# DCLEVERNET: Deep Combinatorial Learning for Efficient EV Charging Scheduling in Large-scale Networked Facilities

Bushra Alshehhi\* 100050085@ku.ac.ae Khalifa University Abu Dhabi, P.O. Box 127788, United Arab Emirates Areg Karapetyan\* areg.karapetyan@nyu.edu Khalifa University Abu Dhabi, P.O. Box 127788, United Arab Emirates New York University Abu Dhabi Abu Dhabi, P.O. Box 129188, United Arab Emirates

Sid Chi-Kin Chau sid.chau@acm.org Australian National University Canberra ACT 2601, Australia Khaled Elbassioni khaled.elbassioni@ku.ac.ae Khalifa University Abu Dhabi, P.O. Box 127788, United Arab Emirates

Majid Khonji majid.khonji@ku.ac.ae Khalifa University Abu Dhabi, P.O. Box 127788, United Arab Emirates

# ABSTRACT

With the electrification of transportation, the rising uptake of electric vehicles (EVs) might stress distribution networks significantly, leaving their performance degraded and stability jeopardized. To accommodate these new loads cost-effectively, modern power grids require coordinated or "smart" charging strategies capable of optimizing EV charging scheduling in a scalable and efficient fashion. With this in view, the present work focuses on reservation management programs for large-scale, networked EV charging stations. We formulate a time-coupled binary optimization problem that maximizes EV users' total welfare gain while accounting for the network's available power capacity and stations' occupancy limits. To tackle the problem at scale while retaining high solution quality, a data-driven optimization framework combining techniques from the fields of Deep Learning and Approximation Algorithms is introduced. The framework's key ingredient is a novel input-output processing scheme for neural networks that allows direct extrapolation to problem sizes substantially larger than those included in the training set. Extensive numerical simulations based on synthetic and real-world data traces verify the effectiveness and superiority of the presented approach over two representative scheduling algorithms. Lastly, we round up the contributions by listing several immediate extensions to the proposed framework and outlining the prospects for further exploration.

This work was supported by the Khalifa University of Science and Technology under Award No. CIRA-2020-286.

e-Energy '23, June 20-23, 2023, Orlando, FL, USA

© 2023 Copyright held by the owner/author(s).

# CCS CONCEPTS

• Theory of computation → Packing and covering problems; Algorithm design techniques; Discrete optimization; Scheduling algorithms; • Hardware → Smart grid; • Computing methodologies → Neural networks.

# **KEYWORDS**

Electric Vehicles, Charging Scheduling, Deep Neural Networks, Algorithm Design, Combinatorial Optimization, Smart Grid.

#### **ACM Reference Format:**

Bushra Alshehhi, Areg Karapetyan, Khaled Elbassioni, Sid Chi-Kin Chau, and Majid Khonji. 2023. DCLEVERNET: Deep Combinatorial Learning for Efficient EV Charging Scheduling in Large-scale Networked Facilities. In *The 14th ACM International Conference on Future Energy Systems (e-Energy* '23), June 20–23, 2023, Orlando, FL, USA. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3575813.3595205

## **1 INTRODUCTION**

Recent years have witnessed a surge in the adoption of electric vehicles (EVs) as eco-conscious and economical alternatives to combustion engine vehicles. Case in point, over one million EVs were sold in China in 2018 alone [42], and it is estimated that by 2040, approximately 700 million EVs will hit the roads worldwide [27]. Smart Grid (SG) technologies can facilitate EV integration and allow utilizing their elasticity for real-time load regulation purposes. However, even with a moderate EV population, the excess demand for charging power might place local distribution circuits under critical strain, leading to potential stability issues and deteriorated efficacy [5, 32]. In fact, as demonstrated in [5, 32], just 10% EV penetration rate in a residential distribution network suffices to cause inordinate voltage deviations, branch congestion, and overheating of substation transformers, leaving the power system equipment with shortened lifespan.

While the increased energy demand can be supported via gradual capacity upgrades, the sudden spikes in EV and non-EV loads require expensive fast generators as a backup, entailing high power losses. Coordinated, or "smart", charging strategies can mitigate

<sup>\*</sup>Both authors contributed equally to this work.

