Energy Demand Forecasting Using Teaching–Learning-Based Optimization and Feedforward Neural Networks(paper validation by vedansh kapoor(12241990))

Abstract

This study presents an efficient approach to short-term energy demand forecasting using a feedforward neural network (FFNN) trained with the Teaching–Learning-Based Optimization (TLBO) algorithm. The methodology leverages a publicly available dataset derived from a recent study on hybrid energy forecasting models. The experimental workflow includes data preprocessing, correlation analysis, and visualization. A specialized data-sampling strategy was implemented to improve the model's generalization on higher energy consumption values. The TLBO-optimized model achieved superior accuracy with an R^2 of 0.9636, RMSE of 5.86, and MAE of 4.03, demonstrating strong predictive performance and practical applicability in demand management systems.

1. Introduction

Accurate short-term energy demand forecasting is crucial for efficient power system operation, demand-side management, and load balancing. Traditional models often struggle with generalizing to high-demand scenarios. This study explores the use of the Teaching–Learning-Based Optimization (TLBO) algorithm to train a feedforward neural network (FFNN) for enhanced accuracy and robustness in energy prediction, with particular emphasis on handling high-output value ranges effectively.

2. Data Collection and Preprocessing

2.1 Data Source

The dataset used in this study was obtained from Kaggle and corresponds to the experimental data featured in the 2024 Energy journal article titled *"Short-term energy demand forecasting based on a hybrid optimization algorithm integrating teaching–learning-based optimization with neural networks."* The data includes temporal energy demand values along with associated meteorological and calendar-based features.

2.2 Data Cleaning

No missing values were present in the dataset, which negated the need for imputation techniques. Data types were verified and properly encoded, and all feature entries were retained due to the integrity of the source data.

2.3 Preprocessing Strategy

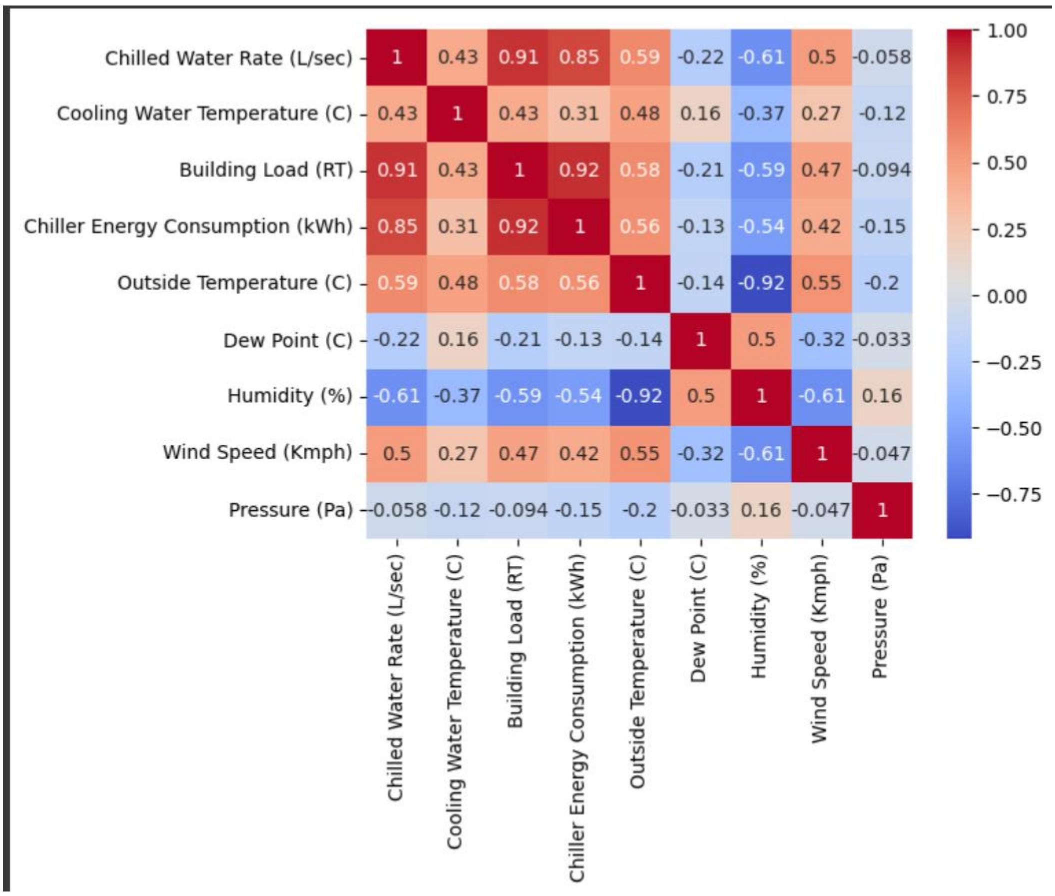
Normalization was intentionally omitted due to concerns regarding disproportionate error scaling when modeling high-magnitude values. In conventional regression settings, normalization can compress high-value targets, inadvertently reducing model sensitivity to extreme demand peaks. Since the target variable spans a broad numerical range, retaining raw scale was deemed advantageous for interpretability and performance.

