# AI Driven Near Real-time Locational Marginal Pricing Method: A Feasibility and Robustness Study

Naga Venkata Sai Jitin Jami Institute of Computing (CI) Università della Svizzera italiana Lugano, Switzerland jitin.jami@usi.ch Juraj Kardoš Institute of Computing (CI) Università della Svizzera italiana Lugano, Switzerland juraj.kardos@usi.ch

Harald Köstler Department of Computer Science Friedrich-Alexander-Universität Erlangen, Germany harald.koestler@fau.de Olaf Schenk Institute of Computing (CI) Università della Svizzera italiana Lugano, Switzerland olaf.schenk@usi.ch

Abstract—Accurate price predictions are essential for market participants in order to optimize their operational schedules and bidding strategies, especially in the current context where electricity prices become more volatile and less predictable using classical approaches. The Locational Marginal Pricing (LMP) pricing mechanism is used in many modern power markets, where the traditional approach utilizes optimal power flow (OPF) solvers. However, for large electricity grids this process becomes prohibitively time-consuming and computationally intensive. Machine learning (ML) based predictions could provide an efficient tool for LMP prediction, especially in energy markets with intermittent sources like renewable energy. This study evaluates the performance of popular machine learning and deep learning models in predicting LMP on multiple electricity grids. The accuracy and robustness of these models in predicting LMP is assessed considering multiple scenarios. The results show that ML models can predict LMP 4-5 orders of magnitude faster than traditional OPF solvers with 5-6% error rate, highlighting the potential of ML models in LMP prediction for large-scale power models with the assistance of hardware infrastructure like multi-core CPUs and GPUs in modern HPC clusters.

Index Terms—Future Energy Markets, Locational Margin Pricing, Machine Learning, Price Prediction, Uncertainty Management, High Performance Computing

## I. INTRODUCTION

Nodal Pricing, also known as Locational Marginal Pricing (LMP), is a widely-used pricing mechanism in energy markets around the world, including the US, Ireland, New Zealand, and Singapore [1]. It calculates nodal level remuneration based on location, acknowledging the importance of location when it comes to electricity price [2]. 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 [3]. A regular power system is generally very large and complex, thus computing the LMP of each node becomes prohibitively expensive. To address this challenge, machine learning tools can be leveraged not only

to predict prices, but also to guide their operational schedules based on changing market conditions.

The increasing penetration of renewable energy sources in the energy mix, such as solar and wind, increases the volatility and unpredictability of electricity prices. ML models can help to mitigate these uncertainties and enable stakeholders in the energy market to make more informed decisions. The LMP is usually determined by solving the direct current optimal power flow (DC-OPF) problem, a simplified version of optimal power flow (OPF) which is a constrained nonlinear programming problem [4]. The large-scale nature of OPF problems exhibits excessive hardware and computational-time requirements, especially when dealing with real-time predictions of realistic electricity grids [5]. In order to retain the operational standards of the power grid operators or the market participants, they need to re-run the OPF very often, following the forecast updates of the renewable energy sources.

Price prediction and price forecasting tasks are increasingly relying on ML methodologies to reduce processing time [2]. Solution of ML models can also be used as warm-start points for solution using iterative solvers. Models can be tailored towards solving for a subset of the OPF solution like system voltages, generator scheduling or LMP prediction. With our proposed ML model solution, the ML model can be pre-trained on the data samples around the most-likely scenarios based on grid-specific data and the on-line prediction for any updated scenario can be obtained from the pre-trained ML model.

In previous studies, Graph Neural Networks were used in [6] to create a topology-aware Neural Network (NN) so that any changes to the grid do not trigger retraining of the model or hamper prediction capabilities due to over-dependence on historical data. In [7], the author tries to solve the AC-OPF problem with random forests to predict voltage and generation solutions that would serve as an intelligent, warm start for further solving the AC-OPF. In [8], the authors use a NN to predict optimal AC-OPF solution for 3 IEEE standard electricity grids. They also comment on the feasibility of the solution and the amount of time saved. The authors of [9] focuses further on implementing hard feasibility strategies to ensure the predicted solutions are within the feasible space. Their subsequent paper [10] uses similar strategies on the DC-OPF formulation. DC-OPF formulation was directly solved using a convex NN in [11] and used the KKT conditions to construct a loss metric for training. In [12], the authors use a blend of seasonal, weather, economic and operational data to predict LMP value in Mexico using ANNs and global senstivity analysis (GSA) to determine the importance of external factors. Transformers, a well-known sequence to sequence(seq2seq) architecture for deep learning, were used to predict LMP in the PJM power market using historical data.

