Machine Learning Assignment

PROJECT REPORT

TEAM ID : 23

Neural Architecture Search for Time Series Forecasting

Name	SRN
Diya D Bhat	PES2UG23CS183
Dhanya Prabhu	PES2UG23CS169

Problem Statement

With the rapid integration of renewable energy sources, smart grids generate vast amounts of real-time energy consumption and generation data. It's important to predict this data accurately to keep the power system stable, use resources efficiently, and handle energy needs properly. But creating the best neural network for predicting this kind of time-based data is challenging because there are so many possible designs and the model needs to work efficiently.

This project aims to automate the architecture search process using **Neural Architecture Search (NAS)** to identify the most effective model for forecasting household-level net energy balance in a **smart grid system**.

Objective / Aim

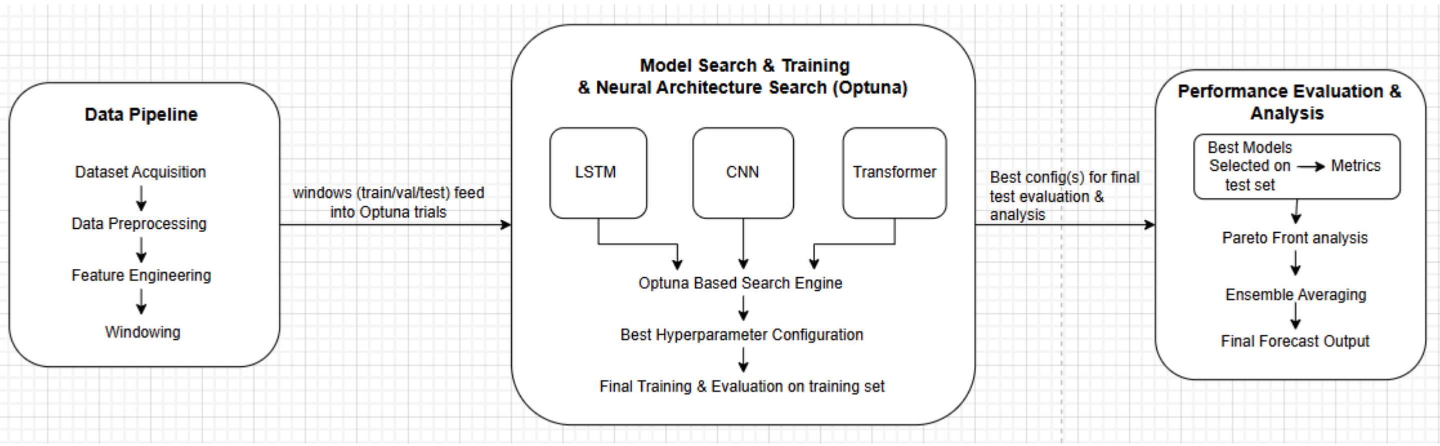
The objective of this project is to develop an **Optuna-based NAS framework** that automatically discovers the best-performing deep learning architecture for time series forecasting.
Specifically, the model aims to:

- Predict future **net energy balance** values based on historical smart grid data.
- Optimize model accuracy and computational efficiency simultaneously.
- Compare the performance of Optuna-based NAS with traditional hyperparameter optimization approaches such as Hyperband.

Dataset Details

- **Source:** Kaggle
- **Raw Dataset Size:** 153810 samples, 71 attributes.
- **Processed/Cleaned dataset size:**153810 samples, 23 attributes
- **Key Features:** Timestamp (15-minute intervals), Grid Import, Grid Export, Solar PV Generation, Net Energy Balance, Hour and Day-of-Week features, Sine/Cosine time encodings, and rolling statistical features (mean and standard deviation of PV values).
- **Target Variable:** Net Energy Balance (difference between household grid import and export, indicating net consumption or generation)

Architecture Diagram



Methodology

1. Data Pipeline

- The raw household smart grid dataset was first preprocessed to handle missing values, zero-target regions, normalization and for scaling.
- Dropped columns with **more than 50% missing data** (48 columns were removed).
- Feature engineering included time-based features (hour-of-day sine/cosine encoding) and rolling statistical features such as mean and standard deviation of solar PV generation.
- The time series was then segmented using a **sliding window approach** (96-timestep input windows, stride = 4) to prepare training, validation, and test samples.
- The data was split chronologically into 70% training, 15% validation, and 15% testing subsets to prevent information leakage.

