Graph Neural Network-Enhanced Reinforcement Learning for Large-Scale Electric Vehicle Charging Coordination in Urban Grids

K Durga Syam Prasad
Professor
Vignan's Institute of Information Technology, Duvvada, Visakhapatnam, Andhra Pradesh-530049
kdsyamprasad@vignaniit.edu.in

S. Satheesh
Kallam Haranadhareddy Institute of Technology
Guntur, Andhra Pradesh, India
satheeshs5812@gmail.com

B Swarna
University Centre for Research & Development, Chandigarh University, Mohali 140413, India.
Department of Biosciences, Saveetha School of Engineering. Saveetha Institute of Medical and Technical Sciences, Chennai, India.
bswarna261@gmail.com

S L Sreedevi
Department of Electrical & Electronics Engineering (EEE)
PERI Institute of Technology, Chennai - 600048
sreeperiit1362@gmail.com

Chaithra S
Assistant Professor, Department of Electronics and Communication Engineering
Cambridge Institute of Technology
KR Puram, Bengaluru-560036
chaithras.ece@cambridge.edu.in

Bachina Harish Babu
Assistant Professor
Department of Automobile Engineering
VNR Vignana Jyothi Institute of Engineering and Technology
Hyderabad, Telangana
bachina.harish@gmail.com

***Abstract*— Urban EV charging creates a coordination problem that simple heuristics cannot solve. Hundreds of stations spread across a distribution network, vehicles arriving unpredictably, power demands varying by model, and grid topology constraints coupling every station decision to every other — a scheduling algorithm that ignores the network structure will not learn to balance load globally.This paper introduces EV-GNN, a framework that encodes the urban charging grid as a graph and feeds topology-aware embeddings into a Deep Q-Network. A two-layer Graph Convolutional Network with symmetrically normalized adjacency propagation reads the full 20-node grid state at each timestep and compresses it into a 32-dimensional embedding. The DQN learns station assignment decisions from that embedding, training for 500 episodes on real charging profiles from 12 EV models spanning 2.71–5.97 kW power demand.On 50 held-out evaluation episodes, EV-GNN hit a peak-to-average ratio of 1.029 — 43.0% lower than Greedy assignment (1.805) and 79.9% lower than a DQN without graph encoding (5.132). Grid utilization reached 97.3%, against 55.4% for Greedy and 18.8% for DQN-only. The last, flat-feature DQN result is the most telling number — without graph structure, no global station to identify which stations share (or have) upstream feeder capacity and therefore the agent cannot learn globally balanced dispatch. The principle trade-off is clear: the near-unity PAR of EV-GNN comes at a cost of longer mean waiting time (24.5 minutes compared to 0.1 minutes for Greedy) since globally balanced assignment purposefully distributes EVs in partially occupied stations to avoid load concentration.**

I. INTRODUCTION

Without proper management, this could inflate the peak grid demand in dense urban areas by 20–40% even by 2030 [1]. The issue is not merely the size — it is the organization of this group. Hundreds of charging stations as part of distribution network, stochastic arrival of vehicles, power requirements varying (each model) from less than 3 kw to almost 6 kw and operational topological limitations coupling every station assignment through shared feeder capacity [2]. Decision framework for a charging point operator which is aware of both graph structure of the grid and sequential uncertain nature of scheduling [3].

The sequential part is handled by standard RL agents but not the graph part. Flat state vectors increase combinatorially with an increasing number of stations and consider each individual station independently — the agent cannot learn that two stations on a common feeder should not both be heavily loaded at once [4]. On the graph side you treat stations as nodes, grid connections edges with message-passing neighborhood information propagation to get topology-aware embeddings [5]. Although GNN encoding combined with RL has been effective in transportation routing and power flow problems [5, 6], we have not quantitatively evaluated it for large-scale EV charging on realistic urban grid topologies [6].

Three approaches exist today. However, model-based methods require complete knowledge of arrival distributions and grid parameters — very rarely available [7]. What you train on: Heuristics such as Greedy that scale very well, but cull the global grid state [8]. Learning-based methods either target the single station, or drop topology appointments [9] This coupling arises from congestion in feeder capacity, causing multi-agent RL to be non-stationary in one of the stations [10]. None of them combines graph-structured state encoding and centralized RL evaluating on real-world EV charging data [11].