Accuracy of several popular ML and deep learning models proposed in the literature is studied in context of LMP price prediction based on data involving demand, supply and generator cost curves. Various edge-case scenarios and the grid topology adjustments are considered to test model robustness and assess the performance scalability to larger electricity grids. The ML models considered in this study are Decision Tree Regression (DTR), Gradient Boosting Regression (GBR), Random Forest Regression (RFR), and Deep Neural Network with multiple hidden layers (NN - 1 & 2). When assessing the accuracy and robustness of different ML models, there are a few key factors to consider. One important factor is the model's predictive performance, which can be measured using metrics like Mean Absolute Percentage Error (MAPE). It is also essential to consider the robustness of the model, which refers to its ability to perform well on a wide range of input data. This can be evaluated by testing the model on various data sets that simulate changes in market conditions and other external factors. The scalability of the accuracy and robustness can be tested by running experiments on electricity grids of various sizes.

The remaining part of this paper is structured as follows. Sec. II explains the data generation methodology used to generate training and testing data. Sec. III gives an overview on the numerical experiments conducted and the results of the experiments. Finally, Sec. IV provides concluding remarks and directions for future work.

# II. DATA GENERATION

All the training and testing ground truth data is generated with the help of MATLAB. The electricity grids of choice are chosen from PGLib-OPF [13]. The PGLib-OPF models contain static data, i.e., a snapshot of the grid state in a single time instance, consisting of individual voltage levels across nodes, grid topology, power injection and power withdrawal at nodes. This single snapshot translates into a single DC-OPF optimization problem imported into MATPOWER [5] that is solved using an optimization solver such as MOSEK [14]. The choice of the electricity grid and their properties are detailed in Tab. I.

The PGLib-OPF provides a single feasible snapshot for each electricity grid. However, training a ML model will require more than a single snapshot (or single instance) that have

TABLE I: Description of electricity grids under consideration.

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

to be dissimilar from each other. In order to generate these heterogeneous instances of data, perturbations at the nodal level are introduced. When the active loads are perturbed at all nodes, the resulting DC-OPF problem differs from the base case also in terms of the nodal prices. The nodal level perturbations  $s_{Pd}$  are calculated using the following equation:

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

where the scalar  $s_{grid}$  is the nodal perturbation expressed in percent relative to the base case and is constant for all the grid nodes in a given problem instance. The scaling factor  $s_{nodal}$  is a vector of uniformly distributed random numbers in the range (0.9, 1.1) with the length equal to the number of the grid nodes. The motivation behind the nodal perturbation Eq. (1) is to simulate a given global trend ( $s_{grid}$ ) in the grid, such as an increase or decrease of the active load, but this is not equal across all the nodes in the grid. Thus, we apply random noise at the nodal level ( $s_{nodal}$ ) to these global trends to achieve the final scaling factor.

TABLE II: Range of the global perturbations  $s_{grid}$ .

Case	Perturbation Range (%)
case30	{-30, 30}
case240	{-70, -10}
case1354	$\{-50, 0\}$
case1888	$\{-40, 10\}$

# A. Training Data

The base snapshot of each grid is first manipulated with the help of Eq. (1) to generate up to 5000 heterogeneous instances. The  $s_{grid}$  parameter was randomly sampled from the given range for each problem instance. Each instance corresponds to a separate optimization problem solved with MATPOWER and MOSEK. We extract the information to use as input features  $x_i$  for training the ML model. The input features of each node (demand and generator nodes) on the electricity grid are active power demands  $P_d$  at each load bus and transmission capacity factor  $P_l$  is defined as

$$P_l = \frac{P_d}{P_l^{max}},\tag{2}$$

where  $P_d$  represents nodal active power demand and  $P_l^{max}$  represents the maximum capacity of the connected transmission lines of a given node. The input features are not part of the solution of MATPOWER and MOSEK, they are part of the problem definition.