2. Model Development

Initially, we tried doing the optimization using AutoKeras, but soon realised that AutoKeras version 2 did not support time series forecasting, and version 1 was incompatible with the new Python version that we were trying with. So, we switched to Optuna to perform a more flexible and customizable Neural Architecture Search (NAS), allowing precise control over hyperparameter ranges, objective functions, and evaluation strategies made specifically for time series forecasting.

Three neural forecasting architectures were developed:

- **LSTM (Long Short-Term Memory):** to capture long-range temporal dependencies,

- **CNN (Convolutional Neural Network):** to extract local temporal patterns efficiently,
- **Transformer:** to model global temporal relationships using self-attention.

Each architecture was implemented using TensorFlow/Keras and trained on the prepared windowed data.

3. Neural Architecture Search using Optuna & Hyperband

- An **Optuna-based Bayesian optimization framework** was used to perform automated neural architecture search (NAS).
- Optuna sampled hyperparameters such as number of layers, units per layer, dropout rates, learning rate, and batch size.
- Each trial trained a model and evaluated its validation Mean Absolute Error (MAE).
- The best hyperparameter configuration for each model type was selected based on the lowest validation error.
- The selected configurations were then retrained on the full training data for final evaluation.

Additionally, we also used **KerasTuner's Hyperband algorithm** as another method to tune and select the best model. In this approach, the model was trained for 30 epochs. Hyperband works by using early stopping to save time and it quickly stops trials that perform poorly and gives more resources to the better ones.

During our tests, we trained several model versions and tracked their **training and validation losses** to check how well they learned. We also compared their **actual vs predicted graphs**. Hyperband was able to find good model settings, but its results were a bit less consistent than those from **Optuna**. This showed that while Hyperband is faster, Optuna gives more reliable optimization results.

Hence, the Optuna-based NAS approach was ultimately chosen as the primary framework due to its greater flexibility, better control of search space, and ability to perform multi-objective optimization.

4. Model Evaluation and Pareto Optimization

The optimized models were evaluated on the test set using performance metrics such as **MAE, RMSE, R^2, and MAPE**.

To balance **accuracy and computational efficiency**, a **Pareto front analysis** was performed, plotting model error against parameter count to identify

Pareto-optimal models (i.e., models that achieved strong performance without excessive complexity).

5. Ensemble Averaging and Final Forecast

The top-performing Pareto-optimal models were combined using **ensemble averaging** to improve robustness and reduce variance in predictions.
This ensemble produced the final forecast of the **net energy balance** for future time steps, representing the predicted difference between grid import and export.

Results & Evaluation

Optuna Approach:

- The Optuna-based NAS approach effectively optimized LSTM, CNN, and Transformer models for energy balance forecasting.
- The **LSTM model** achieved the best performance with the lowest MAE and highest R^2 (~0.9998), showing strong predictive accuracy.
- A **Pareto analysis** confirmed its efficiency balance, and an **ensemble averaging** of models further improved stability and robustness of the final forecasts.

Trial values after training:

Best LSTM trial value (val MAE): 0.00026602589059621096
Best Conv trial value (val MAE): 0.0011673319386318326
Best Trans trial value (val MAE): 0.0013588154688477516

	model	mae	mse	rmse	r2	mape_pct	params	latency	name
0	LSTM	0.000556	3.946977e-07	0.000628	0.999875	5.365874	68225	0.052521	LSTM
1	Conv	0.001495	2.884721e-06	0.001698	0.999087	2.917985	265473	0.054067	Conv
2	Trans	0.003875	2.333804e-05	0.004831	0.992610	16.861804	2245	0.052835	Trans

Fig 1: Final Results after Training (15 rounds)

```
Best Performing Model Summary:
Model: LSTM
MAE: 0.000556
RMSE: 0.000628
R²: 0.999875
MAPE: 5.366%
Params: 68225, Latency: 0.052521s
```

