Accepted manuscript (author version)

To appear in:

International Journal of Mathematical Modelling & Computations

Online ISSN: 2228-6233 Print ISSN: 2228-6225

Received: 8- November-2025

Revised: 3- January -2026

Accepted: 2- February-2026

DOI: https://doi.org/10.57647/ijm2c.2026.1604.28

ORIGINAL RESEARCH

Optimized Multilayer Perceptron Neural Network Using Adam Optimizer and Huber Loss Function for Electricity Consumption Forecasting in Residential Buildings: A Case Study of Isfahan City

Amirreza Alizade[1], Hojjatollah Khajeh Salehani[1], Ahmad Reza Vali[2], Ghasem Derakhshan[1], Farhad Azimifar[3]

[1]Department of Electrical Engineering, Dam.C., Islamic Azad University, Damavand, Iran

[2]Electrical Engineering Department, Malek Ashtar University of Technology, Tehran, Iran

[3]Department of Biomedical Engineering , Isf.C., Islamic Azad University, Isfahan, Iran

Abstract

Forecasting electricity consumption in residential buildings is a critical area in building energy management systems (BEMS). This study develops an optimized Multilayer Perceptron Neural Network (MLP-NN) for accurate short-term forecasting of residential electricity consumption in Isfahan City. Data were preprocessed using Min–Max Normalization (MMN). The model utilizes 9 input parameters: HVAC power consumption, lighting, household appliances, energy stored in a 5-kWh supercapacitor, ambient temperature, solar irradiance, hour of the day, holiday indicator, and seasons. The model predicts electricity consumption for the next 15-minute interval. Datasets comprised 15-minute interval records over one year from the Isfahan Electricity Distribution Company and ambient temperature data from the Iranian Meteorological Organization. The MLP-NN was trained with early stopping to prevent overfitting and optimized using Adaptive Moment Estimation (Adam) and Stochastic Gradient Descent (SGD) algorithms. Performance was compared with several machine learning models (Random Forest, SVR, XGBoost, Gradient Boosting, Extra Trees, and Linear Regression), with hyperparameters tuned by Random Search (10 iterations). Additionally, feature importance analysis using Chatterjee's Xi correlation coefficient, combined with physics-informed feature engineering, led to the selection of a refined set of 6 features. This reduced-dimensionality model, trained with the Adam optimizer and Huber loss function, achieved the best performance across all configurations on the test sets (test Huber loss = 0.002594), outperforming both the full 9 features MLP-NN and all benchmark machine learning models. This study demonstrates the superiority of the proposed MLP-NN, enhanced by temporal features, early stopping, and feature selection, for reliable short-term electricity consumption forecasting in residential buildings.

Keywords: Multilayer Perceptron Neural Network (MLP-NN), Machine Learning (ML), Huber Loss Function, Adaptive Moment Estimation (Adam), Stochastic Gradient Descent (SGD)

1. Introduction

Global research efforts have recently intensified their focus on improving the energy efficiency of existing structures [1]. Driven by mounting anxieties over pervasive energy waste and adverse environmental effects, the academic community is vigorously exploring diverse approaches to elevate building energy performance and diminish their ecological impact, given their substantial contribution to global energy usage and greenhouse gas emissions [2]. Accurate prediction of building energy consumption is crucial for the implementation of advanced energy solutions, including smart control systems [3], demand-side management protocols [4], and fault detection and diagnosis (FDD) techniques [5]. These methodologies are fundamentally dependent on predictive models to rationalize energy usage, curtail excess waste, and guarantee the seamless operation of building infrastructure. By pinpointing opportunities for energy conservation and rectifying operational shortfalls, these strategies directly facilitate optimal building performance [2]. Reliable energy consumption forecasts enable facility managers and occupants to make data-driven decisions and initiate preemptive measures, which may entail recalibrating Heating, Ventilation, and Air Conditioning (HVAC) operation, fine-tuning lighting timetables, utilizing high-efficiency appliances, and promoting energy-conscious behaviors [6]. Despite advancements, many studies report substantial divergence, with recorded energy consumption occasionally exceeding conventional predictions by up to threefold. Although conventional models rely on engineering principles, they are constrained in their ability to robustly account for the high complexity of real-world operating environments [7]. Although numerical simulation offers an alternative for consumption modeling, Artificial Intelligence (AI) methods present a more effective solution for optimizing and forecasting building energy use. These models deliver precise predictions and actionable intelligence for energy governance by processing large volumes of historical data, real-time sensor readings, and sophisticated Machine Learning (ML) algorithms [1]. Forecasting and classifying building energy consumption perform a vital role in improving decision-making related to energy use and reducing CO_2 emissions, as they enable the development of intelligent and sustainable design strategies [8]. Building energy prediction models are classified into physics-based (relying on thermodynamic equations and detailed structural data) and data-driven categories. Due to the significant challenge of acquiring specific operational and architectural data, physics-based models are often restricted. Therefore, data-driven methodologies, particularly those employing various architectures of Artificial Neural Networks (ANNs), have become the dominant approach for reliable building energy usage prediction [9, 10]. In this field, numerous studies have focused on data-driven approaches for energy consumption forecasting. Assadian et al. in their work on Data-Driven Modeling of Appliance Energy Usage, successfully employed feature engineering and hyperparameter tuning to improve the accuracy of machine learning models for electricity consumption prediction. Crucially, the set of machine learning models benchmarked in that study (including Random Forest, SVR, and XGBoost) aligns with the comparative models utilized in the current research, which reinforces the technical validity of the proposed MLP-NN's performance comparison [11]. Furthermore, Ghomeishi et al. optimized energy usage in a research building by integrating EnergyPlus simulation and a Multilayer Perceptron Neural Network (MLP-NN). By subsequently applying a Genetic Algorithm (GA), they achieved a significant 35% energy reduction. Sensitivity analysis confirmed that resident occupancy and wall insulation quality were the most influential variables on consumption [12]. Acknowledging the substantial electrical load of HVAC operations, Study [13] utilized an optimized MLP-NN trained on eight key architectural variables (e.g., floor area, wall/roof characteristics, and window area) to predict heating and cooling requirements. The Biogeography-based optimization (BBO) algorithm demonstrated superior predictive performance, achieving the highest R^2 and lowest error metrics (RMSE, MAE) across both phases. This resulted in improved prediction accuracy, exemplified by an R^2 of 0.920 for the heating load in the training phase. In another study, a Deep Neural Network (DNN) combined with an entropy-based optimization algorithm was employed for high-accuracy short-term electrical load forecasting, validated using PJM electricity market data from 2006 and 2016. The results demonstrated improved load forecasting for production and distribution planning [14]. Reference [15] presented a novel approach for medium- and short-term hourly load forecasting using Enhanced Random Artificial Forest (RAF) neural networks and the Grasshopper Jump Optimization Algorithm. By overcoming the local minima problem of the backpropagation algorithm, this method was successfully tested on the Dubai power grid, yielding effective daily load predictions. In [16], a model optimized with the Tree-structured Parzen Estimator (TPE) algorithm was used for electricity consumption forecasting, incorporating influential variables such as non-oil GDP, average temperature, holidays, and consumption trends. Results showed that including these variables improved prediction accuracy compared to models without them. Reference [17] proposed research established a comprehensive framework for data-driven building energy consumption (BEC) forecasting, focusing on four critical aspects: data collection, feature selection, model construction, and evaluation. Subsequent efforts, aimed at optimizing HVAC systems, successfully employed a Multilayer Perceptron Neural Network (MLP-NN) to investigate and accurately forecast building heating and cooling loads. Expanding on thermal load modeling, Study [18] focused on predicting heating and cooling demands for HVAC installations within Building Energy Management Systems (BEMS). The research utilized an MLP-NN and synthesized eight distinct metaheuristic algorithms to optimize its predictive operational parameters. Ultimately, the findings demonstrated that the Particle Swarm Optimization-Grey Wolf Optimizer (PSO-GWO) hybrid algorithm, when integrated with the MLP-NN, delivered the most accurate performance for thermal load forecasting.