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *The 14th ACM International Conference on Future Energy Systems (e-Energy '23), June 20–23, 2023, Orlando, FL, USA*, https://doi.org/10.1145/3575813.3595205.

these issues by exploiting the flexibility of EV users' charging reservation requests (e.g., at which station and/or time period to charge). By optimizing the charging schedule based on these requests, coordinated charging can improve energy utilization while still meeting the needs of EV owners. Such schemes can be deployed in public parking sites, workplaces, shopping malls, residential complexes and would be particularly effective for large-scale charging facilities.

To unlock this tremendous demand-side management potential, SG operators require *efficient* and *scalable* means to tackle largescale scheduling problems. The optimization involved is time-coupled and *combinatorial* in nature: deciding which EV may charge at which station/rate and time given the available net power supply over the scheduling horizon. Unlike convex programming, which can be readily handled by off-the-shelf numerical solvers (e.g., Cplex or Gurobi), combinatorial (discrete) optimization is generally notorious for being computationally expensive. To exemplify, Fig. 1 plots the execution time of Gurobi when applied to the combinatorial EV charging reservation management problem at hand (dubbed EVCRP and formalized in Sec. 3). As observed from Fig. 1, with growing problem size, the solver's running time quickly turns prohibitive and may vary significantly depending on the input data.

Thanks to the inherent massive parallelism and powerful learning capacity (of approximating sophisticated non-linear mappings from labeled data), Deep Neural Networks (DNNs) have gained traction as an increasingly viable method to streamline decisionmaking in complex large-scale engineering systems [14, 16, 25]. Specifically, problem instances routinely solved in practice often share similar patterns or stem from related data distributions, which DNNs can exploit and learn to imitate the known optimal/nearoptimal iterative algorithms, thereby dramatically boosting the computing speed. However, the application of DNNs to constrained optimization problems confronts with a number of hurdles. First, the solutions predicted by DNNs may violate the problem constraints. Second, since the number of neurons in DNNs' input layer is normally predetermined, even a tiny increase in the problem size (e.g., when several new EV users subscribe to EVCRP program) will necessitate retraining the model<sup>1</sup>. Last but not least, obtaining training sets for large-scale combinatorial problems with tens of thousands of decision variables could prove computationally intractable unless one accepts crude approximations.

To address the aforementioned concerns, we introduce a Deep Combinatorial Learning framework, coined as DCLEVERNET, that can produce *close-to-optimal feasible* solutions to large-scale EVCRP instances in sub/near-linear time. Notably, the proposed approach conforms to EVCRP *inputs of arbitrary cardinality* without requiring to retrain or alter the network structure (namely, the number of neurons and connecting weights) for each input size, while also maintaining satisfactory performance as the count of participant EVs continues to rise. In summary, the current work complements and advances the existing research with the following three-fold contributions:

 By drawing on ideas from the toolbox of algorithmic techniques, we devise a computationally conducive input-output (IO) representation scheme for neural networks allowing to

Alshehhi and Karapetyan, et al.



Figure 1: Average computational time (across 10 runs) taken to solve randomly generated EVCRP instances (considering 3 charging stations) with the Gurobi optimizer against the input size at 95% confidence interval. Outliers are plotted as crosses.

tackle packing (and covering) combinatorial problems in a *scalable* and *efficient* fashion. With this scheme in place, a neural network trained on small problem sizes can be *directly invoked* (without retraining) to generate approximate solutions for the problem in high dimensional spaces. Furthermore, the adaptive structure of the scheme allows tuning the *granularity* of predicted optimization decisions to govern the trade-off between complexity and performance of the learning process (in a sense resembling an approximation scheme).

- (2) We assemble a sample DCLEVERNET model<sup>2</sup> consisting of a feed-forward DNN<sup>3</sup> augmented by the proposed IO scheme and complemented by a simple (sorting based) and fast post-processing routine for extracting feasible, approximate solutions to EVCRP. Leveraging the scalability of DCLEVERNET, we employ the model trained on the inputs of EVCRP problem for 1500 users (4500 binary decision variables) to produce solutions to instances with up to 10,000 users (30,000 binary decision variables), achieving on average *nearly* 80% of the optimal objective value.
- (3) To consolidate the contributions, the performance of the featured approach is scrutinized extensively under various case studies based on both synthetic and *real-world data traces* (acquired from Caltech's Adaptive Charging Network (ACN) [22]). First, we demonstrate the scalability and efficiency of DCLEVERNET by contrasting it against two benchmarks: (i) An adapted version of the polynomial-time approximation scheme (PTAS) recently developed in [19]; (ii) A greedy baseline strategy prioritizing higher-valued requests. Subsequently, we investigate the generalisability and robustness of DCLEVERNET to out-of-sample and out-of-distribution inputs generated by perturbing the values of EVCRP parameters (e.g., the available power capacity or the number of charging slots in stations).