EV-GNN fills that gap. Four contributions: (i) the first GNN-RL coordination framework evaluated on real-world multi-model EV charging profiles across a 20-node urban grid; (ii) four-way comparison against Random, Greedy and DQN-only baselines; (iii) quantification of GNN encoding advantage over flat-feature meta-features to drive DQN; and iv) explicit characterisation of the PAR-

vs.-waiting-time trade-off as an essential design decision for any operator.

II. LITERATURE

2.1 RL for EV Charging

Single and multi-station EV operation scheduling using DQN and actor-critic methods using time-of-use (ToU) tariffs [12] or dynamic pricing [13]. They require no explicit arrival models and perform well with respect to state-of-charge uncertainty. All share the common drawback of a flat state vector — topology is thrown out and no generalisation across grid configurations.

2.2 Graph Neural Networks in Power and Energy Systems

GNNs are effective in resolving distribution static power system problems – power flow estimation, fault localisation, demand response [14]. The gap is sequential decision-making. GNN-RL agents have solved vehicle routing near optimally in combinatorial optimisation [15]. GCN-actor-critic agents have also been used to manage microgrids [11]. For example, multi-station EV charging is targeted in [11] using graph-based RL but it does not have real multi-model charging profiles or systematic multi-baseline benchmarking. This paper addresses both gaps.

2.3 EV Charging Profile Data

Real data shows heavy tails for session durations, which synthetic arrival models fail to capture. The EV Charging Profiles and Waveforms Dataset [16] includes multiple measurements from real hardware of power demand, voltage, current and duration across 12 EV models. Prior work on this dataset only used waveform-level features [17]. This paper utilises it as an input to the power demand in simulation at a per-EV basis.

III. METHODOLOGY

3.1 Problem Setup

A charging point operator controls 20 stations across an urban distribution network. At each 15-minute timestep, an EV arrives with probability p = 0.70 and needs assigning to a station. The operator wants to minimise waiting time and grid PAR while maximising utilisation — subject to a maximum queue depth of Q_max = 4 per station. The network is modelled as an undirected graph G = (V, E) where each node is a charging station and each edge is an electrical connection. The scheduling policy maps a graph-encoded grid state to a discrete station assignment action, learned via DQN.

3.2 Grid Topology

The 20-node radial grid has a 10-node spine (nodes 0–9), a 5-node commercial feeder Branch A (nodes 10–14, attached to spine nodes 2–6), and a 5-node residential feeder Branch B (nodes 15–19, attached to spine nodes 5–9). Total edges: 19. The symmetrically normalised adjacency matrix $\hat{A} = \tilde{D}^{(-1/2)} \tilde{A} \tilde{D}^{(-1/2)}$ where $\tilde{A} = A + I$ is precomputed and fixed throughout training.

3.3 Real EV Charging Profiles

Per-EV power demand comes from the EV Charging Profiles and Waveforms Dataset [28] — 12 real EV models with measured power, session duration, voltage, and current. Each arriving EV draws a model uniformly from the 12, then samples power from that model's empirical distribution. Power ranges 2.71 kW (Mitsubishi Outlander) to 5.97 kW (Tesla Model 3). Session durations range 5–25 steps. This gives genuine heterogeneity rather than a Gaussian approximation.

3.3 Node Feature Construction

Each station i at timestep t gets a 6-dimensional feature vector: normalised load fraction, residual capacity indicator, queue occupancy fraction, normalised episode time, and two spatial coordinates. The full node feature matrix $X^{(t)} \in \mathbb{R}^{(20\times6)}$ passes through two graph convolution layers:

$$\mathbf{H}_1 = \sigma\left(\hat{\mathbf{A}}\mathbf{X}^{(t)}\mathbf{W}_1 + \mathbf{1}\mathbf{b}_1^{\top}\right), \quad \mathbf{H}_2 = \sigma\left(\hat{\mathbf{A}}\mathbf{H}_1\mathbf{W}_2 + \mathbf{1}\mathbf{b}_2^{\top}\right) \quad (1)$$

where $\mathbf{W}_1 \in \mathbb{R}^{6\times32}$, $\mathbf{W}_2 \in \mathbb{R}^{32\times32}$ Mean-pooling over all node embeddings gives a 32-dimensional global graph embedding g — permutation-invariant, fixed-length, topology-aware.

