Università della Svizzera italiana

Evaluation of GPU Accelerated Machine Learning Algorithms for Energy Price Prediction

Naga Venkata Sai Jitin Jami, Juraj Kardoš, Olaf Schenk

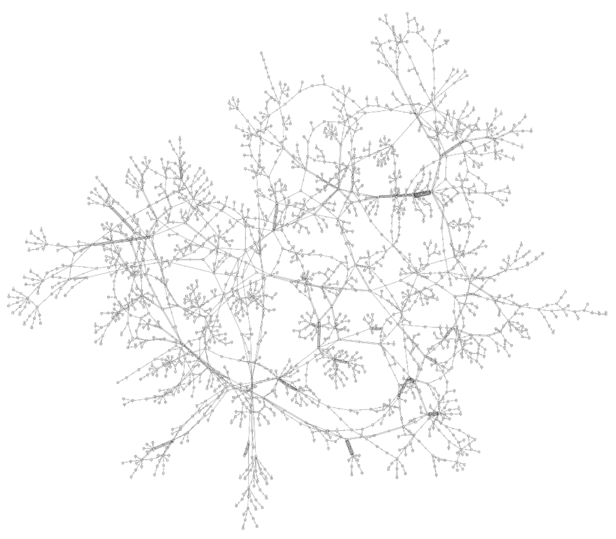
jitin.jami@usi.ch, juraj.kardos@usi.ch, olaf.schenk@usi.ch

Locational Marginal Pricing

LMP represents the minimum cost to supply additional load from electricity generators at a specific node. Accurate forecasting of LMPs is, therefore, essential for market participants, such as balancing and flexibility service providers, to optimize the scheduled operation and bidding strategy.

DC-OPF Problem formulation

$$
\begin{aligned}
\text{minimize} \; & \sum_{i \in \mathbf{G}} C_i(P_i^G) \\
\text{subject to} \; & P_i(\delta) = P_i^G - P_i^L, \forall i \in \mathbf{N}, \\
& P_i^{G min} \leq P_i^G \leq P_i^{G max}, \forall i \in \mathbf{G}, \\
& \delta_i^{min} \leq \delta_i \leq \delta_i^{max}, \forall i \in \mathbf{N}, \\
& h_f(\delta) \leq 0, \\
& h_t(\delta) \leq 0
\end{aligned}
$$



- Regular power systems are generally large and complex, thus computing LMP becomes prohibitively expensive.
- The increasing penetration of renewable energy sources increases the volatility and unpredictability of electricity prices.
- To address these challenges, machine learning tools can be leveraged to predict LMP with much less computational effort and time.

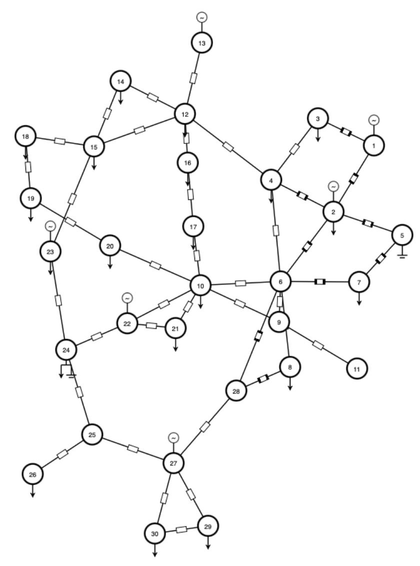
PGLib-OPF, MATPOWER and MOSEK

PGLib-OPF is a benchmark library curated and designed to evaluate a well-established version of the AC-OPF or DC-OPF problem formulations of 60 standard electricity grids sourced from various literature. The choice of electricity grids and their descriptions are stated below.

Description of electricity grids under consideration

Case	# load buses	# generator buses	# branches
case30[a]	30	6	41
case240	240	143	448
case1354	1354	260	1991
case1888[b]	1888	296	2531

- The PGLib-OPF models contain static data, i.e., a snapshot of the grid state in a single time instance.
- They consist data like individual voltage levels across nodes, grid topology, power injection and power withdrawal.
- This single snapshot translates into a single DC-OPF optimization problem imported into MATPOWER.
- The DC-OPF formulation is then solved using an optimization solver like MOSEK.