<sup>&</sup>lt;sup>1</sup>When the input data is straightforwardly fed into DNN.

<sup>&</sup>lt;sup>2</sup>The trained model along with its sample application tutorial can be accessed online at https://drive.google.com/drive/folders/17miO6eaIxDqYmtXGrPQHV1brnjU81owh.
<sup>3</sup>Note that the framework is applicable beyond DNNs and can incorporate more complex architectures, such as Convolutional Neural Networks or Transformers as elaborated in Sec. 6.

The remainder of this paper is organized as follows. Sec. 2 reviews the related literature on the EV charging scheduling problem. Sec. 3 formulates the problem mathematically. In Sec. 4, we lay out the proposed Deep Combinatorial Learning approach, and in Sec. 5 validate its performance through extensive simulation studies. Lastly, Sec. 6 sketches several immediate extensions for future work, followed by concluding remarks in Sec. 7.

# 2 RELATED WORK

As surveyed in [4, 29, 39], a considerable body of literature has been published on controlled EV charging, proffering a rich arsenal of techniques to cater for various operational objectives, such as relieving power system congestion [1], maximizing EV owners' convenience [38], minimizing charging expenses [11], enhancing voltage profile [6], valley filling [12, 38], to name a few. From a methodological standpoint, the existing approaches can be broadly categorized into three groups: approximation algorithms, heuristics/metaheuristics, and Deep Learning (DL) based methods.

Approximation algorithms are relatively simpler and faster than exact solution methods, such as branch-and-bound and dynamic programming. The salience of these algorithms manifests in provable optimality guarantees quantified by approximation ratio, which measures the performance gap against the optimal objective value over all possible input realizations (i.e., in the worst possible scenario). In [2], Alinia et al. formulate EVCRP as a 0-1 programming problem that aims to maximize the total revenue obtained from charged EVs while respecting local and global peak power constraints. Assuming all EVs start charging simultaneously, the authors develop a primal-dual scheduling algorithm with a bounded approximation ratio. Towards a more realistic heterogeneous scenario, Majid et al. [19] devised a  $(1 - \epsilon)$  polynomial-time approximation scheme (PTAS), where  $\epsilon \in (0, 1)$  parameterizes the desired approximation ratio and the running time, to solve EVCRP for constant-length scheduling horizons. Though theoretically significant, these methods could be of limited practicality as improvements in solution quality typically come at the expense of scalability.

As such, heuristics/meta-heuristics are devoid of any optimality guarantees, yet often prove practically valuable when applied to combinatorial problems. The study in [28] formulates EVCRP as a multi-objective optimization problem to maximize EV users' revenue (by minimizing the charging cost and duration while also increasing the supplied energy). An evolutionary-inspired heuristic search algorithm, Ant Colony, is invoked to attain scalability. However, the setting in [28] does not account for the power network capacity constraint. Taking a step further, Liu et al. [24] consider an extended formulation incorporating the power constraints. Yet, a somewhat restrictive assumption is imposed that EVs must stop in multiple charging stations to fulfill their power demand. On the other hand, Sun et al. [38] frame EVCRP as two consecutive problems, where the first schedules the requests to achieve a flat load profile, whereas the second, which is combinatorial, aims to minimize interruptions during charging to preserve EVs' battery life. The authors provide a simple heuristic algorithm and evaluate its performance empirically through simulations.