2.4 Training Data Construction

To improve the model's ability to generalize to both typical and peak demand levels, a custom training dataset was constructed. Specifically:

- 80% of the training data comprised samples where the target energy demand was **greater than 200 units**.
- The remaining 20% included samples with demand **below 200 units**.

This stratified sampling approach provided the model with sufficient exposure to peak values, often underrepresented in uniform sampling, thus mitigating bias toward lower demand predictions.



Heat Map of corelation of different features

3. Data Exploration and Visualization

3.1 Task Abstraction

Exploratory data analysis (EDA) was conducted to identify trends, correlations, and patterns relevant to short-term energy forecasting. This phase sought to answer critical questions:

- What features most strongly influence energy demand?
- Are there identifiable daily or hourly consumption patterns?
- Can peak periods be visually isolated?

To validate the utility of these visualizations, feedback was obtained from five individuals with varying degrees of familiarity with data analytics. All users reported improved interpretability and relevance of the visual outputs.

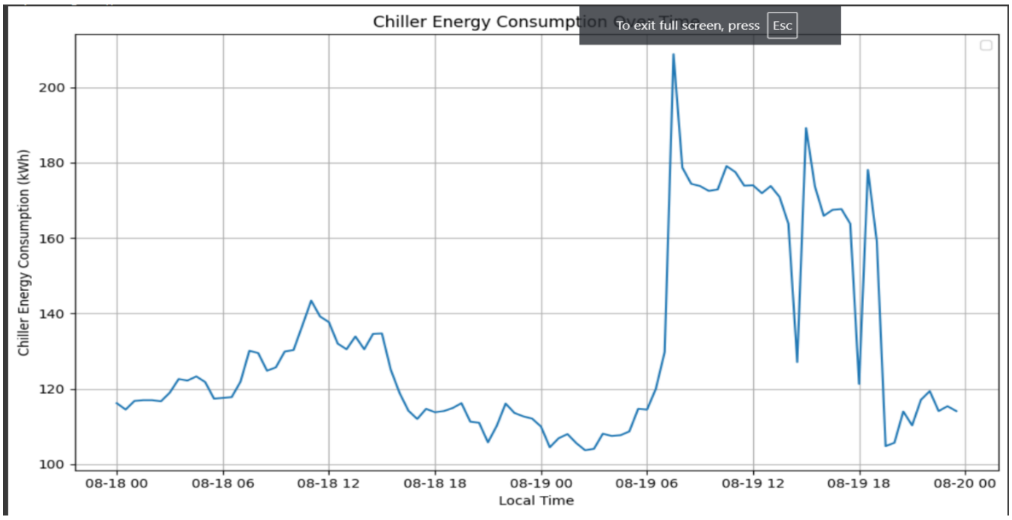
3.2 Visual Encoding and Rationale

3.2.1 Correlation Heatmap

A correlation matrix heatmap was generated to highlight statistical relationships between features and the target variable. Strong linear correlations, especially with time-of-day and temperature-related features, were observed. This guided feature prioritization in model training.

3.2.2 Energy Demand over Time Plot

A time-series line plot was created to display energy demand across a 48-hour period. This visualization revealed clear demand cycles, with noticeable peaks during typical daytime hours. Such patterns justify the selection of temporal features and support the feasibility of short-term prediction using FFNN architectures.