In the present study, the primary focus is on achieving high-accuracy short-term (15-minute ahead) electricity consumption forecasting for residential buildings in Isfahan City using an optimized Multilayer Perceptron (MLP) neural network. The proposed approach incorporates data preprocessing with Min–Max Normalization, the integration of temporal features (hour of the day, holiday indicator, and season), early stopping during training to prevent overfitting, and feature selection based on importance analysis to enhance model performance and reduce dimensionality. The MLP model, optimized with Adam and Stochastic Gradient Descent algorithms, is benchmarked against several machine learning techniques to demonstrate its superiority. A key contribution of this research is the utilization of a comprehensive, up-to-date one-year dataset recorded at 15-minute intervals from the Isfahan Electricity Distribution Company, supplemented with meteorological data. This dataset captures current consumption patterns influenced by modern household appliances, HVAC systems, energy storage devices, and real climatic conditions, thereby enabling more accurate and realistic forecasting compared to studies relying on older or less specialized datasets.

2. Model and Methodology

This section outlines the methodology adopted for short-term electricity consumption forecasting using a Multilayer Perceptron Neural Network (MLP-NN). The workflow encompasses data acquisition and preprocessing, model architecture definition, training strategy, benchmark comparison with conventional machine learning algorithms, and a hybrid feature selection and engineering stage to optimize the input space. The dataset consists of one-year electricity consumption records at 15-minute intervals from the Isfahan Electricity Distribution Company, supplemented with meteorological data. Preprocessing included Min–Max Normalization and the incorporation of temporal features (hour of the day, holiday indicator, and season) to account for daily and seasonal variations. The MLP-NN was trained with Huber loss, early stopping to prevent overfitting, and a multi-stage learning rate schedule. Two model configurations were investigated: an initial version utilizing 9 input features (Section 6.1) and a refined version with 6 selected and engineered features (Section 6.2). Performance evaluation was conducted on a held-out test set against several machine learning baselines.

2-1 Architectural Design of the MLP Prediction Model

ANNs have garnered significant scholarly interest stemming from their capacity to model and execute complex pattern recognition and intelligent decision-making processes. This computational effectiveness makes them highly valuable for addressing sophisticated analytical challenges. A range of essential features distinguishes these types of networks. Among the core attributes are: effective learning capacity, parallel processing, robust stability, self-organization, high generalization potential, operational flexibility, and the capability for real-time execution. Consequently, the utility of these networks extends across a broad spectrum of applications, encompassing areas such as function approximation, complex pattern recognition, predictive forecasting, and system control. They therefore serve a critical function in advancing computational efficiency. MLP, which is characterized as a specialized architecture within the broader class of ANNs, is critically acclaimed for its superior capacity to model and precisely represent non-linear functional mappings. A key attribute of this network is its ability to effectively model systems characterized by intricate dynamics even in the absence of a predefined or explicit mathematical formulation [19]. Figure 1 illustrates the structure of a classical feedforward Multilayer Perceptron (MLP) neural network, comprising *I* input neurons ($g_1, g_2, \ldots, g_I$) and *N* predicted outputs ($y_1, y_2, \ldots, y_N$).

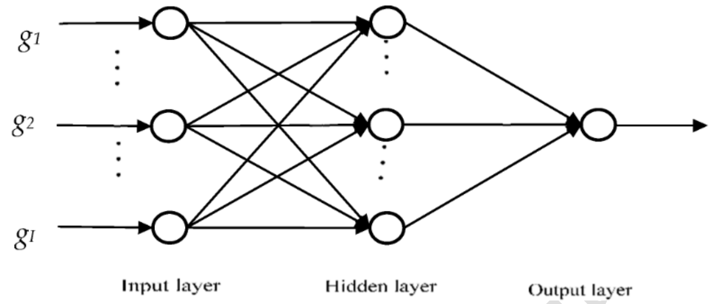

Figure 1. Architectural Representation of a Three-Layer Multilayer Perceptron (MLP) Network [19].

This model comprises *h* hidden layers, each containing K_h hidden neurons. It is designed to learn complex nonlinear relationships between the input data and the predicted outputs. Specifically, the weights connecting the input layer to the hidden layer are denoted by w_{ij}^1 while the weights connecting the hidden layer to the output layer are represented by w_{ij}^2. Additionally, C_j represents the threshold (bias) of the hidden neurons. Adjusting the weights and biases allows the network to learn the underlying relationship connecting the input dataset to the desired output feedback. Therefore, the output forecast of the MLP architecture, specifically for the *j-th* neuron at the *k-th* node, is mathematically represented by the following [20]:

$$\widehat{y_n}(t) = \sum_{j=1}^{k_h} w_j^2 \, F\left(\sum w_{ij}^1 \, g_i(t) + C_j\right) \tag{1}$$

For $1 \le n < m$, $1 \le j < k_h$, (w_j, $j = 0, 1, \ldots, k_h$), (w_{ij}, $i = 0, 1, \ldots, m$; $j = 0, 1, \ldots, k_h$)

The terms *m*, *h*, and K_h denote the number of input nodes, the count of hidden layers, and the quantity of hidden neurons per layer, respectively. The input connected to the *j-th* neuron of the hidden layer is represented by *i*. Within Equation (2), *F(a)* represents the Sigmoid transfer function, an exceedingly prevalent and vital function within Multilayer Perceptron (MLP) structures [20].

$$F(a) = \frac{1}{1+e^{-a}} \tag{2}$$

2-2 Optimization Algorithms

Within engineering domains, optimization algorithms perform a pivotal function by systematically working to either maximize or minimize a designated objective function or variable. These techniques are fundamentally utilized to ascertain the most favorable values for a given set of parameters. During this process, the input features effectively serve as control variables, while the corresponding output typically represents either the resultant controlled variable or the defined objective (cost) function. The primary goal is thus to isolate the combination of input values that produces the required output, which may range from performance maximization or cost minimization to the attainment of a specific predefined benchmark. Optimizers systematically explore for the optimal solution by executing iterative searches within a set framework of constraints, employing diverse mathematical methodologies and algorithms. Through successive modification of the input variables, these algorithmic approaches enable the pinpointing of the most suitable parameter configuration tailored for the task at hand. The final choice of an appropriate optimizer is consequently dependent upon the inherent nature and specific characteristics of the problem being addressed [18].