Fig 2: Optuna Based NAS

Pareto-optimal models (best trade-offs between accuracy and complexity):

	model	mae	mse	rmse	r2	mape_pct	params	latency	name	pareto
0	LSTM	0.000556	3.946977e-07	0.000628	0.999875	5.365874	68225	0.052521	LSTM	True
2	Trans	0.003875	2.333804e-05	0.004831	0.992610	16.861804	2245	0.052835	Trans	True

Fig 3: Pareto Analysis Results

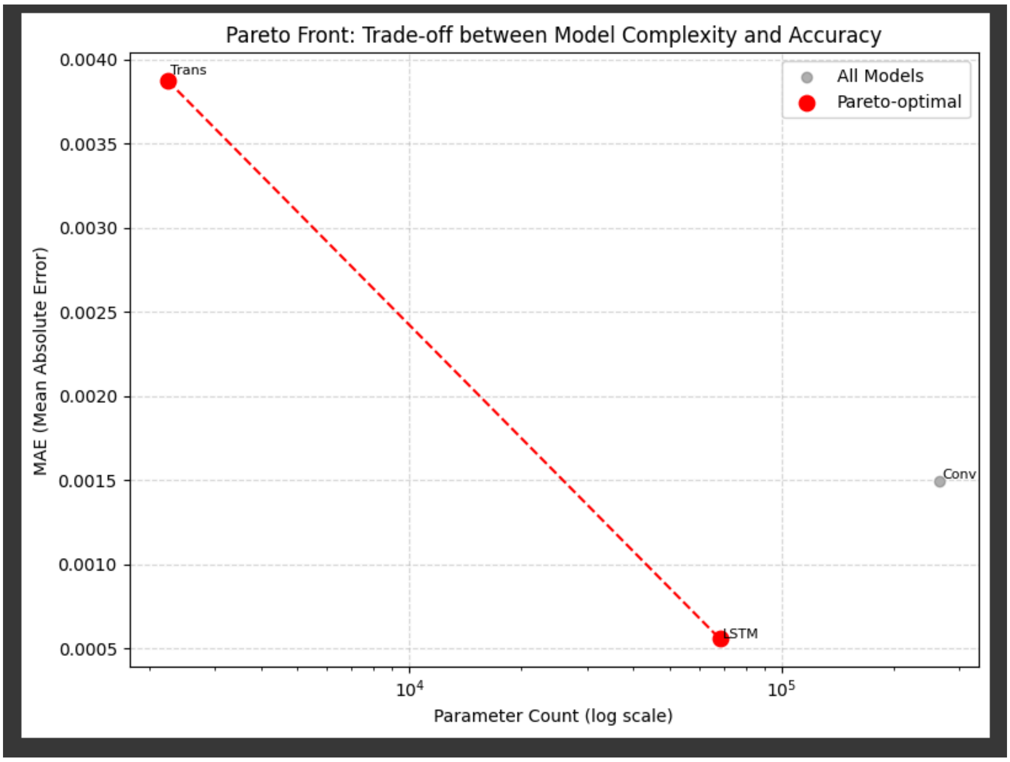


Fig 4: Pareto Plot

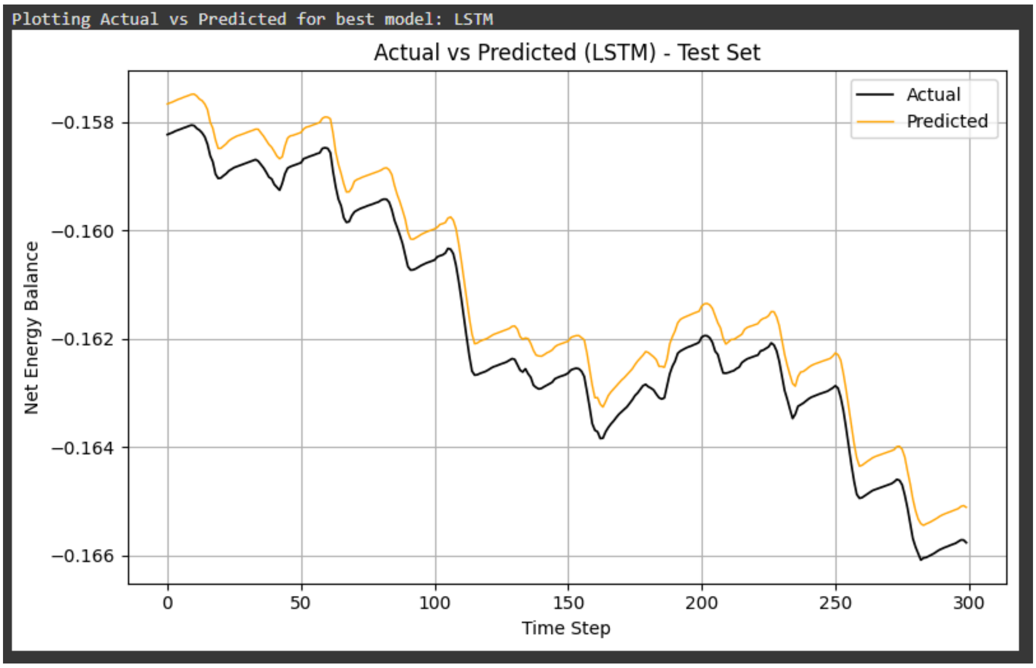


Fig 5: Actual vs Predicted Plot for LSTM

HyperBand Approach:

- The Hyperband-based Neural Architecture Search (NAS) efficiently tuned and compared CNN, LSTM, and GRU models for energy balance forecasting.
- Among them, the **Transformer model** achieved the best overall performance, demonstrating the lowest validation loss, indicating excellent predictive accuracy.
- Hyperband's early-stopping strategy ensured faster convergence by pruning underperforming trials, optimizing training time without compromising accuracy.

```
Trial 13 Complete [00h 01m 03s]
val_mae: 0.18868233263492584

Best val_mae So Far: 0.16899573802947998
Total elapsed time: 00h 09m 02s
413/413 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step
Transformer model best MAE: 0.1867
Best hyperparameters:
  num_heads: 2
  ff_dim: 96
  dropout_rate: 0.1
  tuner/epochs: 10
  tuner/initial_epoch: 0
  tuner/bracket: 0
  tuner/round: 0

Best overall model: Transformer (MAE = 0.1867)
```

Fig 6: Best model found after training

```
Best Hyperparameters Found:
  num_heads: 2
  ff_dim: 96
  dropout_rate: 0.1
  tuner/epochs: 10
  tuner/initial_epoch: 0
  tuner/bracket: 0
  tuner/round: 0
```