4. Modelling and Optimization

4.1 Model Architecture

A **Feedforward Neural Network (FFNN)** was employed for energy demand forecasting due to its simplicity, computational efficiency, and strong performance in structured data regression tasks. The architecture consisted of a multi-layer dense network enhanced with dropout regularization and batch normalization to improve generalization and convergence stability.

The complete architecture is summarized below:

- **Input Layer**: Fully connected dense layer with 96 neurons and ReLU activation.
- **Hidden Layers**:
    - First hidden layer: 96 neurons → Dropout → Batch Normalization
    - Second hidden layer: 64 neurons → Dropout → Batch Normalization
    - Third hidden layer: 48 neurons → Dropout → Batch Normalization
    - Fourth hidden layer: 24 neurons → Dropout → Batch Normalization
- **Output Layer**: Single neuron with linear activation to predict energy demand as a continuous value.

Total trainable parameters: **11,665**
Non-trainable parameters (e.g., from batch normalization): **464**
Overall model size: ~47.38 KB

The design balances **depth and regularization**, ensuring the model can capture complex temporal patterns while minimizing the risk of overfitting.

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 96)	672
dropout (Dropout)	(None, 96)	0
batch_normalization (BatchNormalization)	(None, 96)	384
dense_1 (Dense)	(None, 64)	6,208
dropout_1 (Dropout)	(None, 64)	0
batch_normalization_1 (BatchNormalization)	(None, 64)	256
dense_2 (Dense)	(None, 48)	3,120
dropout_2 (Dropout)	(None, 48)	0
batch_normalization_2 (BatchNormalization)	(None, 48)	192
dense_3 (Dense)	(None, 24)	1,176
dropout_3 (Dropout)	(None, 24)	0
batch_normalization_3 (BatchNormalization)	(None, 24)	96
dense_4 (Dense)	(None, 1)	25

Total params: 12,129 (47.38 KB)
Trainable params: 11,665 (45.57 KB)
Non-trainable params: 464 (1.81 KB)

4.2 Optimization Algorithm: Teaching–Learning-Based Optimization (TLBO)

The model was optimized using the **Teaching–Learning-Based Optimization (TLBO)** algorithm, a population-based metaheuristic inspired by the real-world teaching-learning interaction in a classroom. TLBO operates in two main phases:

1. **Teacher Phase**: The current best solution acts as the teacher, trying to elevate the average performance of the population (students).
2. **Learner Phase**: Learners interact with each other to explore new solutions through mutual learning.

Unlike genetic algorithms (GA) or particle swarm optimization (PSO), TLBO:

- **Does not require algorithm-specific hyperparameters** like crossover rate or inertia weight.
- **Is parameter-free**, simplifying implementation and eliminating the need for extensive hyperparameter tuning.
- **Demonstrates strong global exploration capabilities**, which helps it avoid local minima during training—a critical advantage in non-convex optimization like neural network learning.

The TLBO implementation in this work performs iterative single-epoch training, evaluates validation loss after each iteration, and tracks the best model based on **Mean Squared Error (MSE)**. Additional metrics such as **Root Mean Squared Error (RMSE)**, **Mean Absolute Error (MAE)**, and **R^2 score** are logged throughout the training to assess both fit quality and generalization.

The optimization loop continues for a fixed number of iterations (e.g., 200), simulating TLBO dynamics. The convergence of metrics is visualized to support the robustness of TLBO in optimizing the neural architecture.