2-2-1 Stochastic Gradient Descent (SGD)

SGD is acknowledged as a pivotal and widely adopted optimization strategy for calibrating machine learning models, particularly within the deep learning domain. Where Batch Gradient Descent (BGD) necessitates calculating the gradient from the full training corpus, thereby rendering each update prohibitively costly and time-consuming for substantial data volumes, SGD offers a crucial alternative. It tackles this challenge by deriving a gradient estimate from either a single, randomly chosen data point or, more commonly, a mini-batch of randomly selected examples at every iteration. This fundamental difference using only a single instance or a small batch to estimate the gradient and subsequently adjust the parameters confers superior convergence speed and efficiency, an essential benefit for models trained on exceptionally massive datasets [21].

The main parameter update formula for SGD is expressed as follows:

$$W_{t+1} = w_t - \alpha \nabla \mathrm{w} \mathrm{J}(w_t) \tag{3}$$

Where:

- W_{t+1}: the updated weight parameter after performing one optimization step.
- w_t: The current weight (parameter at the present time step) on which the gradient is computed.
- α: The learning rate, a hyperparameter that controls the step size of each update. A larger α results in bigger changes to the weights, while a smaller α leads to smaller adjustments.
- $\nabla wJ(w_t)$: The gradient of the loss function with respect to the weight w at w_t. This gradient represents the slope of the loss function at that point and indicates the direction in which the loss increases most rapidly. Since the objective is to minimize the loss, the update is performed in the negative gradient direction.
- The new weight is determined by starting from the current weight and taking a step in the negative gradient direction (in the direction of decreasing the loss function) scaled by the learning rate. This

process is repeated iteratively until the loss function is minimized. However, the oscillatory updates of this optimization algorithm can make convergence challenging and may result in entrapment in local minima.

2-2-2 Adaptive Moment Estimation (Adam)

The Adaptive Moment Estimation (Adam) algorithm is acknowledged as a foremost and highly prevalent adaptive optimization technique. It successfully overcomes several inherent drawbacks of the SGD method, consequently leading to marked performance enhancement. Adam skillfully integrates the concepts of momentum and Root Mean Square Propagation (RMSProp) to facilitate the parameter-specific adjustment of the learning rate. This algorithm maintains two moving estimates of the gradients [21]:

- *m* (first moment): This involves an exponentially weighted moving average (EWMA) of the gradients, which mirrors the momentum technique. This average effectively serves to track the cumulative direction of the gradients throughout the training process.
- *v* (second moment): A second key component involves computing an exponentially weighted moving average (EWMA) of the squared gradients, which effectively quantifies the variability or magnitude of the gradient updates across iterations

The Adam update formula can be summarized in the following steps:

- Gradient computation:

$$g_t = \nabla_w J(w_t, mini - batch) \tag{4}$$

- Momentum update (*m*):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_1 \tag{5}$$

- RMSprop update (*v*):

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{6}$$

- Bias correction:

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t} \text{ and } \widehat{v_t} = \frac{v_t}{1-\beta_2^t} \tag{7}$$

- Parameter update:

$$w_{t+1} = w_t - \alpha \frac{\widehat{m_t}}{\delta + \sqrt{\widehat{v_t}}} \tag{8}$$

Here, $\beta 1$ and $\beta 2$ are the decay rate hyperparameters (typically 0.9 and 0.999), α is the primary learning rate, and δ is a small value added for numerical stability. As emphasized in [21], the Adam optimization algorithm has become one of the default and widely adopted optimizers in deep learning due to its ability to adaptively adjust learning rates and its robust performance across a wide range of problems.

2-3 Rectified Linear Unit (ReLU) Activation Function

The ReLU activation function is formally represented by the following mathematical expression:

$$ReLU(x) = \max\ (0, x) \tag{9}$$

The ReLU activation function outputs zero for negative input values and equals the input itself for positive values [23].

3. Proposed Model Based on Multilayer Perceptron (MLP)

The inputs of the MLP include various parameters such as the electricity consumption of HVAC systems, lighting, household appliances, the energy consumption of the energy storage system (5-kWh supercapacitor), ambient temperature in Isfahan, daily solar irradiance, hour, holiday and season in Isfahan. The datasets used in this study were obtained from the Isfahan Province Electricity Distribution Company, with 15-minute intervals over one year, and from the Iranian Meteorological Organization for ambient temperature data. To begin the data preprocessing stage, outliers were removed from the dataset using the Interquartile Range (IQR) method to ensure data quality and reduce the impact of anomalous observations [22]. In Figure 2, these inputs are used to predict the electricity consumption of a residential building. Initially, the Min-Max normalization function was independently applied to all features (inputs and outputs). This function linearly maps the values of each feature to the range of 0 to 1. After normalization, the predictive model becomes insensitive to the scale of individual features, thereby preventing scale-induced bias in the learning process. The formula for the Min-Max normalization function is presented in the following section [23].

$$X_{scale} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{9}$$

According to Equation (9), X_{scale} signifies the normalized value, whereas *min(x)* and *max(x)* represent the minimum and maximum bounds of the associated feature, in that order. Upon data normalization, the dataset was partitioned such that 20% of the entire data was randomly designated as the test set, and the remaining 80% was utilized to form the train set. A neural network with numerus neurons can end up memorizing the specific details and noise in the training data rather than capturing the underlying general patterns a problem known as overfitting [21]. By reducing the number of neurons in the subsequent layers, the model is compelled to learn only the most important patterns and avoid retaining unnecessary information. Such an approach significantly bolsters the model's generalization capability, enabling it to achieve enhanced predictive accuracy on data points it has not encountered during training [21]. The structure of the model, detailed in Figure 2, comprises two hidden layers that are fully connected. The first of these contains 128 units, while the second employs 64 units. A singular neuron is positioned in the final output layer, corresponding to the 15-minute interval prediction of the building's energy usage.

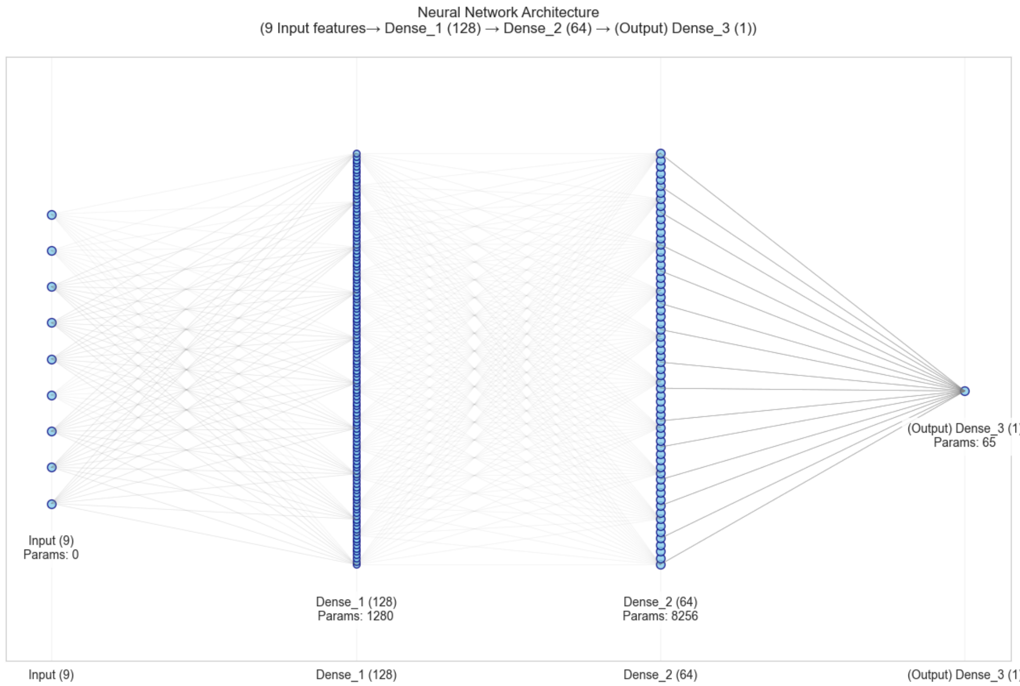

Figure 2. Architecture of the proposed Multilayer Perceptron Neural Network (MLP-NN) model. The network consists of an input layer with 9 (or 6 in the refined configurations) nodes, followed by two hidden layers with 128 and 64 neurons, respectively, both using ReLU activation functions, and a single-node output layer for predicting the next 15-minute electricity consumption. The model is trained with early stopping and a multi-stage learning rate schedule.