TABLE III: Test cases.

Test Case	Description
1	Base case
2	10% reduction on transmission capacity
3	Missing 1 transmission line at random
4	Missing 1 generator at random

# B. Testing Data

The generation of test data is crucial to our objectives of measuring accuracy and testing robustness. We achieve this by generating 4 testing datasets described in Tab III. Data for each case was generated the same way our training data was but for only 100 heterogeneous instances. We also choose the same input features. However, the data for each instance in this dataset has been generated by further manipulating the electricity grid apart from introducing nodal perturbations.

The test cases aim to test the robustness of trained ML models under various edge-case scenarios. All the data has been generated by first introducing the same grid level perturbations shown in Tab. II. The test case 1 has no further manipulation. It has the same characteristics as training data but the specific instances have not been seen by the models during the training case. The remaining cases have been chosen to mimic real life scenarios like maintenance cycles or outages of the power grid components and simulate additional edge-case scenarios commonly occurring in security analysis of the power grid in the literature.

#### **III. NUMERICAL EXPERIMENTS**

The simulations are performed on the ICS cluster at USI, Lugano, which consists of 41 nodes equipped with two 10core 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. Training and testing data was generated using MATPOWER and MOSEK. 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 hyper-parameters of all the models are shown in Tab. IV.

The generated data is first pre-processed such that all features have zero mean and unit variance across all the instances of that experiment. In each experiment, 100 instances of the ML models were tested on the same testing dataset. Each model is first trained. Next, the LMP value of the test cases is predicted and the MAPE shown in Eq. (3) is calculated. The error of the prediction is calculated against the ground truth, which is the LMP value calculated by solving the DC-OPF with MATPOWER and MOSEK. This metric expresses the average error of the predictions in the 100 trained model instances.

DTR			odes	min_samples_le		leaf	min_samples_split		
case30		110			130	120			
case240		60			190			170	
case1354		60			190		170		
case1888		110			130		120		
RFR	n	nax_leaf_nodes	min	samples_leaf	amples_leafmin_sam		oles_split	n_estimators	
case30		100		100			140	100	
case240		170		180			180	700	
case1354		30		190		70		300	
case1888		60		190			170	1200	
GBR		learning rate max depth		n	n estimators		subsample		
0220		0.00		- 1	1	1500		0.2	
case30		0.09		2	-	1500		0.2	
Casez40		0.	09	<i>y</i> 2		1000		0.1	
case1354		0.01		4			200	0.1	
case188	8	0.	08	2		1800		0.1	
DNN-1	1	# hidden lave	rs	Topology	I	earnii	1g Rate	Batch Size	
case30			1	[128]	1		0.009	128	
case240	+		2	[128.64]			0.006	128	
case1354	+		2	[128.64]			0.004	128	
case1888			3	[4096,512,32]			0.001	128	
				. , , , ,					
DNN-2	#	# hidden layers Tope		oology	ology		ning Rate	Batch Size	
case30		2		[512,3	32]		0.008	32	
case240		3		[128,32,3	32]		0.005	128	
case1354		3		[128,32,1	[128,32,16]		0.005	128	
case1888		4 [409		96, 2048, 512, 3	5, 2048, 512, 32]		0.001	128	

The MAPE is defined as

$$MAPE(\bar{y}, \hat{y}) = \frac{100}{NS} \sum_{i=0}^{NS-1} \frac{|\bar{y}_i - \hat{y}_i|}{|\bar{y}_i|},$$
(3)

where NS is the number of samples used in the experiments,  $\bar{y}_i$  is the ground truth and  $\hat{y}_i$  is the model prediction.

We focus on 3 different aspects of our study, namely processing time, training time and accuracy. Additionally, we aim to investigate the feasibility, robustness and scalability of using ML algorithms for the LMP predictions.