Different from the above two, DL-based approaches can learn representations of data and then capitalize on the inferred information to guide or aid the decision-making process. In general, the existing learning-based approaches to optimization problems fall under two themes: end-to-end (i.e., standalone) and hybrid. Studies focusing on the former, e.g., [7, 31, 34, 44], utilize DL techniques to predict the solutions directly (without solving the optimization problem) by mapping the input data to the desired output (e.g., the optimal solution). For instance, the work in [44] constructs a DNN to learn the optimal load-generation mapping in the DC Optimal Power Flow problem. Such methods can capture complex relationships between the input data and the output, yet may not be able to exploit the problem's structural properties. In contrast, hybrid approaches pair known traditional optimization methods or numerical solvers with machine learning techniques [8, 13, 15, 40, 41]. This allows incorporating problem-specific knowledge/techniques to improve the computational efficiency of these algorithms or speed up the solvers. For example, Xu et al. [41] empirically enhance dynamic programming with DNNs to tackle combinatorial optimization problems in a computationally more efficient manner. Gasse et al. [13], on the other hand, present a Graph Convolutional Neural Network model to speed up mixed-integer linear programming solvers by automatically learning on which variables to branch.

Among various DL techniques, DNNs [21] are by far the most prevalent and mature models [14]. Typically, DNNs comprise a series of layers stacked on top of each other, which collectively seek to approximate complex non-linear mappings. The universal approximation capability [16] coupled with the coveted computational efficiency renders DNNs particularly suited for large-scale optimization tasks. This stimulated application of DNNs to smart charging of EVs [3, 23, 26, 35, 36] (see [43] for a recent review) as well as to power system problems in general [10, 44]. The study in [3] proposed to utilize Deep Reinforcement Learning within a dynamic pricing framework as an alternative to the traditional on-peak and off-peak pricing. Some prior research studies considered a simplified variant of EVCRP, where the problem is reduced to a classification task of predicting the EV charging rate, charging location, or the mode of charging (i.e., charging or discharging) [35, 36]. In [36], the authors compare the performance of various machine learning approaches, including Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Long Short-Term Memory (LSTM) and DNN. The developed method relies on two separate classifiers for predicting the charging rate and the charging location, respectively. For both classifiers, RF and LSTM achieved the highest accuracy. However, it is important to note that casting EVCRP as the said classification task limits the network operator functionalities, hence the demand-side management potential.

To reduce charging costs, Lopez et al. [26] develop a DNN-based strategy that can determine the optimal EV charging periods in response to real-time electricity price signals. The model was trained with historical data of charging sessions as inputs and the corresponding optimal solutions (calculated by Dynamic Programming) as prediction labels. In [23], Li et al. additionally consider the effect of discharging operations and propose a reinforcement learning framework wherein policies are optimized by invoking DNNs. Nevertheless, for constrained optimization problems, which are a departure from those studied in [23, 26], DNNs *do not necessar-ily* guarantee a feasible solution due to lacking representation of the exact feasibility region as well as their inherent and inevitable approximation errors. Recently, Zhao et al. [44] proposed a DNN-based scheme for DC Optimal Power Flow problem that assures the feasibility of output generator set-points. The underlying idea is to "preventively" adjust (perturb by an appropriate magnitude) the constraints during the training stage, thereby anticipating approximation errors and warding off potential violations in the testing phase. While beneficial on its own, this method does not seem amenable to combinatorial optimization problems with binary/integer variables.

# **3 PROBLEM STATEMENT**

Recall that in the problem under study, the charging network operator seeks to determine binary scheduling assignments (reflecting the accept/reject decisions) for the charging reservation requests of subscribed EV users containing the preferred charging stations (CSs), intervals, and valuations. We assume a centralized control scheme wherein the requests are elicited in an apriori manner (e.g., a day ahead); hence, all users' profiles are accessible beforehand.

In the said network, embedded within and supplied by a power distribution grid, CSs are indexed by the set *C* and distinguished by their charging power rate  $r_c \in \mathbb{R}, c \in C$ . We consider a time-slotted system model in which the scheduling horizon  $\mathcal{T} \triangleq \{1, \ldots, T\}$  is discretized into *T* equidistant intervals according to the desired frequency of control signals. In the set of participant EVs, denoted by  $\mathcal{A}$ , each customer  $a \in \mathcal{A}$  lodges a charging request which includes  $(C^a, \{\mathcal{T}_c^a\}_{c \in C^a})$ , where  $C^a \subseteq C$  is the set of favored CSs and  $\mathcal{T}_c^a \subseteq \mathcal{T}$  are the corresponding preferred charging periods, in order to fulfill their charging power demand  $P^a = P_c^a \triangleq r_c |\mathcal{T}_c^a|$  for  $\forall c \in C$  (Note that the power demand is invariant across CSs). Accordingly, shall user *a*'s request be accepted, the scheduler will *select one among a*'s preferred options and reserve the corresponding electric vehicle supply equipment (EVSE) for the requested duration. Let  $x_c^a$  denote the scheduling decision for user  $a \in \mathcal{A}$ , then

$$x_c^a = \begin{cases} 1, & \text{If user } a \text{ is assigned to charge at the CS } c \in C^a \\ 0 & \text{Otherwise} \end{cases}$$