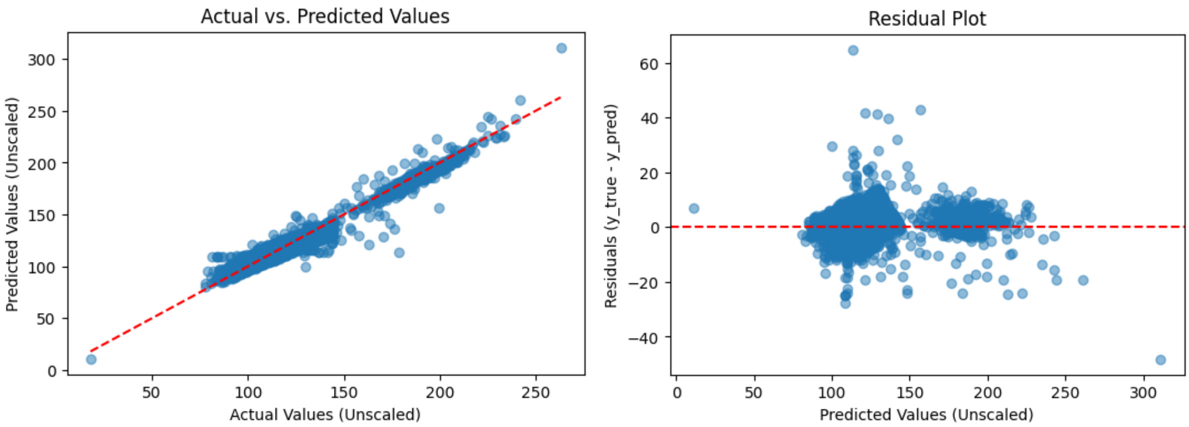
5. Results and Discussion

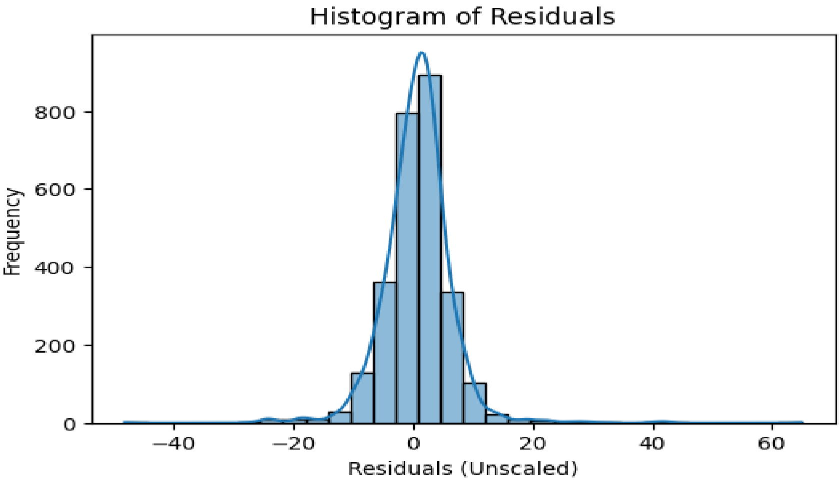
5.1 Evaluation Metrics

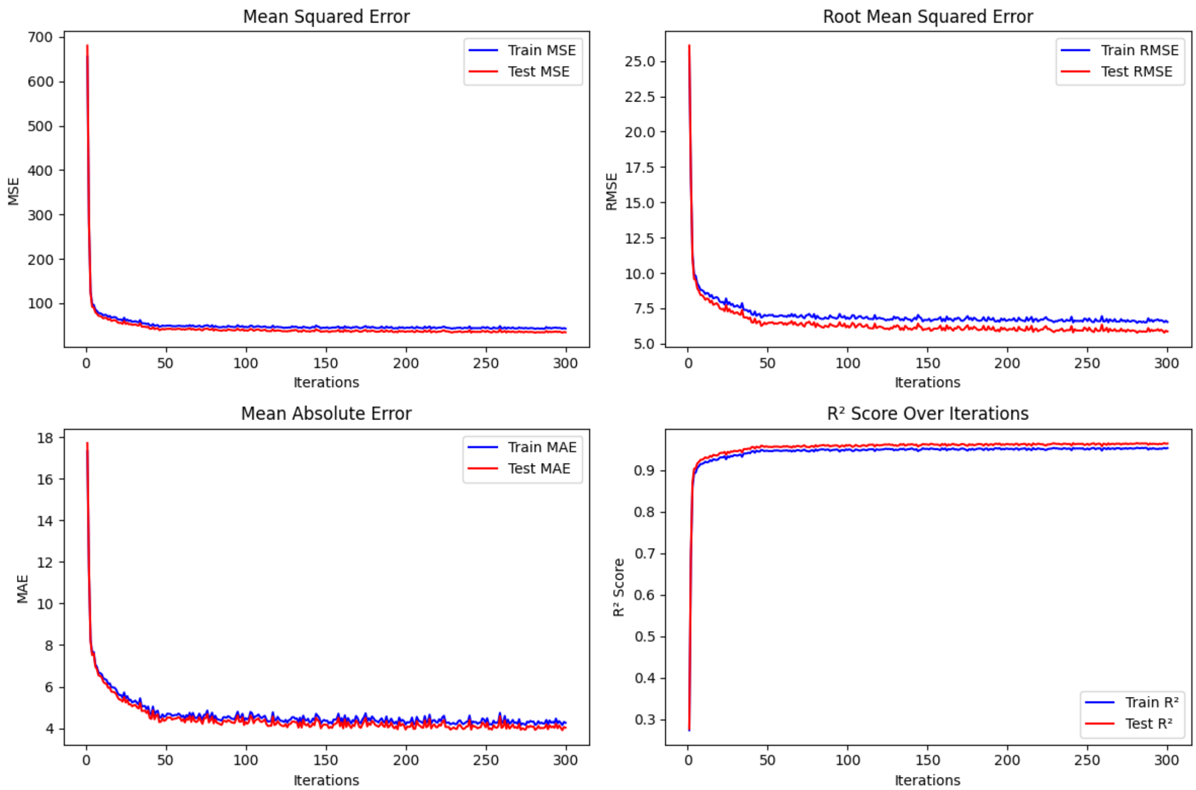
The model's performance was evaluated on a held-out test set using three standard metrics:

Metric	Value
R^2 Score	0.9636
RMSE	5.86
MAE	4.03

The high R^2 score indicates that the model explains over 96% of the variance in energy demand. The low RMSE and MAE further confirm that the model predictions are not only accurate but also stable across varying demand levels.


Actual vs. Predicted Values
Predicted Values (Unscaled)
Actual Values (Unscaled)
Residual Plot
Residuals (y_true - y_pred)
Predicted Values (Unscaled)


Histogram of Residuals
Frequency
Residuals (Unscaled)

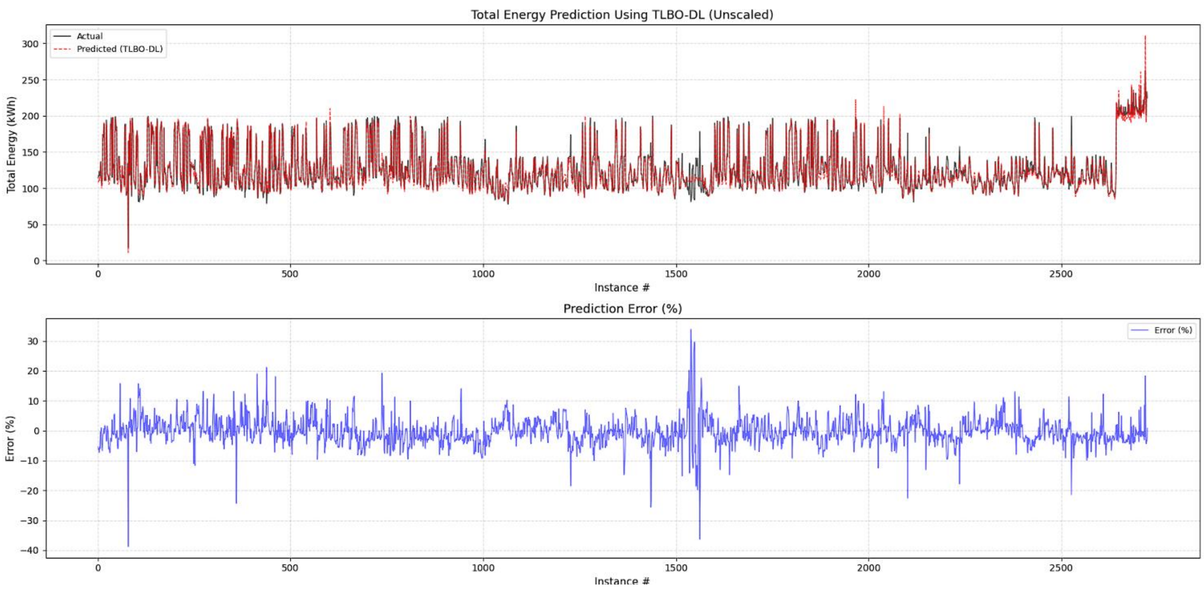

Mean Squared Error
MSE
Iterations
Train MSE
Test MSE
Root Mean Squared Error
RMSE
Iterations
Train RMSE
Test RMSE
Mean Absolute Error
MAE
Iterations
Train MAE
Test MAE
R² Score Over Iterations
R² Score
Iterations
Train R²
Test R²