[a] The grid shown on the right
[b] The grid shown on the left

Data Generation and ML Models

The process followed to generate data is mention in the flowchart below. The nodal level perturbations s_{P_d} are calculated using global and local scaling factor s_{grid} and s_{nodal} respectively.

$$s_{P_d} = 1 + \frac{s_{grid} \times s_{nodal}}{100}$$

Features extracted from grid data:

- Active power demands P_d at each load bus.
- Transmission capacity factor P_l is defined as:

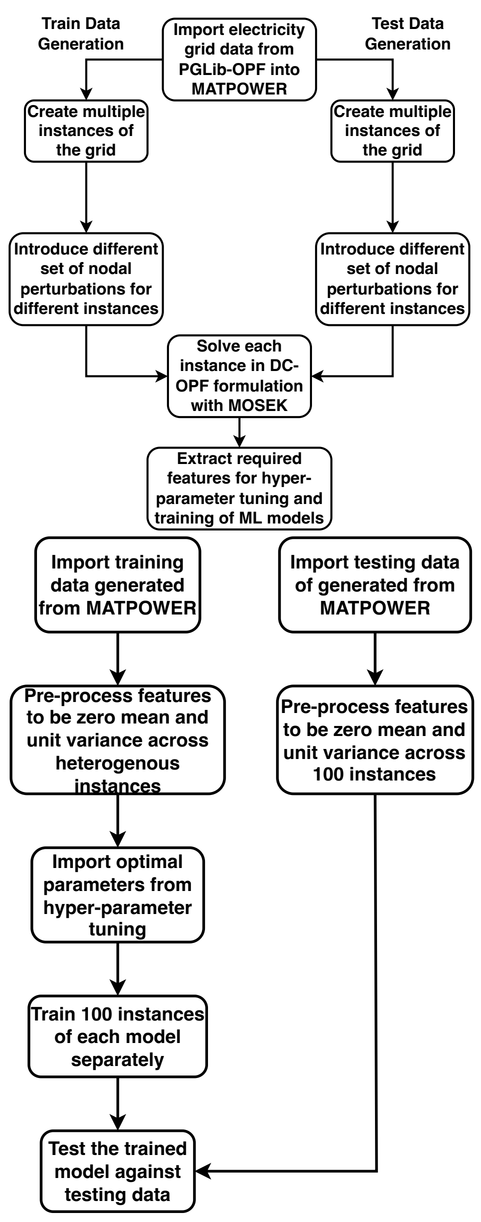
$$P_l = \frac{P_d}{P_l^{max}}$$

Hyper-parameters tuning for all our ML models were conducted using bayesian optimisation using a subset of the training data to decide the hyper-parameters used for the experiments. The choice of ML models for the study are:

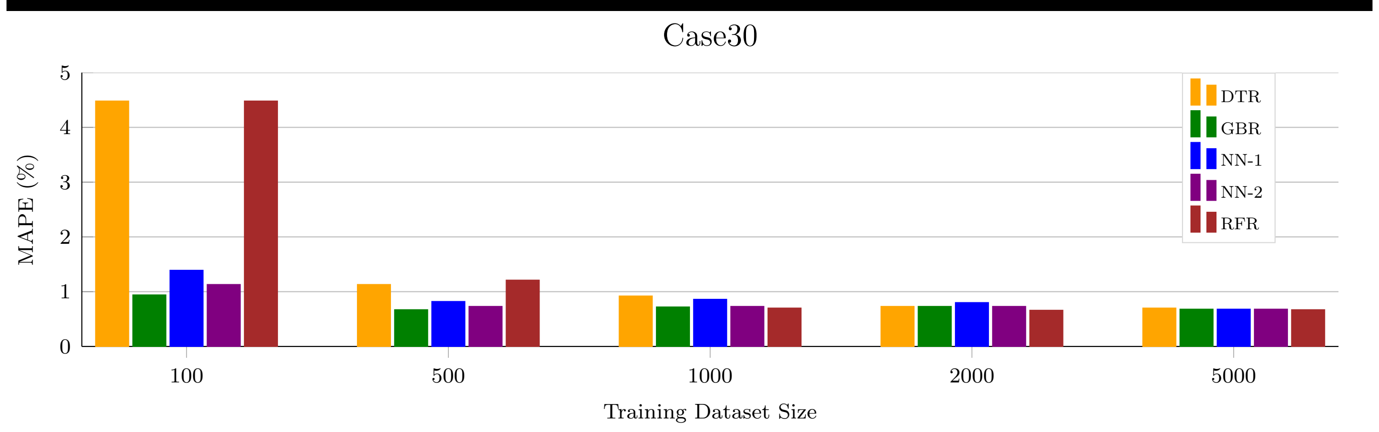
1. Decision Tree Regression (DTR),
2. Gradient Boosting Regression (GBR),
3. Random Forest Regression (RFR),
4. Deep Neural Network with multiple hidden layers (NN - 1 & 2).