A summary of the specifications of the proposed model is presented in Table 1.

Table 1. Specifications of the Proposed Model

Layer Type	Layer Dimensions	Learnable Parameters
Input Layer	(None, 9)	0
Dense_1	(None, 128)	1,280
Dense_2	(None, 64)	8256
Output Layer	(None, 1)	65

For the non-linear transformation within the model, the proposed architecture incorporates the ReLU as the selected activation function. This final configuration resulted in a total count of 1,280 trainable parameters for the network. The following section details the complete implementation of the proposed model. According to [24], provides a summary of the formulas for the loss functions employed, namely Mean Absolute Error (MAE), Mean Squared Error (MSE),Root Mean Squared Error (RMSE), coefficient of determination (R^2) and Huber Loss.

Table 2. Mathematical Formulations of the Loss Functions

Loss Function Name	Formulas
MAE	$L_{MAE}(y,\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}\lvert y_i - \hat{y}_i\rvert$
MSE	$L_{MSE}(y,\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$
Huber	$L_{Huber}(y,\hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & if\ \lvert y-\hat{y}\rvert \leq \delta \\ \delta\lvert y - \hat{y}\rvert - \frac{1}{2}\delta^2 & if\ \lvert y-\hat{y}\rvert > \delta \end{cases}$ The hyperparameter δ establishes the crossover threshold from the loss function's quadratic to its linear regime.
RMSE	$RMSE(y,\hat{y}) = \sqrt{\frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{n}}$
R^2	$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y}_i)^2}$

A feedforward network undergoes training over several epochs, with each epoch consisting of numerous iterations, each of which processes an individual training batch. Specifically, when utilizing the Huber loss function, the error is computed by the Huber function immediately after the input signals propagate through the layers and the model generates the forecast. Using the threshold parameter (δ) it applies a quadratic function (MSE) for small differences between the prediction and the actual value to enhance optimization accuracy, and a linear function (MAE) for large differences to limit the influence of outliers (data points with a very low probability of occurrence). Finally, the Adam optimizer immediately uses the computed error gradients at the end of each batch to continuously adjust the weights and biases, thereby minimizing the overall model error [21].

4. Implementation

To implement the proposed model, Google Colab was utilized as a cloud-based environment. The Python programming language, along with the TensorFlow and Keras libraries, was employed to develop the deep learning model based on the MLP. Pandas was used to read data from Excel files, while TensorFlow and Keras functions facilitated the construction of the proposed model with loss functions including MAE, MSE, RMSE, R^2 and Huber, and optimizers such as SGD and Adam. Scikit-learn was employed for Min-Max normalization. To visualize the model's performance across epochs, plotting functions from the Matplotlib library were used. Additionally, the proposed model architecture was visualized using functions from Visual Keras, IPython, and Matplotlib.

5. Dataset

The dataset comprises 11 features and 34401 records, corresponding to 15-minute intervals from March 21, 2022, to March 20, 2023. A summary of the dataset is presented in Table 3.

Table 3. Characteristics of the Collected Dataset

Feature	Value
Total number of samples	34401
Number of features	11
Data type	Textual

Additionally, Table 4 summarizes the dataset features obtained from the Isfahan Electricity Distribution Company and the Iranian Meteorological Organization, along with their definitions and units of measurement.

Table 4. Values, Features, and Units in the Collected Dataset

Feature	Definition	Unit of Measurement
Date and time	Sampling date and time in the format: 3/21/2022 0:00	Date / Time
Lighting	Electricity consumption for lighting	Watt (W)
HVAC	Electricity consumption of HVAC systems	Watt (W)
Appliances	Electricity consumption of electrical and mechanical appliances for daily activities (e.g., refrigerator, TV, food processor, etc.)	Watt (W)
SOC (State of Charge)	Battery state of charge	Percent (%)
Outdoor Temperature	Outdoor temperature	Celsius (°C)
Solar Irradiance	Solar radiation received at ground level	Wh/m^2
Hour	Exact hour of the day	0-23
Holiday	1 = Holiday (incl. Fridays), 0 = Working day	0 or 1
Season	1=Spring, 2=Summer, 3=Autumn, 4=Winter	1–4
Active Energy	Electricity consumption of the building	Watt (W)

6. Results Analysis

6.1 Performance Evaluation Using 9 Input Features

Initially, the first 9 features listed in Table 4 were selected as input variables, while the last feature (Active Energy) was designated as the target output for the prediction task. To benchmark the performance of the proposed MLP-NN model, it was compared against 6 widely used machine learning regression algorithms: multiple linear regression, support vector regression (SVR) [25], random forest (RF) [26], gradient boosting (GB) [27], XGBoost (XGB) [28], and extra trees (ET) [29] with hyperparameters tuned by Random Search (10 iterations). The results obtained from training the proposed 2-layer Perceptron model on the collected proprietary dataset, using various hyperparameters including loss functions and optimizers, and trained over a total of 600 epochs, are presented in this section.