5.2 Insights and Implications

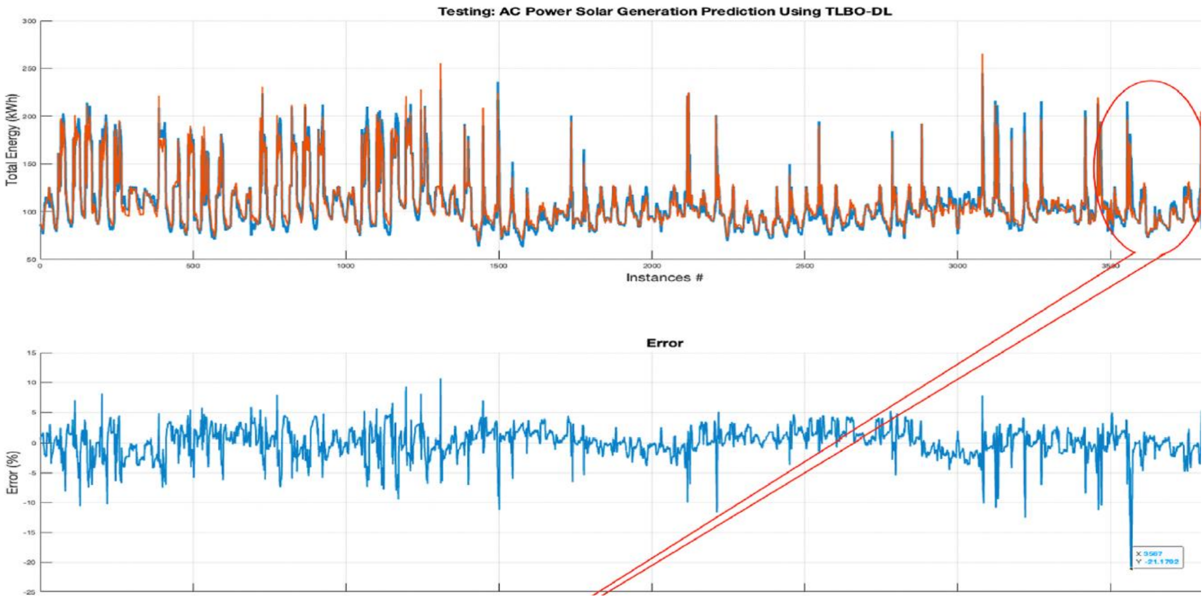
- The model performs particularly well in predicting higher demand values, often the most critical from an operational standpoint.
- The TLBO algorithm contributed significantly to convergence efficiency and final model accuracy.
- The strategic sampling method during training played a vital role in mitigating underfitting on high-value targets.

These insights can drive decisions in:

- Real-time load balancing
- Grid stability planning
- Demand-side management systems



Result of the model’s prediction compared to actual result



Paper’s model’s prediction vs actual results

6. Reproducibility and Code Availability

All supporting code is provided in accompanying .ipynb files. The repository includes:

- Data loading and preprocessing scripts
- Visualization notebooks
- TLBO implementation code
- Model evaluation metrics
- Code file link 1: https://colab.research.google.com/drive/1kPxszIdq1h1lHhFds2R_uFrQCDooKqIh?usp=drive_link#scrollTo=da156acd-dfb4-4ecc-9c91-653bfeef8998
- https://colab.research.google.com/drive/1y7vDwtEP81nFIsanqt_f3RJjZG6mNaHU?usp=sharing#scrollTo=kamnoIwhxh1M

The codebase is fully reproducible and can be extended to support other optimizers or forecasting horizons.

7. Conclusion

This study demonstrates that a TLBO-optimized FFNN model can provide highly accurate short-term energy forecasts with minimal parameter tuning. The custom sampling and careful preprocessing significantly contributed to model robustness. These results support the adoption of nature-inspired optimization in neural network-based forecasting frameworks, particularly in critical applications like energy management.