To model users' welfare, define the *gain* for each user  $a \in \mathcal{A}$  with respect to the charging decision  $x_c^a$  as

$$G^{a}(x^{a}_{c}) \triangleq u^{a}x^{a}_{c} - \sum_{t \in \mathcal{T}^{a}_{c}} \operatorname{cost}(t) \cdot x^{a}_{c}, \qquad (1)$$

where  $\cot(t)$  refers to the time-varying electricity  $\cot tand u^a$  is a *user-defined* parameter included in the charging request that measures the *utility* perceived by customer *a*. Here, utility quantifies the extent of user-obtained comfort or, alternatively, the *worthiness* (valuation) of receiving the requested charging power. In this context,  $G^a(x_c^a)$  can be interpreted as the savings gained by user *a* from participating in the reservation program. Without loss of generality, we suppose that  $u^a$  is sufficiently large so that  $G^a(x_c^a)$  is non-negative for  $\forall a \in \mathcal{A}, c \in C$ . For brevity, in what follows, we shall write  $G^a$  to denote users' gain if the charging request is satisfied (i.e.,  $x_c^a = 1$  for some  $c \in C^a$ ) and henceforth refer to  $G^a$ 

as user *a*'s *conditional gain*. Also, we define  $R \triangleq \frac{G^a}{P^a}$  to be user *a*'s *conditional gain-to-power ratio*.

To account for practical aspects, we assume that in addition to the EV load, the network supplies energy to residential and commercial consumers, referred to as background active power demand or *base load* and denoted by  $d(t) \in \mathbb{R}$  at time  $t \in \mathcal{T}$ . Additionally, to cater for physical limitations, we bound the total power capacity of the network by  $\overline{P}$  and cap the occupancy limit of each CS  $c \in C$  by  $N_c$  corresponding to the number of installed EVSE.

With the above notation, EVCRP translates into the following combinatorial optimization problem:

$$\max_{x_c^a} \quad \sum_{a \in \mathcal{A}} \sum_{c \in C^a} G^a(x_c^a) \tag{EVCRP}$$

s.t. 
$$d(t) + \sum_{a \in \mathcal{A}} \sum_{c \in C^a : t \in \mathcal{T}_c^a} r_c \cdot x_c^a \le \overline{P}, \quad \forall t \in \mathcal{T}$$
 (2)

$$\sum_{e \subset a} x_c^a \le 1, \quad \forall \ a \in \mathcal{A} \tag{3}$$

$$\sum_{c \in C^a, \ t \in \mathcal{T}_c^a} x_c^a \le N_c, \quad \forall \ c \in C, t \in \mathcal{T}$$
(4)

$$x_c^a \in \{0,1\}, \quad \forall \ a \in \mathcal{A}, c \in C.$$
 (5)

EVCRP seeks to maximize users' total welfare gain while considering both local and global peak constraints in the networked CSs. Constr. (2) stipulates the total supplied power to remain below the network's maximum capacity. Constr. (3) ensures that each user is assigned only to one CS. Constr. (4) imposes a limit on the number of customers who can charge at a particular CS at any given time so that it does not exceed the maximum number of EVSE. Finally, Constr. (5) enforces the integrality of decision variables.

The crux of solving EVCRP lies in its combinatorial structure imposed by the binary decision variables  $X \triangleq (x_c^a)_{a \in \mathcal{A}, c \in C}$ . As such, EVCRP specializes to several classical NP-hard combinatorial problems, including the two-dimensional Knapsack problem (when  $|\mathcal{T}| = |C| = 1$ ) and the Unsplittable Flow on a Path problem (when  $|\mathcal{C}| = 1$  and Constr. (4) is ignored). This rules out EVCRP's NP-hardness and hints that it is substantially more complicated than the latter two problems. Given these facts, designing scalable and near-optimal scheduling algorithms for EVCRP becomes markedly challenging. In the proceeding section, through a judicious combination of algorithmic and DL techniques, we devise a Deep Combinatorial Learning framework that, on average, can closely approximate the optimal solutions of large-scale EVCRP instances in linear time.