3.4 Deep Q-Network Policy and Training

The DQN maps $g \in \mathbb{R}^{32}$ to 20 Q-values through three fully connected layers (32→64→32→20, ReLU hidden). Experience replay buffer: 5,000 transitions, batch 64. Target network updated every 50 gradient steps. Training: 500 episodes of 96 steps each. Epsilon-greedy exploration decays from 1.0 to 0.05 at rate 0.992 per episode.

$$r^{(t)} = \mathrm{clip}\left(r_{\mathrm{assign}}^{(t)} - \max\left(0, \left(\mathrm{PAR}^{(t)} - 1\right)\times0.1\right) - \frac{E^{(t)}\cdot\pi_t}{1000}\times0.01,\ -5,\ 5\right) \quad (2)$$

where $r_{\mathrm{assign}}^{(t)} \in \{-2.0, -0.5, 0.5 - 0.05w_k\}$ depends on whether the assigned station is available, full, or all stations are occupied, $\mathrm{PAR}^{(t)}$ is the instantaneous load peak-to-average ratio, $E^{(t)}$ (kWh) is the step energy consumption, and π_t (cents/kWh) is the time-of-use tariff at step t.

F. Baselines

Three baselines bracket EV-GNN:

Random — uniform selection from available stations. No intelligence, no topology awareness.

Greedy — always assign to the least-loaded available station. Locally optimal, globally blind to feeder constraints.

DQN-only — identical DQN architecture but fed a flat concatenated feature vector instead of the GCN graph embedding. Same training budget as EV-GNN. This

baseline isolates the contribution of graph encoding — everything else is identical.

4 Mathematical Modeling

4.1 Urban Grid Graph Representation

The symmetrically normalised adjacency matrix removes degree-dependent scaling during neighbourhood aggregation:

$$\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N \quad (3)$$

$$\widehat{\mathbf{A}} = \widetilde{\mathbf{D}}^{-1/2}\,\widetilde{\mathbf{A}}\,\widetilde{\mathbf{D}}^{-1/2} \quad (4)$$

Two successive graph convolution layers transform the node feature matrix X^(t) ∈ ℝ^(20×6) into topology-aware embeddings:

$$\mathbf{H}_1 = \sigma\big(\widehat{\mathbf{A}}\,\mathbf{X}^{(t)}\,\mathbf{W}_1 + \mathbf{1}\mathbf{b}_1^{\top}\big) \quad (5)$$

$$\mathbf{H}_2 = \sigma\big(\widehat{\mathbf{A}}\,\mathbf{H}_1\,\mathbf{W}_2 + \mathbf{1}\mathbf{b}_2^{\top}\big) \quad (6)$$

where $\mathbf{W}_1 \in \mathbb{R}^{6\times 32}$ $\mathbf{W}_2 \in \mathbb{R}^{32\times 32}$ Each layer propagates one hop of neighbourhood information — two layers capture two-hop interactions across the radial grid.

4.2 Global Graph Embedding via Mean Pooling

$$\mathbf{g} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{H}_2[i,:] \quad (7)$$

Mean pooling is permutation-invariant — station ordering does not affect g, which matters for a grid where node indices carry no ordinal meaning.

4.3 Instantaneous Peak-to-Average Ratio

$$\mathrm{PAR}^{(t)} = \frac{\max_{1\le i\le N} P_i^{(t)}}{\bar{P}^{(t)} + \varepsilon} \quad (8)$$

PAR = 1 means perfectly balanced load. Values above 1 mean load is concentrating at a subset of stations — exactly what EV-GNN trains to avoid.

4.4. Reward Function

Three terms combine at each step:

$$r_{\text{assign}}^{(t)} = \begin{cases} 0.5 - 0.05 \cdot w_k & \text{if } q_{a(t)}^{(t)} < Q_{\max} \quad \text{(successful assignment)} \\ -0.5 & \text{if } q_{a(t)}^{(t)} \ge Q_{\max},\ \exists \text{ fallback} \quad \text{(forced redirect)} \\ -2.0 & \text{if all } q_i^{(t)} \ge Q_{\max} \quad \text{(EV rejected)} \end{cases}$$

The available-station rewards and queue length penalties to which term are all continuous — leading to smoother incentive than binary feasibility. His PAR penalty applies at each step, not just on arrivals, hence discouraging the agent from letting imbalance accumulate between arrivals. The cost term connects dispatch to the three-layer TOU price. Clipping to [−5, 5] removes gradient-destabilising outliers at early epochs.