Fig 7: Best hyperparameters w.r.t. to best model

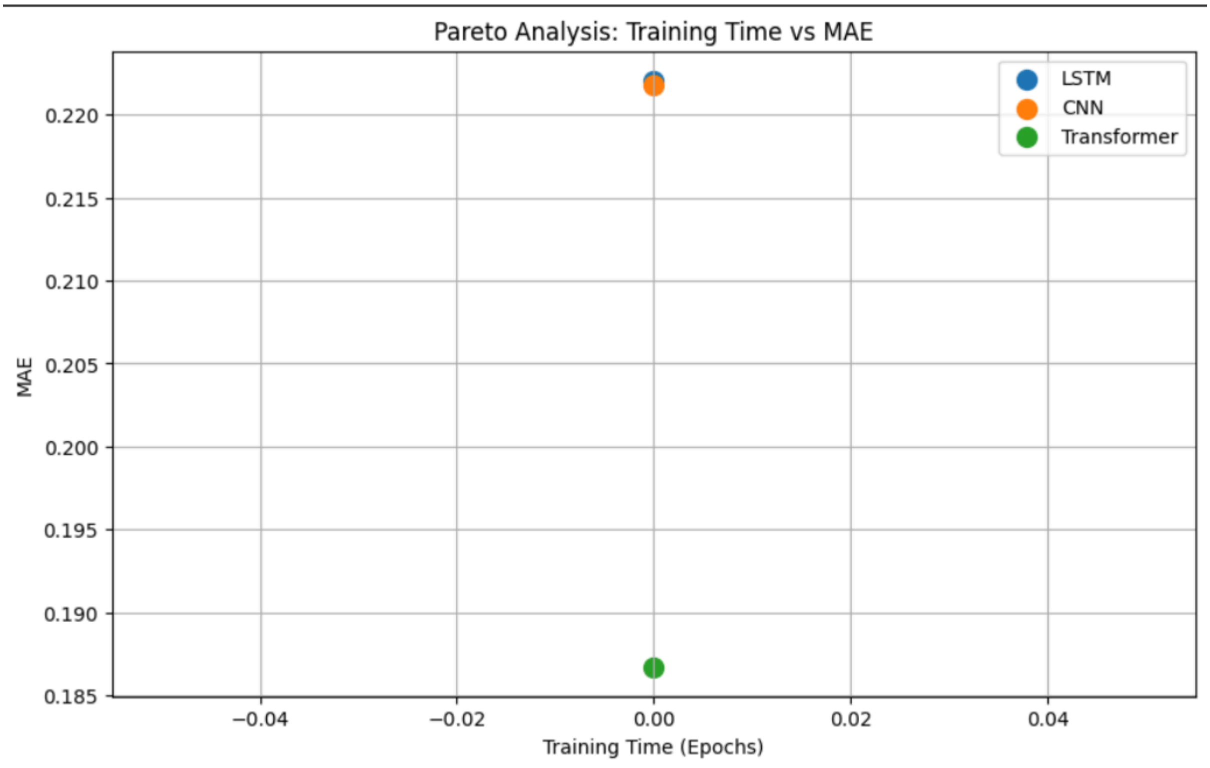


Fig 8: Pareto Analysis

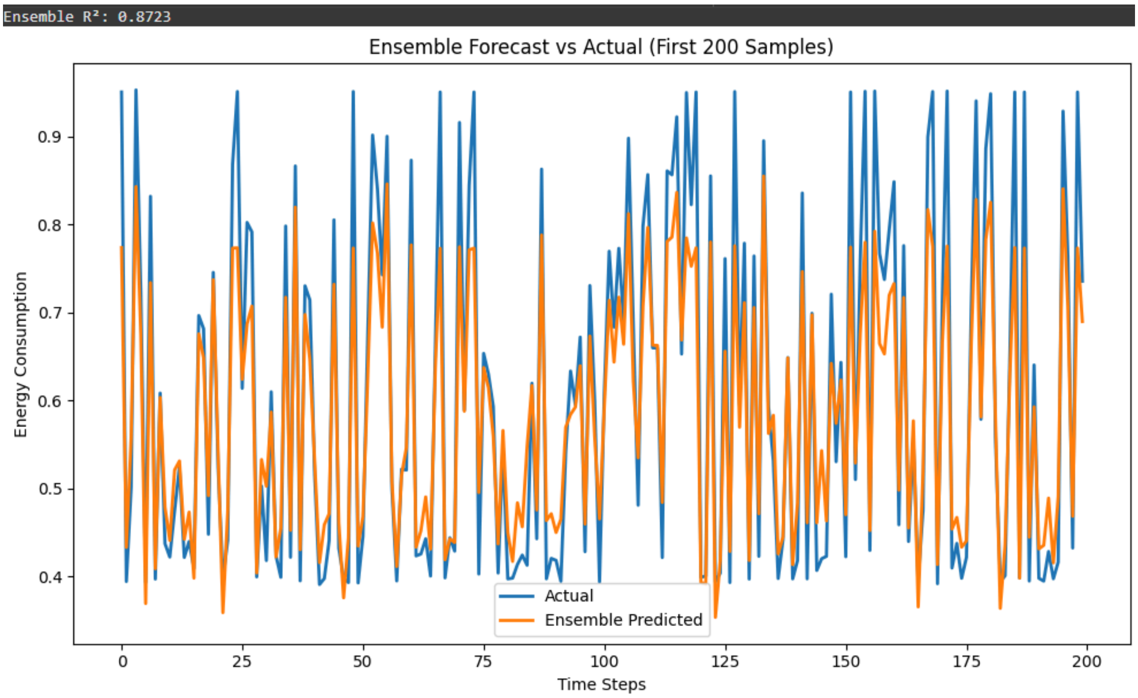


Fig 9: Ensemble Forecast vs Actual graph

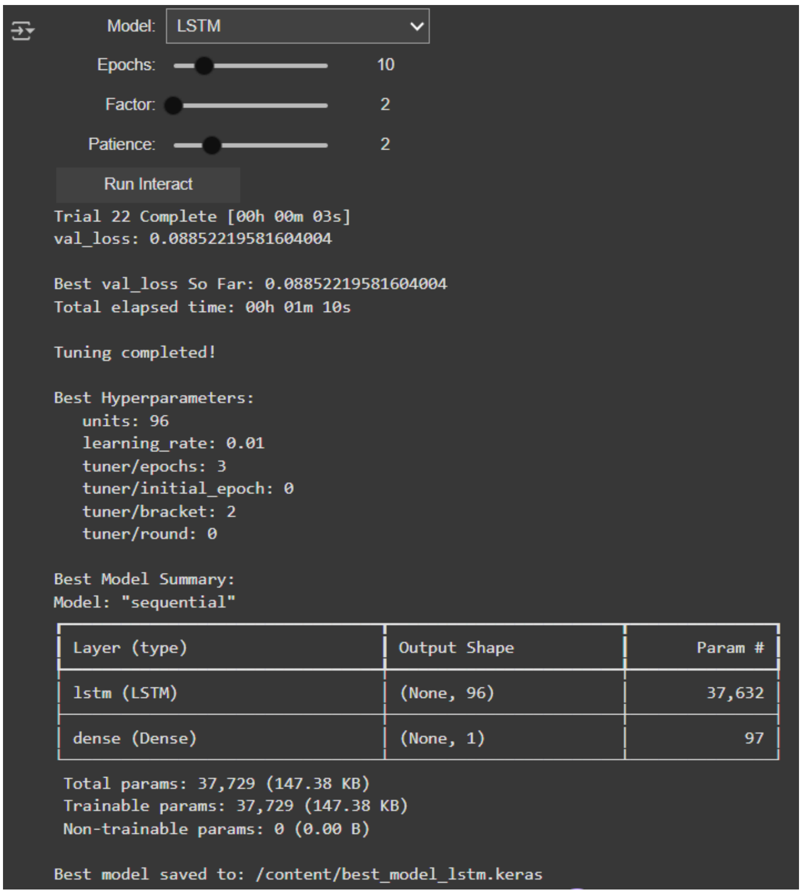


Fig 10: UI

Conclusion

This project successfully developed an automated **Neural Architecture Search (NAS)** framework, powered by **Optuna & Hyperband**, to identify the most effective deep learning model for forecasting **household-level net energy balance** within a smart grid system.

Key Achievements and Findings

- **Effective Model Optimization:** The Optuna-based NAS framework was highly effective in optimizing three distinct architectures: **LSTM**, **CNN**, and **Transformer**, by exploring a wide search space of hyperparameters.
- **High Forecast Accuracy:** The LSTM model achieved the best performance with an **R^2 of 0.9998**, indicating excellent fit and minimal forecasting error.
- **Computational Efficiency:** Pareto front analysis identified models that balanced accuracy and parameter count, enabling better model selection based on trade-offs between performance and complexity.
- **Robustness through Ensembling:** An ensemble averaging of top-performing models further stabilized predictions and improved generalization on unseen data.

Key Learnings

- **Data Preprocessing and Feature Engineering:** The thorough data pipeline, including handling missing values, scaling, creating time-based features, and using a sliding window approach for segmentation, was fundamental to preparing the complex smart grid data for training deep learning models.
- **Optuna's Flexibility:** Switching from AutoKeras to an Optuna-based NAS was a critical learning point, as Optuna provided the necessary flexibility and control over the search space, objective functions, and evaluation strategies specifically tailored for time series forecasting, which was not available in the attempted version of AutoKeras.
- **Importance of Multi-Objective Optimization:** The project highlighted the necessity of multi-objective optimization beyond just error metrics. Plotting model error against parameter count in the Pareto analysis was vital for selecting models that are both accurate and practical for deployment in a real-world smart grid system.

References:

1. Multi-Objective Neural Architecture Search by Learning Search Space Partitions (2024)
2. Automatic selection of the best neural architecture for time series forecasting via multi-objective optimization and Pareto optimality conditions (2025)
3. Learning the Pareto Front with Hypernetworks (2021)
4. Optuna: A Next-generation Hyperparameter Optimization Framework
5. Efficient Multi-Objective Optimization for Deep Learning