# 4 PROPOSED APPROACH

с

a

This section introduces DCLEVERNET and provides a detailed description of its architecture. First, we outline the methodology of the proposed framework and highlight the challenges inherent to designing a learning-based approach for combinatorial optimization problems, and how the framework addresses these hurdles. The framework's structure is comprised of three modules: (1) Preprocessing step that downsamples (compresses) EVCRP input into a succinct representation; (2) DNN for predicting the distribution of users in the optimal solution; (3) Fast post-processing procedure to extract a feasible solution from the prediction. Each of these components is thoroughly analyzed and explained in the following subsections.

**Notational Convention:** In the remainder of this paper, unless otherwise explicitly stated, we shall enclose the inputs of procedures, such as counting or averaging, within the operator []. Given a vector/set v we let |v| symbolise its magnitude/size and we write  $\tilde{v}$  to denote the normalization of the elements in v by a certain value.

## 4.1 Overview and Challenges

As illustrated in Fig.2, which depicts the high-level schematics of DCLEVERNET, the methodology can be broken down into two distinct phases: training and prediction. During the training phase, input instances of EVCRP and their corresponding solutions (from which the prediction labels are constructed) are fed to the neural network. The primary objective of the training phase is for the DNN to acquire the ability to predict solutions for previously unseen instances of EVCRP during the prediction phase.

A significant amount of research has been conducted to explore the application of DNNs for solving combinatorial optimization problems (COPs). One of the challenges encountered is adequately representing the optimization problem as features and labels (i.e., input and output of the DNN). COPs often involve a large number of variables and complex constraints and objectives, making it difficult to encode an appropriate input representation for the network. Despite the robust nature of DNNs, poorly represented input can affect their performance.

A well-defined input reduces the number of parameters to be learned by the network, thus resulting in a more efficient training process. Furthermore, compact inputs can also help reduce overfitting, which is a common problem in DNNs. Overfitting occurs when a model performs well on the training data but poorly on unseen data. This is because the model has learned to fit the noise and redundancy in the training data rather than generalizing to new examples.

The main challenge in remodeling a COP solution as a label is the discrete format of the data representation. Applying this to deep learning models that work best with continuous data can be challenging. In addition, encoding the optimal solution as the output of a DNN is not a recommended practice due to its inherent sparsity. The optimal solution to many real-world problems is often sparse. In such cases, encoding the solution directly into the



Figure 2: High-level block diagram of the featured DLassisted optimizer enabling rapid computation of nearoptimal solutions to EVCRP.

output layer of a DNN would result in a large number of parameters, many of which would have little impact on the solution. This would render the model difficult to train and could lead to overfitting.

In view of the above, the authors in [26] designed a DNN that outputs decisions for a single EV at a time. Given only the current state of the battery, the output is a single variable to indicate whether to admit the user or not. The downside of this approach is that the decisions are (locally) made without considering the entire set of EV requests, which could lead to highly suboptimal global solutions. The study in [17] addresses the issue by utilizing the capability of DNN to predict a partial solution to the optimal power flow problem (OPF). Then, the final solution is constructed by solving OPF initialized with the DNN output. This approach is suitable for small-scale instances since it requires numerical solvers for constructing the final solution and is hence limited by the capabilities of these solvers.

Feeding DNNs with a feasible solution is not sufficient to guarantee the feasibility of output solutions. Therefore, the authors in [30] propose incorporating a penalty term in addition to the mean square error (MSE) loss function. However, it is still possible that the obtained solution will not satisfy all the constraints. Therefore, a post-processing step, in which the nearest feasible solution is chosen greedily, was employed. To ensure feasibility, in [44], the authors improve the approach by calibrating the penalty to be enforced when the constraints are within a percentage of the violation value. While this approach shows some promise, it is not trivial to tune the calibration for optimal performance. In addition, for EVCRP, calibration of constraints may lead the network to learn a significantly suboptimal solution due to the discreteness of the problem.