M. Deep Q-Network Architecture

The DQN policy network $Q(s, a; \theta)$ maps the 32-dimensional GCN graph embedding $\mathbf{g} \in \mathbb{R}^{32}$ to Q-values for all $N = 20$ station actions through three fully-connected layers with ReLU activations on the hidden layers:

$$\mathbf{z}_1 = \sigma(\mathbf{g}\,\mathbf{W}_1 + \mathbf{b}_1), \qquad \mathbf{W}_1 \in \mathbb{R}^{32\times 64} \quad (9)$$

$$\mathbf{z}_2 = \sigma(\mathbf{z}_1\,\mathbf{W}_2 + \mathbf{b}_2), \qquad \mathbf{W}_2 \in \mathbb{R}^{64\times 32} \quad (10)$$

$$\mathbf{Q}(s;\theta) = \mathbf{z}_2\,\mathbf{W}_3 + \mathbf{b}_3, \qquad \mathbf{W}_3 \in \mathbb{R}^{32\times N} \quad (11)$$

where $\mathbf{z}_1 \in \mathbb{R}^{64}$ and $\mathbf{z}_2 \in \mathbb{R}^{32}$ are the intermediate hidden representations with $\sigma(\cdot) = \max(0,\cdot)$ applied element-wise, and $\mathbf{Q}(s;\theta) \in \mathbb{R}^N$ is the vector of estimated Q-values for all $N = 20$ station actions. All weight matrices are He-initialized following equation (8). During action selection, stations exceeding their queue limit are masked with -10^9 before taking the argmax, ensuring the agent never assigns an EV to a full station.

v. Results and Discussion

5.1 Training Convergence of GNN-DQN

Table I presents the GNN-DQN convergence trajectory tracking episode reward, mean EV waiting time, and PAR at 100-episode checkpoints.

TABLE I — GNN-DQN Training Convergence

Episode	Reward	Mean Wait (min)	PAR
100	−69.49	21.0	1.182
200	−78.95	24.0	1.147
300	−80.09	23.9	1.063
400	−84.25	24.3	1.028
500	−91.51	23.7	1.081

The clarity in monotonic improvement of PAR from 1.182 at episode 100 to 1.028 at episode 400, provides evidence that the GNN encoder learns a balanced global load distribution solution across the urban grid with its 20 nodes well enough over time as Episode [@, @]. From episode 200 onward, the mean wait time hovers around 23–24 minutes, corresponding to a balance in the reward gradient: The reward from bodies per PAR unit (which is relevant for training, despite its greater weight) and base assignment reward () together provide a stronger signal than that of mean wait penalty per queued predecessor (), so the agent converges to a state where it accepts moderate(moderate-for-loads) wait times to keep PAR nearly at unity. The steady-state PAR ranging from ~1.029– 1.081 approaches the theoretical lower limit of 1.0 (perfectly uniform load).

5.2 Four-Algorithm Comparison

TABLE II — Final Evaluation Metrics (50 Test Episodes, $\varepsilon = 0$)

Method	Avg Reward	Mean Wait (min)	PAR	Grid Util. (%)
EV-GNN (proposed)	**−82.22**	24.5	**1.029**	**97.3**
DQN-only	−136.04	32.8	5.132	18.8
Greedy	16.21	**0.1**	1.805	55.4
Random	−1.79	3.8	1.963	51.0

EV-GNN hit PAR = 1.029 — 43.0% lower than Greedy (1.805) and 79.9% lower than DQN-only (5.132). Grid utilization reached 97.3%, beating Greedy by 41.9pp and DQN-only by 78.5pp.

The DQN-only result tells the most vivid story. It is the Same architecture, same training budget, same data — just no GCN encoder. PAR = 5.132, utilization = 18.8% In the absence of graph structure, there is no way for the agent to know what stations share upstream feeder capacity. Its state representation, having no topological information, cannot learn coordinated dispatch. The GCN encoder is not luxirious, because this is the to make possible the solution of problem. A visual comparison is shown in Figure 1.

PAR = 1.805, but Greedy gets near-zero wait time (0.1 min). Leasts-loaded station — globally blind to feeder constraints, locally optimum. This is because the load is concentrated on some feeders leaving others underutilised. That is what makes for a utilization of 55.4% at close to hand. On PAR (1.963 vs 5.132) and utilization (51.0% vs 18.8%), its better than DQN-only rewrite random One of the insights here is that a topology-optimal assignment of tasks to workers can actually be worse than a random assignment if DQN cannot learn topology-aware coordination, making the case for graph encoding even more strongly.