The proposed model was trained in three stages: Stage 1 with a learning rate of 0.005 for 100 epochs, Stage 2 with a learning rate of 0.003 for 100 epochs, and Stage 3 with a learning rate of 0.0005 for 400 epochs. The training was divided into distinct phases, each serving a specific purpose. According to the optimization principles outlined in [21], using a variable learning rate balances rapid exploration and precise exploitation of the parameter space. In the initial stages (Stages 1 and 2), relatively high learning rates (0.005 and 0.003) allow the network to traverse the parameter space quickly, escape early local minima, and identify a region near the global optimum. This phase functions as a high-speed search. In Stage 3, the learning rate is reduced to 0.0005 and the number of epochs increased to 400 to ensure precise convergence to the global minimum. The lower learning rate enables the model to take smaller steps, approach the optimum with high accuracy, minimize oscillations around it, and achieve the best possible performance. This strategy ensures that the model reaches a suitable solution rapidly while ultimately converging stably and accurately to the optimum. To mitigate overfitting and improve generalization, early stopping was implemented during the training phase. Table 5 presents the performance comparison of the proposed Multilayer Perceptron (MLP) neural network with two optimizers (Adam and SGD) against several benchmark machine learning models on the test set. The MLP regressor configured with a 128-64-1 architecture and the Adam optimizer achieved the highest predictive accuracy, recording the lowest MSE = 0.005947, RMSE = 0.07711, MAE = 0.01800, and Huber Loss =0.00297 among all evaluated models. It also yielded the highest R^2 = 0.82624, indicating that approximately 82.6% of the variance in the electricity consumption data was explained by the model. In contrast, the same MLP architecture trained with the SGD optimizer exhibited substantially inferior performance MSE = 0.020797, R^2 = 0.3924, highlighting the superiority of the Adam optimizer for this task. Among the benchmark models, Random Forest performed the closest to the Adam-optimized MLP, with an MSE = 0.00671, RMSE = 0.08194, MAE = 0.02323, and R^2 = 0.8038. The remaining models SVR, XGBoost, Gradient Boosting, Extra Trees, and Linear Regression showed progressively lower performance, with Linear Regression recording the poorest results MSE = 0.02554, R^2 = 0.2539. These results demonstrate the effectiveness of the proposed MLP-NN with Adam optimization in delivering superior short-term electricity consumption forecasts compared to both the SGD variant and the conventional machine learning baselines. The training and testing loss curves of the proposed MLP-NN optimized with Adam are shown in Figure 3. As illustrated, the model converges effectively, achieving the lowest test Huber loss of 0.00297 (consistent with the results reported in Table 5) with a corresponding training Huber loss of 0.00269.

Table 5. Performance comparison of the proposed MLP-NN and benchmark machine learning models using 9 input features on the test set.

Model	MSE	RMSE	MAE	R^2	Huber
MLP Regressor (128-64-1) with Adam	0.005947	0.07711	0.01800	0. 82624	0. 00297

MLP Regressor (128-64-1) with SGD	0.020797	0.144212	0.059736	0.392419	0.010399
Random Forest	0. 00671	0.08194	0.02323	0.8038	0.00336
SVR (RBF)	0.008372	0.09150	0.02069	0.7554	0.00419
XGBoost	0.00942	0.09705	0.03753	0.7248	0.00471
Gradient Boosting	0.01193	0.10924	0.04899	0.6514	0.00597
Extra Trees	0.02476	0.15736	0.07091	0.2766	0.01238
Linear Regression	0.02554	0.15981	0.08195	0.2539	0.01277

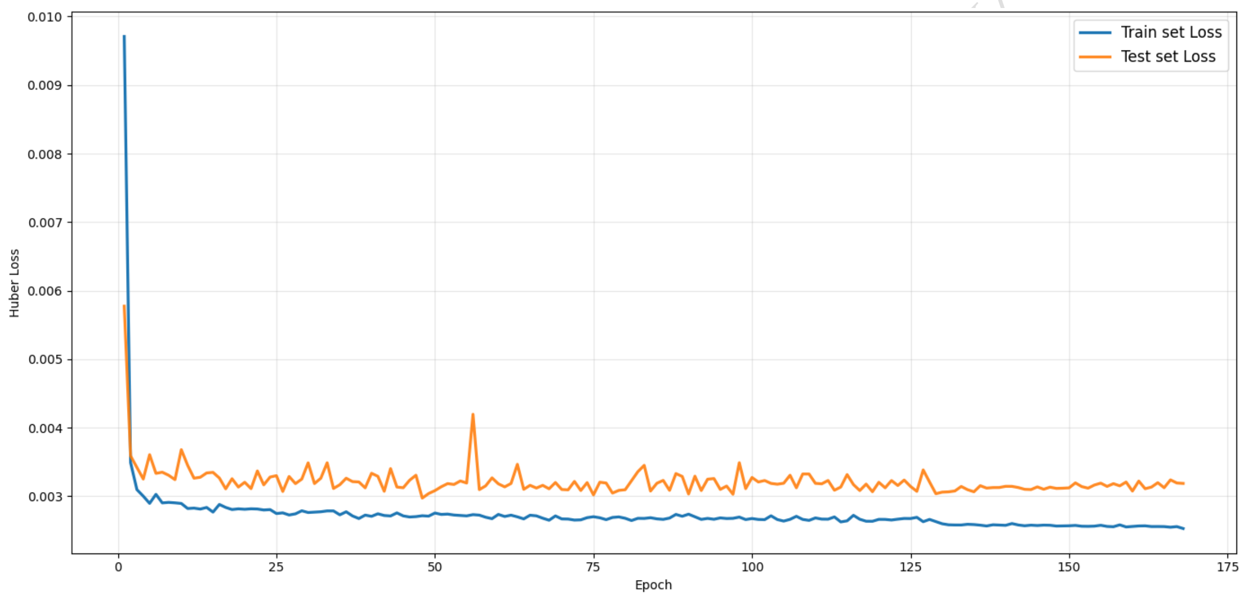

Figure 3. Training and testing Huber loss curves of the proposed MLP-NN optimized with Adam over epochs using 9 input features. The model was trained in three stages with decreasing learning rates (0.005, 0.003 and 0.0005) and batch sizes (64, 32 and 16), using Huber loss as the objective. Early stopping (patience = 48 epochs) and checkpointing were applied to prevent overfitting and retain the best weights, yielding the lowest test Huber loss of 0.00297 and a corresponding training Huber loss of 0.00269.

6.2 Performance with Correlation-Based Feature Selection

To further enhance model performance and reduce input dimensionality, a correlation-based feature selection process was conducted using Chatterjee's Xi correlation coefficient [30], which effectively captures both linear and non-linear dependencies. As shown in Figure 4, the analysis identified HVAC power consumption, household appliances, and lighting as the three most influential features (normalized importance ≈ 0.275 each), followed by hour of the day, SOC and season. The remaining features (solar irradiance, ambient temperature, and holiday indicator) exhibited negligible correlation with the target variable (active energy consumption).