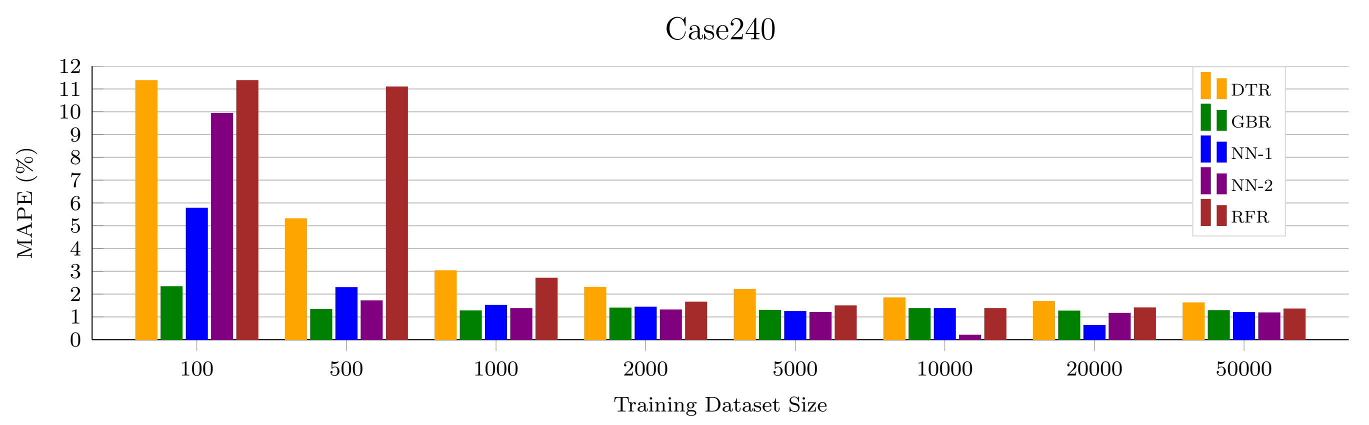
In each experiment, 100 instances of the ML models were trained on the same training dataset. The average of result metrics (accuracy) from 100 instances is considered. The result metric is:

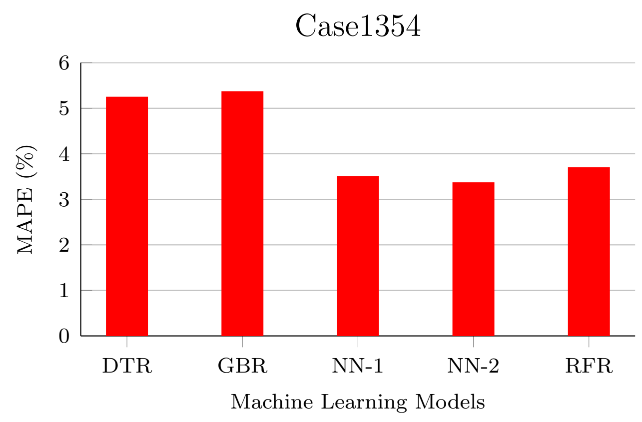
$$MAPE(\bar{y}, \hat{y}) = \frac{100}{NS} \sum_{i=0}^{NS-1} \frac{|\bar{y}_i - \hat{y}_i|}{|\bar{y}_i|},$$