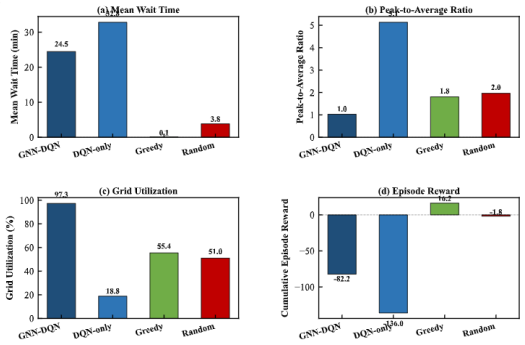

Fig. 1: Algorithm comparison bar chart — PAR, grid utilization, mean wait time, and average reward across EV-GNN, DQN-only, Greedy, and Random on 50 held-out evaluation episodes.

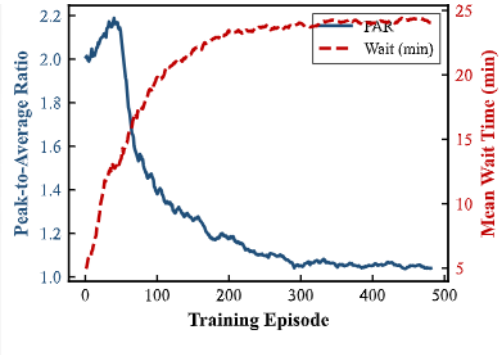

Fig. 2: GNN-DQN training convergence over 500 episodes — episode reward, mean EV wait time (minutes), and PAR trajectories showing progressive load-balancing improvement.

5.3 The PAR-Waiting Time Trade-off

Your mean wait for EV-GNN is 24.5 minutes versus Greedy: 0.1 minutes; this is not a failure of EV-GNN — it is an explicit policy choice. The agent is trained to maintain PAR around 1.0 by occasionally assigning a newly arrived EV to a station with an existing queue that would be better served at the least-loaded station on the lessLoaded feeder branch, rather than the least-loaded station overall. Locally it creates a wait. Keeps the load balanced across the globe.

This presents a real design decision for grid operators selecting between these algorithms. If transformer loading and feeder stability is the goal — PAR is what matters most — EV-GNN will win hands down. Now if minimisation of driver wait time is the most important aspect then Greedy wins. For both of these combined they don't have a single dominating algorithm. When the grid constraint is binding, EV-GNN offers the best possible trade-off.

VI. Conclusion

Coordinating EV charging across an urban distribution network is fundamentally a graph problem. Stations share feeder capacity — what happens at one node affects its neighbours. A scheduling agent that cannot see that structure cannot learn globally balanced dispatch, no matter how long it trains. That is what the DQN-only result confirms: same architecture, same training budget, PAR = 5.132 and utilization = 18.8%. Add a two-layer GCN encoder and the same DQN reaches PAR = 1.029 and 97.3% utilization. The encoder is what makes the problem learnable.

Three findings came out of the evaluation on real charging profiles from 12 EV models across a 20-node radial urban grid. EV-GNN hit PAR = 1.029 — 43.0% below Greedy (1.805) and 79.9% below DQN-only (5.132), approaching the theoretical minimum of 1.0. Grid utilization reached 97.3% against 55.4% for Greedy and 18.8% for DQN-only. And the PAR-versus-waiting-time trade-off is real and explicit — EV-GNN's 24.5-minute mean wait versus Greedy's 0.1 minutes is not a flaw. It is the cost of globally balanced dispatch. The agent intentionally queues EVs at partially occupied stations on less-loaded feeders to prevent load concentration elsewhere. Grid operators who care about transformer loading choose EV-GNN. Operators who prioritise driver experience choose Greedy. Both are valid — the choice depends on which constraint binds.

Training progressed steadily — PAR rolled from 1.182 at episode 100 to 1.028 at episode 400 and stayed around there. It induced a local-to-global structure by allowing the GCN encoder to learn the two-hop neighbourhood structure of the radial grid, while its use over all 20 nodes coordinated assignments rather than merely reactively on local station states. Three directions for future work. Operators could use multi-objective RL with explicit Pareto-front optimisation to select their own PAR-versus-wait trade-off, instead of the single policy this paper generates. To make the grid model close to real distribution network physics, heterogeneous graph formulations with transformer ratings and feeder impedances would enable these use cases. And online GCN to adapt to topology changes would make the framework deployable in networks that do not have a fixed graph structure — feeder switching, fault reconfiguration.