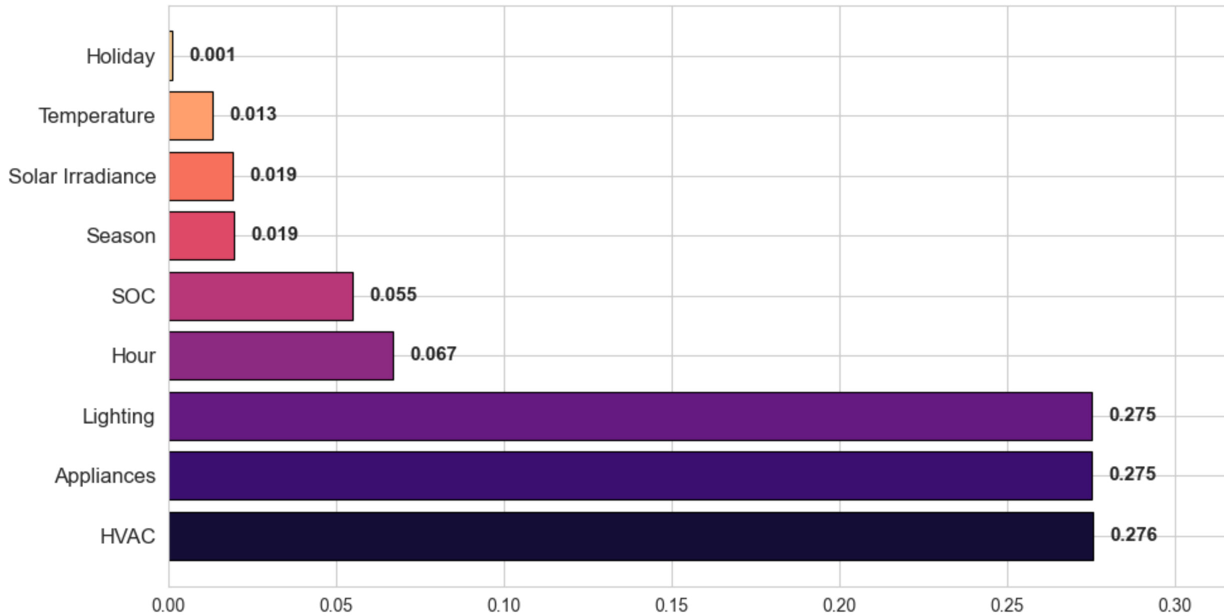

Figure 4 .Normalized feature importance scores based on Chatterjee's Xi correlation coefficient, computed on the scaled training dataset with respect to the target variable (active energy consumption). Features are ranked in descending order of importance; the top 6 features (HVAC, Appliances, Lighting, Hour, SOC, and Season) were selected for the refined model, while solar irradiance, temperature, and holiday indicator showed minimal correlation.

Guided by these results and supplemented with domain-specific feature engineering, a refined set of 6 input features was constructed:

- Total $_{\text{Sub-load}}$: The aggregated power consumption of controllable sub-loads, calculated as the sum of HVAC, appliances, and lighting.
- Hour $_{\text{sin}}$ and Hour $_{\text{cos}}$: To effectively represent the periodic daily patterns in electricity consumption. Cyclic encodings of the hour of the day, defined as:

$$\text{Hour}_{\text{sin}} = sin\left(\frac{2\pi \times \text{ Hour}}{24}\right)$$

$$\text{Hour}_{\text{cos}} = cos\left(\frac{2\pi \times \text{ Hour}}{24}\right)$$

- SOC: Energy stored in the supercapacitor (directly retained from the original features).
- Peak $_{\text{Hour}}$: Binary indicator for peak hours (set to 1 if Hour ∈ [31 18, 19, 20, 21, 22], otherwise 0). This feature was defined based on the observed hourly consumption patterns in figure 5, where the highest average power consumption occurs at hours 1 and 2 primarily due to scheduled supercapacitor charging during off-peak electricity pricing periods and elevated demand persists in the evening hours (17–22), reflecting typical high-demand residential activities. During peak pricing hours, the supercapacitor supplies energy to the load, reducing grid dependence. This energy arbitrage strategy leverages off-peak charging to lower electricity costs.

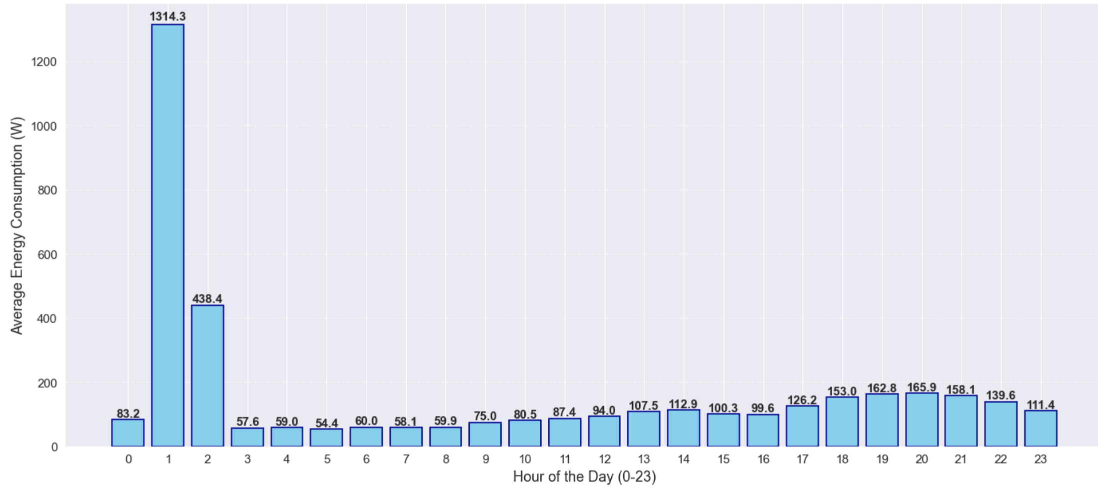

Figure 5. Average active power consumption (W) by hour of the day, sorted in descending order. The plot reveals significantly higher consumption during early morning hours (1–2 a.m.), attributed to scheduled supercapacitor charging under off-peak electricity pricing, and elevated demand in evening hours (17–22), corresponding to typical residential peak activities.

- Season: Categorical encoding of the season (spring = 1, summer = 2, autumn = 3, winter = 4), derived from the date to account for seasonal variations in electricity consumption patterns.

This hybrid approach combines data-driven insights from Xi correlation with physics-informed engineering to create a more compact and interpretable input representation.

The models were retrained using these 6 feature sets, maintaining the same MLP architecture (128-64-1), Adam optimizer, Huber loss, three-stage learning rate schedule, early stopping, and checkpointing strategy. Figure 6 presents the training and testing Huber loss curves for the refined model, demonstrating stable convergence.

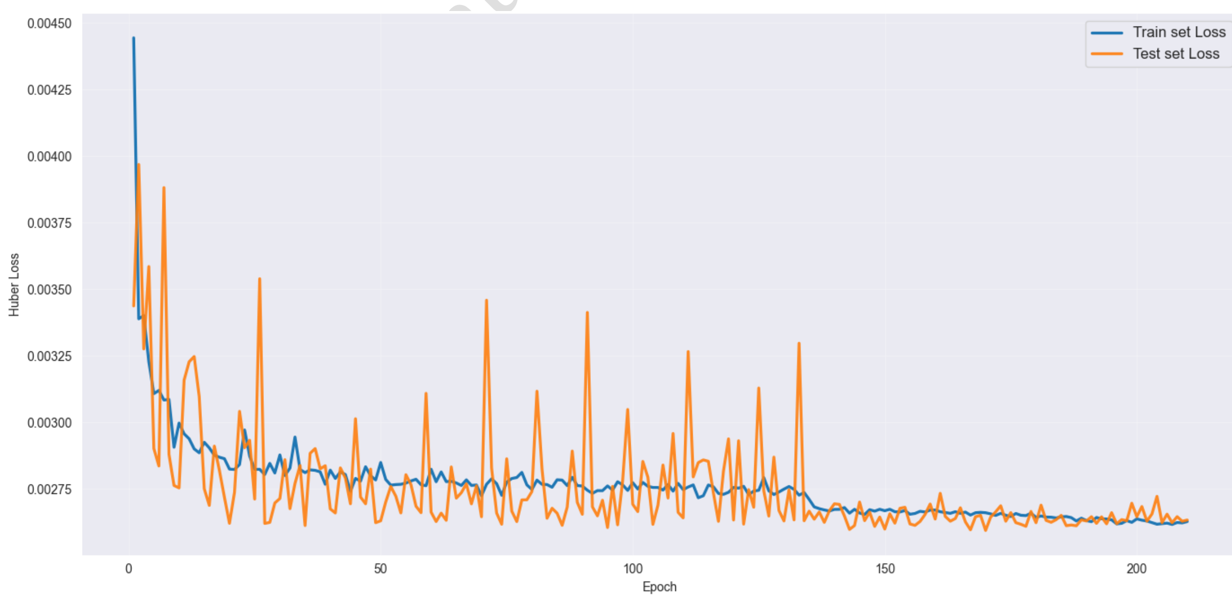

Figure 6. Training and testing Huber loss curves of the proposed MLP-NN optimized with Adam over epochs using the refined 6 feature sets (Total $_{\text{Sub-load}}$, Hour $_{\sin}$, Hour $_{\cos}$, SOC, Peak $_{\text{Hour}}$, and Season). The model

was trained in three stages with decreasing learning rates (0.005, 0.003, and 0.0005) and batch sizes (64, 32, and 16), using Huber loss as the objective. Early stopping (patience = 235 epochs) and checkpointing were applied to prevent overfitting and retain the best weights, yielding the lowest test Huber loss of 0.002594 and a corresponding training Huber loss of 0.002637.

