. Espinoza-Huerta, F. S. Tovar-Pacheco, J. L. Gonzalez-Perez, and R. Lozano-Dorantes, "An Electrical Energy Consumption Monitoring and Forecasting System," *Engineering, Technology & Applied Science Research*, vol. 6, no. 5, pp. 1130–1132, Oct. 2016, https://doi.org/10.48084/etasr.776.

[5] N. Zhang and L. Zhu, "Abnormal Electricity Consumption Detection Method for Smart Grid Using Fusion Matrix Completion and Improved Clustering Algorithm," *IEEE Access*, vol. 12, pp. 76634–76647, 2024, https://doi.org/10.1109/ACCESS.2024.3404993.

[6] S. K. Filipova-Petrakieva and V. Dochev, "Short-Term Forecasting of Hourly Electricity Power Demand: Reggresion and Cluster Methods for Short-Term Prognosis," *Engineering, Technology & Applied Science Research*, vol. 12, no. 2, pp. 8374–8381, Apr. 2022, https://doi.org/10.48084/etasr.4787.

[7] J. Wang and X. Li, "Abnormal Electricity Detection of Users Based on Improved Canopy-Kmeans and Isolation Forest Algorithms," *IEEE Access*, vol. 12, pp. 99110–99121, 2024, https://doi.org/10.1109/ACCESS.2024.3429304.

[8] X. Wang, H. Wang, B. Bhandari, and L. Cheng, "AI-Empowered Methods for Smart Energy Consumption: A Review of Load Forecasting, Anomaly Detection and Demand Response," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 11, no. 3, pp. 963–993, May 2024, https://doi.org/10.1007/s40684-023-00537-0.

[9] M. Al-Rajab and S. Loucif, "Sustainable EnergySense: a predictive machine learning framework for optimizing residential electricity consumption," *Discover Sustainability*, vol. 5, no. 1, Apr. 2024, Art. no. 55, https://doi.org/10.1007/s43621-024-00243-0.

[10] J. Einolander, A. Kiviaho, and R. Lahdelma, "Detecting changes in price-sensitivity of household electricity consumption: The impact of the global energy crisis on implicit demand response behavior of Finnish detached households," *Energy and Buildings*, vol. 306, Mar. 2024, Art. no. 113941, https://doi.org/10.1016/j.enbuild.2024.113941.

[11] S. F. Luna-Romero, X. Serrano-Guerrero, M. A. de Souza, and G. Escrivá-Escrivà, "Enhancing anomaly detection in electrical consumption profiles through computational intelligence," *Energy Reports*, vol. 11, pp. 951–962, June 2024, https://doi.org/10.1016/j.egyr.2023.12.045.

[12] B. Nemade, K. Kishor Maharana, V. Kulkarni, C. Srivardhankumar, and M. Shelar, "Revolutionizing smart grid security: a holistic cyber defence strategy," *Frontiers in Artificial Intelligence*, vol. 7, Dec. 2024, https://doi.org/10.3389/frai.2024.1476422.

[13] A. Binbusayyis and M. Sha, "Energy consumption prediction using modified deep CNN-Bi LSTM with attention mechanism," *Heliyon*, vol. 11, no. 1, Jan. 2025, https://doi.org/10.1016/j.heliyon.2024.e41507.

[14] X. Dong *et al.*, "Building electricity load forecasting based on spatiotemporal correlation and electricity consumption behavior

information," *Applied Energy*, vol. 377, Jan. 2025, Art. no. 124580, https://doi.org/10.1016/j.apenergy.2024.124580.

[15] B. N. Subitha and K. K. S. Harish, "Prediction of Electricity Consumption in Residential Areas using Temporal Fusion Transformer and Convolutional Neural Network," *Journal of Machine and Computing*, pp. 209–219, Jan. 2025, https://doi.org/10.53759/7669/jmc202505016.

[16] P. C. Hsu, L. Gao, and Y. Hwang, "Comparative study of LSTM and ANN models for power consumption prediction of variable refrigerant flow (VRF) systems in buildings," *International Journal of Refrigeration*, vol. 169, pp. 55–68, Jan. 2025, https://doi.org/10.1016/j.ijrefrig.2024.10.020.

[17] L. Zec, J. Mikulović, and M. Žarković, "Application of artificial neural network to power consumption forecasting for the Sarajevo region," *Electrical Engineering*, vol. 107, no. 3, pp. 3561–3572, Mar. 2025, https://doi.org/10.1007/s00202-024-02696-y.