### A. 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. In case of the ground truth, the processing time represents the elapsed time to solve the optimization problems of the same instances using MATPOWER and MOSEK. Figure 1a presents a comparison of the processing time for 5000 grid snapshots using the classical optimization tools and the ML models considered in this study.

As demonstrated in Fig. 1a, MATPOWER requires up to five orders of magnitude more time than ML models to process the data. Among the ML models we assessed, DTR exhibits the lowest processing time. It is noteworthy that the processing time does not increase significantly for ML models as the grid size increases, which is not the case for MATPOWER and MOSEK.

# B. Training Time

Training time is defined as the average time it takes for a ML model to learn from 5000 electricity grid snapshots.



Fig. 1: Timing statistics for MATPOWER and various ML models.

The time required for training is dependent on several factors, such as the complexity of the model, chosen hyperparameters, and the hardware used. Once trained, the model can process new data, which is usually significantly faster than the training process. Fig. 1b compares the training time of the ML models with the processing time of MATPOWER and MOSEK for 5000 instances of electricity grid data.

The comparison in Fig. 1b shows that ML models take a comparable amount of time to train as MATPOWER takes to solve 5000 instances of electricity grid. Furthermore, for larger grids, the ML models can be trained faster compared to the time it takes to solve the instances using MATPOWER. This observation is significant because as the size and complexity of the grid increases, this difference is expected to become more prominent. However, note that the training time excludes the time necessary to generate the training data. The generation of the training data is equivalent to the MATPOWER time shown in the figure.

#### C. Performance

The performance of the models relies on the size of the training dataset. The DTR model was trained on varying dataset sizes ranging from 1000 to 50000 problem instances to see the how the MAPE changes.



Fig. 2: Accuracy of all models on test case 1 for case240.

As can be seen in Fig. 2, the model's MAPE error decreases as the number of training samples increases for test case 1. The difference in performance between 5000 and 50000 points amounts to only a minor improvement of 0.02% (GBR) to 0.6% (DTR) based on the model. The idea of increasing training data points follows the law of diminishing returns. The performance gains may not justify the increased computational effort. Thus, the data set size of 5000 instances is considered in the following experiments to balance computational costs with accuracy.

Next, the performance of ML models are compared in terms of the accuracy of their predictions on the various testing cases. The error is evaluated using the MAPE % for all the grids for different cases from tab III.

**Test Case 1:** Fig. 3a presents the performance of models on data from test case 1. For the case30 grid, all ML models have achieved the MAPE under 1%. The ML models show a slighter lower accuracy on larger grid with MAPE under 2%, which is a trend that is consistent across all grids an exception of case1354. The MAPE of the case1354 grid ranges between 3.5% for DNN and RFR, while DTR and GBR reached MAPE above 5%. However, this might be still considered as acceptable error considering the trade-off with processing time.

**Test Case 2:** The results shown in Fig. 3b indicate that reducing the maximum transmission capacity by approximately 10% did not affect the performance of the ML models compared to their accuracy in test case 1. The base case electricity grid configurations in the PGLib-OPF definitions do not typically operate at maximum capacity. As a result, our data set includes very few instances where resources are pushed to the maximum when introducing perturbations. This could explain why similar levels of MAPE errors are observed in the two test cases.

**Test Case 3**: Fig. 3c shows the results for test case 3, where one transmission line in each electricity grid was removed. The number of transmission lines varies across the different grids, and we observed the highest performance drop in case30, which has only 41 transmission lines, compared to case1888 with 2531 transmission lines. The drop in performance is more likely to occur in smaller grids because the difference between the base case and modified definitions may significantly alter the topology of the grid and the feasible power flows in the grid. Overall, the results suggest that the impact of shutting off one transmission line on the performance of ML models is grid-dependent.

**Test Case 4:** The results for test case 4 are shown in Fig. 3d. This test case further amplifies the impact observed in case 3 by disconnecting a generator from the grid. Since each grid has a limited number of generators, turning off a crucial one can significantly affect the feasible region of the power flows. As a result, the most significant performance drop is observed



Fig. 3: Accuracy of the ML models in terms of MAPE % for all the grids for different test cases.

in the smallest grid, case30. However, the impact on model accuracy is negligible for other grids compared to the first test case.