Table 6 summarizes the performance metrics on the test set. The Adam-optimized MLP-NN achieved an MSE of 0.005187, RMSE of 0.072022, MAE of 0.014544, and R^2 of 0.780471—surpassing all benchmark models. Compared to the 6 feature configurations (Table 6), the reduced model exhibited lower error values across MSE, RMSE, and MAE, indicating higher prediction precision, albeit with a modest reduction in R^2. This improvement suggests that eliminating weakly correlated features reduced noise in the learning process, leading to better generalization and more robust short-term forecasts. Overall, the correlation-based feature selection combined with targeted engineering not only simplified the input space but also enhanced the predictive accuracy of the proposed MLP-NN, reinforcing its effectiveness for residential electricity consumption forecasting.

Table 6. Performance comparison of the proposed MLP-NN and benchmark machine learning models using the refined 6 feature sets (constructed through Chatterjee's Xi correlation-based selection and physics-informed feature engineering) on the test set.

Model	MSE	RMSE	MAE	R²	Huber
MLP Regressor (128-64-1) with Adam	0.005187	0.072022	0.014544	0.780471	0.002594
MLP Regressor (128-64-1) with SGD	0.007079	0.084137	0.028695	0.700404	0.003540
Random Forest	0.00556	0.07454	0.01976	0.76484	0.00278
SVR (RBF)	0.007456	0.086348	0.018022	0.684453	.003728
XGBoost	0.00716	0.08463	0.03428	0.69690	0.00358
Gradient Boosting	0.01105	0.10513	0.04913	0.53223	0.00553
Extra Trees	0.01682	0.12969	0.06005	0.28819	0.00841
Linear Regression	0.01850	0.13601	0.07933	0.21715	0.00925

The refined model with the 6 feature sets achieved lower error metrics compared to the baseline 9 feature configuration, particularly in MAE (19.2% improvement) and MSE (12.8% improvement), indicating enhanced prediction precision. Smaller improvements were observed in RMSE (6.6% improvement) and test Huber loss (12.7% improvement). The percentage improvement was calculated as equation 10:

$$Improvement(model\ , baseline\ )(\%) = \left(\frac{Loss_{baseline} - Loss_{model}}{Loss_{baseline}}\right) \times 100 \tag{10}$$

The selection of the Adam optimizer combined with the Huber loss function as the best-performing configuration in this experiment is justified by the following technical reasons.

- The Adam algorithm is an adaptive optimizer that adjusts the learning rate individually for each parameter. This feature enables it to converge to the minimum of the loss function faster compared to traditional optimizers such as SGD. Adam generally requires less manual tuning and often performs well across a wide range of problems. Due to the combined effects of its accelerated convergence rate and superior computational performance, Adam has been established as a standard and widely implemented optimization approach in the domain of neural network training [32].

- The Huber loss function treats small errors similarly to Mean Squared Error (MSE), providing differentiability and an appropriate gradient for fast convergence. However, for large errors, which may result from outliers (data points with low probability of occurrence), it behaves linearly, akin to Mean Absolute Error (MAE). One key advantage of the Huber loss is its combination of the strengths of both MSE and MAE. While MSE's high sensitivity to outliers can cause the model to focus excessively on these points, potentially degrading overall performance, MAE is more robust to outliers but may introduce challenges for optimizers due to the discontinuity of its derivative at zero. By penalizing large errors linearly and small errors quadratically, the Huber loss provides strong outlier resistance while maintaining a smooth, optimizable loss surface for small deviations [33].

Ultimately, the synergy between the advanced Adam optimizer and the robust, balanced characteristics of the Huber loss function enables the identification of an optimal point in the parameter space. This combination not only ensures efficient convergence but also effectively manages both large and small errors, resulting in a model that achieves minimal prediction error on unseen data. Such superior performance highlights the high suitability of this configuration for the nature of the problem and the dataset employed in this experiment.

7. Novelty and Contributions

The main novelties and contributions of this study are as follows:

1. Hybrid Feature Selection and Engineering: A combination of Chatterjee's Xi correlation coefficient for data-driven ranking and physics-informed engineering resulted in a refined 6 feature sets, improving model accuracy and interpretability while reducing input dimensionality.
2. Incorporation of Energy Arbitrage Behavior: The engineered features explicitly account for real-world supercapacitor charging during off-peak periods and peak-hour discharge under time-of-use pricing, enhancing the realism of the forecasting model in modern residential settings with energy storage.
3. Effective Use of Huber Loss with Adam Optimizer: The MLP-NN trained with Huber loss and a multi-stage Adam optimization schedule demonstrated superior robustness and precision compared to conventional loss functions and optimizers, achieving the lowest error metrics among all tested configurations.
4. Improved Performance through Dimensionality Reduction: The 6 features model outperformed the full 9 features baseline, with notable reductions in MAE (19.2%) and MSE (12.8%), highlighting the benefits of targeted feature selection for short-term load forecasting.
5. Application to a Contemporary Dataset: The study utilizes a recent, high-resolution (15-minute) dataset from Isfahan that includes modern appliances, HVAC systems, and energy storage under actual climatic and tariff conditions, providing a practical benchmark for similar regions.

These contributions advance the accuracy and practical applicability of neural network-based short-term electricity consumption forecasting in residential building energy management systems.

8. Conclusion and Future Work

This study proposed an optimized Multilayer Perceptron Neural Network (MLP-NN) for accurate short-term (15-minute ahead) electricity consumption forecasting in a residential building in Isfahan City, utilizing a high-resolution dataset incorporating modern appliances, HVAC systems, and a supercapacitor-based energy storage unit. The MLP-NN, trained with Huber loss and a multi-stage Adam optimization schedule combined with early stopping, consistently outperformed conventional machine learning benchmarks, including Random Forest, SVR, XGBoost, Gradient Boosting, Extra Trees, and Linear Regression. The integration of temporal features and careful preprocessing enabled robust performance with the full 9 feature sets. Through Chatterjee's Xi correlation-based feature selection and physics-informed engineering (including aggregated sub-loads, cyclic hour encodings, peak-hour indicator for energy arbitrage, and seasonal encoding), the input was reduced to 6 features. This refined model delivered the highest prediction precision, with significant error reductions over the 9 features baseline while preserving strong explanatory power. In particular, the refined 6 features MLP-NN model optimized with the Adam algorithm and employing the Huber loss function yielded the lowest error values among all configurations (test Huber loss = 0.002594), demonstrating substantial improvements in prediction accuracy over the 9 features baseline. These findings demonstrate that a hybrid data-driven and domain-knowledge-guided approach to feature engineering can simultaneously enhance accuracy, reduce model complexity, and improve practical relevance in residential BEMS. The proposed methodology offers a reliable and efficient framework for short-term load forecasting, particularly in settings with energy storage and time-of-use pricing, paving the way for more effective demand-side management and integration of renewable energy resources.