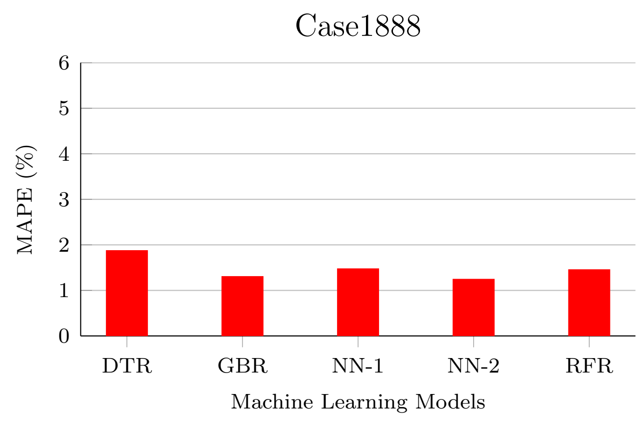


Numerical Results



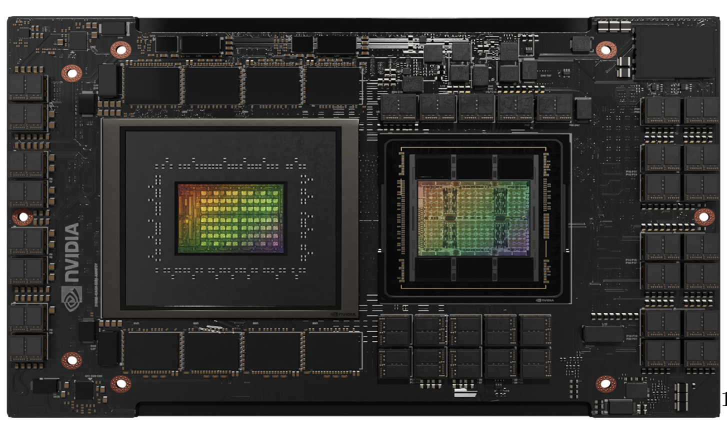






Comparing Processing and Training time

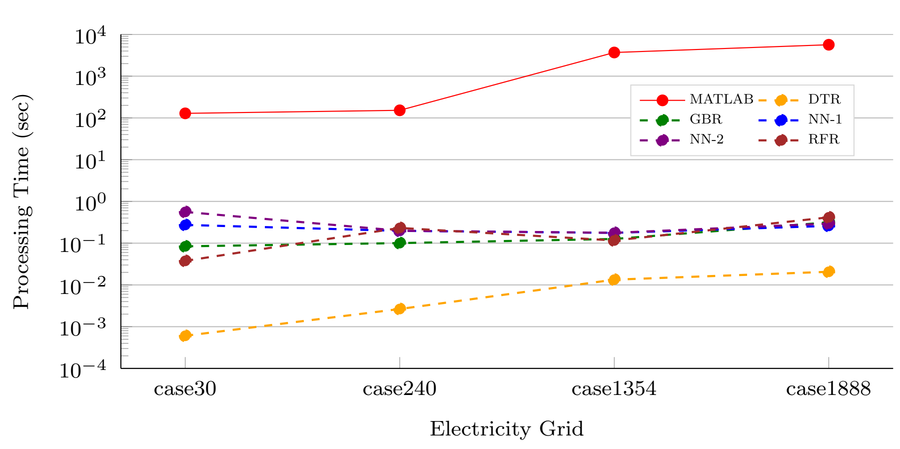
Benchmark Setup



1

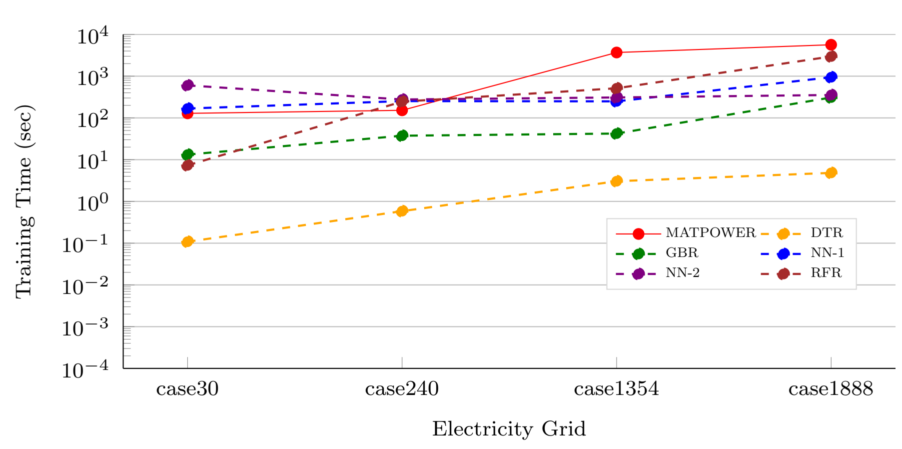
The simulations are performed on the ICS cluster at USI, Lugano, which consists of 41 nodes equipped with two 10-core Intel Xeon E5-2650 v3 with frequency 2.30GHz. The nodes have 128 GB RAM memory. The language and library stack used for this projects are: Python 3.7, MATLAB R2020a, MATPOWER 7.1, MOSEK 10, PyTorch 10.1, Scikit-Learn 1.0.2, Bayesian Optimization 1.4.2 and Scikit-Optimize 0.9.0.

Processing Time



Processing time represents the time required to generate the response by a pre-trained ML model for 5000 instances of electricity grids representing LMP prediction. This measure reveals how long the ML model will take to process the data once the training is complete. We compare it with the time taken to solve the optimization problems of the same instances using MATPOWER and MOSEK.

Training Time



Training time is defined as the average time it takes for a ML model to learn from 5000 electricity grid snapshots. Once trained, the model can process new data, which is usually significantly faster than the training process. The plot compares the training time of the ML models with the processing time of MATPOWER and MOSEK for 5000 instances of electricity grid data.

References

[1] N. V. S. J. Jami, "Feasibility and robustness of machine learning algorithms for calculating locational marginal pricing," Master's thesis, Università della Svizzera italiana, 2023.

[2] N. V. S. J. Jami, J. Kardoš, O. Schenk, and H. Köstler, "Ai driven near real-time locational marginal pricing method: A feasibility and robustness study," 2023.