## **IV. CONCLUSIONS**

This study extends the previous literature by investigating the feasibility and robustness of ML models in predicting LMP. Comparing the processing and training time of ML models with MATPOWER's processing time revealed that ML models are indeed feasible and offer a significantly faster alternative once the ML is trained. ML models provide acceptable accuracy, with MAPE within 6% for all test cases, indicating the robustness of ML models even on data that simulates edge-case scenarios. We found that the DTR tends to have the highest MAPE error among popular ML models, while NN-1 and NN-2 outperformed their peers. However, NNs have a major disadvantage of considerably longer training time. Overall, by significantly reducing the processing and training time compared to traditional methods, ML models can help to reduce the overall cost of LMP predictions and forecasting. This is particularly important in future energy markets that will have high penetration from renewable energy sources, which can make prices more uncertain and volatile. These findings suggest that the adoption of ML models for LMP prediction has significant potential to improve the efficiency of energy markets in the future.

#### References

- J. Lin and F. H. Magnago, Pricing, Modeling, and Simulation of an Electricity Market, pp. 211–238. John Wiley & Sons, 2017.
- [2] S. Cantillo-Luna, R. Moreno-Chuquen, H. R. Chamorro, J. M. Riquelme-Dominguez, and F. Gonzalez-Longatt, "Locational marginal price forecasting using svr-based multi-output regression in electricity markets," *Energies*, vol. 15, no. 1, p. 293, 2022.
- [3] K. Zheng, Y. Wang, K. Liu, and Q. Chen, "Locational marginal price forecasting: A componential and ensemble approach," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4555–4564, 2020.
- [4] H. Bai, M. Yang, D. Song, Y. He, T. Su, and X. Jiang, "A review of the application of locational marginal price theory in the new situation," in 2019 IEEE 3rd Conference on Energy Internet and Energy System Integration (EI2), pp. 2126–2131, IEEE, 2019.

- [5] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on power systems*, vol. 26, no. 1, pp. 12–19, 2010.
- [6] S. Liu, C. Wu, and H. Zhu, "Topology-aware graph neural networks for learning feasible and adaptive ac-opf solutions," *arXiv preprint* arXiv:2205.10129, 2022.
- [7] K. Baker, "Learning warm-start points for ac optimal power flow," in 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6, IEEE, 2019.
- [8] A. S. Zamzam and K. Baker, "Learning optimal solutions for extremely fast ac optimal power flow," in 2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), pp. 1–6, IEEE, 2020.
- [9] T. Zhao, X. Pan, M. Chen, and S. H. Low, "Ensuring dnn solution feasibility for optimization problems with convex constraints and its application to dc optimal power flow problems," *arXiv preprint* arXiv:2112.08091, 2021.
- [10] X. Pan, "Deepopf: deep neural networks for optimal power flow," in Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, pp. 250–251, 2021.
- [11] L. Zhang, Y. Chen, and B. Zhang, "A convex neural network solver for dcopf with generalization guarantees," *IEEE Transactions on Control of Network Systems*, 2021.
- [12] A. Livas-García, O. M. Tzuc, E. C. May, R. Tariq, M. J. Torres, and A. Bassam, "Forecasting of locational marginal price components with artificial intelligence and sensitivity analysis: A study under tropical weather and renewable power for the mexican southeast," *Electric Power Systems Research*, vol. 206, p. 107793, 2022.
- [13] S. Babaeinejadsarookolaee, A. Birchfield, R. D. Christie, C. Coffrin, C. DeMarco, R. Diao, M. Ferris, S. Fliscounakis, S. Greene, R. Huang, *et al.*, "The power grid library for benchmarking ac optimal power flow algorithms," *arXiv preprint arXiv:1908.02788*, 2019.
- [14] M. ApS, The MOSEK optimization toolbox for MATLAB manual. Version 10.0.25, 2022.

This figure "fig1.png" is available in "png" format from:

http://arxiv.org/ps/2306.10080v2