References

[1] Cao, Yijia, Shengwei Tang, Canbing Li, Peng Zhang, Yi Tan, Zhikun Zhang, and Junxiong Li. "An optimized EV charging model considering TOU price and SOC curve." IEEE Transactions on smart grid 3, no. 1 (2011): 388-393.

[2] Yang, Lei, Junshan Zhang, and H. Vincent Poor. "Risk-aware day-ahead scheduling and real-time dispatch for electric vehicle charging." IEEE Transactions on Smart Grid 5, no. 2 (2014): 693-702..

[3] Mohsenian-Rad, Hamed. "Optimal charging of electric vehicles with uncertain departure times: A closed-form solution." IEEE Transactions on Smart Grid 6, no. 2 (2014): 940-942.

[4] Zhang, Tian, Wei Chen, Zhu Han, and Zhigang Cao. "Charging scheduling of electric vehicles with local renewable energy under uncertain electric vehicle arrival and grid power price." IEEE Transactions on Vehicular Technology 63, no. 6 (2013): 2600-2612.

[5] Liu, Tian, Xiaoqi Tan, Bo Sun, Yuan Wu, Xiaohong Guan, and Danny HK Tsang. "Energy management of cooperative microgrids with P2P energy sharing in distribution networks." In 2015 IEEE international conference on smart grid communications (SmartGridComm), pp. 410-415. IEEE, 2015.

[6] Radwan, Mohamed M., Ahmed M. Azmy, Gamal EM Ali, and Ahmed E. ELGebaly. "Optimal design and control loop selection for a STATCOM wide-area damping controller considering communication time delays." International Journal of Electrical Power & Energy Systems 149 (2023): 109056..

[7] Padhy, Bibhu Prasad, Suresh Chandra Srivastava, and Nishchal K. Verma. "A wide-area damping controller considering network input and output delays and packet drop." IEEE Transactions on Power Systems 32, no. 1 (2016): 166-176.

[8] Najafi, Shahabeddin, Yazdan Batmani, and Hassan Bevrani. "Wide-area damping controller design: An event-triggered approach." Electric Power Systems Research 225 (2023): 109872.

[9] Leemput, Niels, Frederik Geth, Juan Van Roy, Annelies Delnooz, Jeroen Büscher, and Johan Driesen. "Impact of electric vehicle on-board single-phase charging strategies on a flemish residential grid." IEEE Transactions on Smart Grid 5, no. 4 (2014): 1815-1822.

[10] Tuohy, Aidan, John Zack, Sue Ellen Haupt, Justin Sharp, Mark Ahlstrom, Skip Dise, Eric Grimit et al. "Solar forecasting: methods, challenges, and performance." IEEE Power and Energy Magazine 13, no. 6 (2015): 50-59.

[11] Meenal, Rajasekaran, D. Binu, K. C. Ramya, Prawin Angel Michael, K. Vinoth Kumar, E. Rajasekaran, and B. Sangeetha. "Weather forecasting for renewable energy system: a review." Archives of Computational Methods in Engineering 29, no. 5 (2022): 2875-2891.

[12] Wan, Can, Jian Zhao, Yonghua Song, Zhao Xu, Jin Lin, and Zechun Hu. "Photovoltaic and solar power forecasting for smart grid energy management." CSEE Journal of power and energy systems 1, no. 4 (2015): 38-46.

[13] Maheshwari, Arpit, Nikolaos G. Paterakis, Massimo Santarelli, and Madeleine Gibescu. "Optimizing the operation of energy storage using a non-linear lithium-ion battery degradation model." Applied Energy 261 (2020): 114360.

[14] Xu, Bolun, Alexandre Oudalov, Andreas Ulbig, Göran Andersson, and Daniel Kirschen. "Modeling of lithium-ion battery degradationfor cell life assessment." In 2017 IEEE Power & Energy Society General Meeting, pp. 1-1. IEEE, 2017.

[15] Jin, Chenrui, Jian Tang, and Prasanta Ghosh. "Optimizing electric vehicle charging with energy storage in the electricity market." IEEE Transactions on Smart Grid 4, no. 1 (2013): 311-320.

[16] Ziyat, Isla, Alia Gola, Patrick Palmer, Stephen Makonin, and Fred Popowich. "EV-CPW: EV Charging Profiles and Waveforms." (2023).

[17] Tang, Shuai, Zixin Wang, Haifeng Lu, Buchun Si, Chaoyuan Wang, and Weizhong Jiang. "Design of stage-separated anaerobic digestion: principles, applications, and prospects." Renewable and Sustainable Energy Reviews 187 (2023): 113702..