In line with the development and enhancement of the proposed model for forecasting electricity consumption in residential buildings in Isfahan, several avenues for future research can be envisioned to improve the model's accuracy, efficiency, and applicability.

Future research should aim to enhance the predictive capacity of the proposed MLP-NN based model. Employing advanced neural architectures such as Recurrent Neural Networks (RNNs) and LSTM models can improve the handling of temporal dependencies and irregular consumption patterns. The integration of deep learning with reinforcement learning approaches is also recommended to enable adaptive and real-time energy management strategies. Moreover, extending the forecasting horizon from short-term (15-minute intervals) to hourly or daily predictions would expand the model's applicability for both residential and grid-level energy optimization tasks.

In addition, enriching the input dataset with supplementary influential factors such as occupancy data, building insulation characteristics, socio-economic indicators, and dynamic electricity pricing could further improve model robustness and generalization. Future studies should also validate the model under diverse climatic and geographical conditions to ensure scalability and reliability. Finally, the practical deployment of the model within real-world Building Energy Management Systems (BEMS), supported by secure data processing and interactive visualization tools, is essential for assessing its operational feasibility and impact on sustainable energy use.

References

[1] Nakhaee Sharif, A., et al., Evaluating and identifying climatic design features in traditional Iranian architecture for energy saving (case study of residential architecture in northwest of Iran). Complexity, 2022. 2022(1): p. 3522883.

[2] Zhang, M., et al., A novel flexible grey multivariable model and its application in forecasting energy consumption in China. Energy, 2022. 239: p. 122441.

[3] Lima-Junior, F.R. and L.C.R. Carpinetti, Predicting supply chain performance based on SCOR® metrics and multilayer perceptron neural networks. International Journal of Production Economics, 2019. 212: p. 19-38.

[4] Runge, J. and R. Zmeureanu, Forecasting energy use in buildings using artificial neural networks: A review. Energies, 2019. 12(17): p. 3254.

[5] Zhong, H., et al., Vector field-based support vector regression for building energy consumption prediction. Applied Energy, 2019. 242: p. 403-414.

[6] Peng, L., et al., Effective energy consumption forecasting using empirical wavelet transform and long short-term memory. energy, 2022. 238: p. 121756.

[7] Khajavi, H. and A. Rastgoo, Improving the prediction of heating energy consumed at residential buildings using a combination of support vector regression and meta-heuristic algorithms. Energy, 2023. 272: p. 127069.

[8] Ye, Z., et al., Identifying critical building-oriented features in city-block-level building energy consumption: A data-driven machine learning approach. Applied Energy, 2021. 301: p. 117453.
[9] Fan, C., et al., Statistical investigations of transfer learning-based methodology for short-term building energy predictions. Applied Energy, 2020. 262: p. 114499.
[10] Saffari, M., et al., Passive cooling of buildings with phase change materials using whole-building energy simulation tools: A review. Renewable and Sustainable Energy Reviews, 2017. 80: p. 1239-1255.
[11] Assadian, C.F. and F. Assadian, Data-Driven Modeling of Appliance Energy Usage. Energies, 2023. 16(22): p. 7536.
[12] Ilbeigi, M., M. Ghomeishi, and A. Dehghanbanadaki, Prediction and optimization of energy consumption in an office building using artificial neural network and a genetic algorithm. Sustainable Cities and Society, 2020. 61: p. 102325.
[13] Xu, Y., F. Li, and A. Asgari, Prediction and optimization of heating and cooling loads in a residential building based on multi-layer perceptron neural network and different optimization algorithms. Energy, 2022. 240: p. 122692.
[14] KEYNIA, F. and G. MEMARZADEH, Short term electric load prediction based on deep neural network and wavelet transform and input selection. 2019.
[15] Ferdosian, M., et al., Mid-Term and Short-Term Load Forecasting Using Rough Neural Networks and Grasshopper Mutation. 2020.
[16] Sadeghi, S.H., Modeling and Forecasting Short-Term Electricity Consumption in Iran Using Neural Networks and the TPE Algorithm. 2024.
[17] Chen, G., et al., A Systematic Review of Building Energy Consumption Prediction: From Perspectives of Load Classification, Data-Driven Frameworks, and Future Directions. Applied Sciences, 2025. 15(6): p. 3086.
[18] Afzal, S., et al., Building energy consumption prediction using multilayer perceptron neural network-assisted models; comparison of different optimization algorithms. Energy, 2023. 282: p. 128446.
[19] Azadeh, A., S. Ghaderi, and S. Sohrabkhani, Forecasting electrical consumption by integration of neural network, time series and ANOVA. Applied mathematics and computation, 2007. 186(2): p. 1753-1761.
[20] Isabona, J., et al., Development of a multilayer perceptron neural network for optimal predictive modeling in urban microcellular radio environments. Applied Sciences, 2022. 12(11): p. 5713.
[21] Goodfellow, I., et al., Deep learning. Vol. 1. 2016: MIT press Cambridge.
[22] Vinutha, H., B. Poornima, and B. Sagar. Detection of outliers using interquartile range technique from intrusion dataset. in Information and decision sciences: Proceedings of the 6th international conference on ficta. 2018. Springer.
[23] Patro, S. and K.K. Sahu, Normalization: A preprocessing stage. arXiv preprint arXiv:1503.06462, 2015.
[24] Terven, J., et al., A comprehensive survey of loss functions and metrics in deep learning. Artificial Intelligence Review, 2025. 58(7): p. 195.
[25] Hearst, M.A., et al., Support vector machines. IEEE Intelligent Systems and their applications, 1998. 13(4): p. 18-28.
[26] Breiman, L., Random forests. Machine learning, 2001. 45(1): p. 5-32.
[27] Friedman, J.H., Greedy function approximation: a gradient boosting machine. Annals of statistics, 2001: p. 1189-1232.
[28] Chen, T., XGBoost: A Scalable Tree Boosting System. Cornell University, 2016.
[29] Geurts, P., D. Ernst, and L. Wehenkel, Extremely randomized trees. Machine learning, 2006. 63(1): p. 3-42.
[30] Xia, L., et al., The improved correlation coefficient of Chatterjee. Journal of Nonparametric Statistics, 2025. 37(2): p. 265-281.
[31] Al-Dahidi, S., et al., Ensemble approach of optimized artificial neural networks for solar photovoltaic power prediction. IEEE Access, 2019. 7: p. 81741-81758.
[32] Konur, O., D. Kingma, and J. Ba. Adam: A method for stochastic optimization. in Proc. Int. Conf. Learn. Represent.(ICLR). 2015.
[33] Tong, H., Functional linear regression with Huber loss. Journal of Complexity, 2023. 74: p. 101696.