Importantly, all of the aforementioned studies *require re-training* if the input size (e.g., number of EV requests) is varied. The present work introduces a novel approach that overcomes the above limitations. In DCLEVERNET, we tackle the encoding of EVCRP inputs and outputs through a judiciously constructed scheme (detailed in Sec. 4.2) that enables *scalability*. Unlike existing methods in the related literature, we model the DNN input as groups of EV charging requests. This parses the input size invariant to the number of EV charging requests. The output of the network is the optimal distribution of allocated users with respect to groups, thus allowing the network to be able to handle large-scale instances. The feasibility is assured through a lightweight post-processing routine that extracts the solution from the predicted distribution (detailed in Sec. 4.4).

In the current study, we utilize a training dataset labeled with optimal solutions obtained from the Gurobi solver. However, for certain combinatorial optimization problems, such as the Traveling Salesman Problem, it may be computationally infeasible to find the optimal solution when the problem size is large. An alternative approach is to construct the training data based on near-optimal solutions obtained from known efficient approximation algorithms. By training the model on these solutions, the network can learn to imitate the underlying principles of the utilized approximation algorithm, thereby unlocking substantial speedup gains.

#### e-Energy '23, June 20-23, 2023, Orlando, FL, USA



Figure 3: Detailed architecture of the constructed sample DCLEVERNET model comprised of a DNN augmented by the proposed input-output processing scheme. For scalability, in the pre-processing step, EVCRP input is downsampled via grouping and averaging to allow succinct representation. The output layer predicts the number of users within groups in EVCRP's optimal solution.

# 4.2 Pre-processing

Recall that EVCRP's input consists of EV users' charging reservation requests characterized by the set of favored CSs, the corresponding preferred charging intervals, and conditional gains associated with each request. As seen from Fig. 3, we first partition the requests by the requested charging station into |C| groups. For each station  $c \in C$ , the conditional gains for users in the respective group are normalized by dividing by the maximum value within that group. Then, the requests in each group are further classified into Q groups based on their normalized conditional gains in descending order. In particular, the set of users  $q_i$  in the  $i^{th}$  group is as follows:

$$q_i \triangleq \left\{ a : \frac{i-1}{Q} < \tilde{G}^a \le \frac{i}{Q}, a \in \mathcal{A} \right\}, \quad \forall \ i = 1, \dots, Q, \quad (6)$$

where  $\tilde{G}^a \triangleq \frac{G^a}{\max_{a \in q_i} G^a}$  denotes the conditional gain of user *a* 

normalized by the maximum conditional gain among the users in  $q_i$ . The rationale underlying this clustering criterion is to produce groups wherein the difference between users' conditional gains is bounded. In other terms, the users within a group share roughly "similar" conditional gains and can be represented by a "centroid" user. This, in turn, allows for decision-making at a group level rather than at an individual level without inflicting a significant loss in the total gain of the selected customers. Such grouping schemes have been frequently employed in theoretical computer science literature to devise approximation algorithms for packing optimization problems. For instance, a similar method in which the users are discerned into a logarithmic number of groups wherein their utilities differ by no more than a constant factor was employed in [9, 18, 20] to devise approximation algorithms for different variants of the Unsplittable Flow on Paths and r-weighted Minimization Knapsack problems. This method provides a systematic and efficient manner of downscaling the inputs of packing/covering COPs. Note that the partitioning criteria can be adapted based on the problem type. In addition to conditional gains, users can be

alternatively clustered based on their utility, power demand, or utility-to-demand ratio.

For DNN to learn the optimal scheduling of EV charging requests effectively, the network must gain a comprehensive understanding of the constraints and parameters of EVCRP. In this regard, as illustrated in Fig. 3 we include the normalized load profile  $\tilde{d}(t)_{t\in\mathcal{T}}$  in the input layer. This intends to convey the congestion status of the power network at any given time slot. Each group  $q_i$ is represented by four key input entities that summarize the information of users in  $q_i$ :

(1)  $Avg[(\tilde{P}^a)_{a \in q_i}]$  - The average  $\tilde{P}^a \triangleq \frac{P^a}{\overline{P}}$  of  $\forall a \in q_i$ , where

 $\tilde{P}^a$  is user *a*'s power demand normalized by the capacity of the power network  $\overline{P}$ .

- (2)  $\tilde{NOU}[q_i]$  The group size normalised by the total number of users  $|\mathcal{A}|$ , i.e.,  $\tilde{NOU}[q_i] \triangleq \frac{|q_i|}{|\mathcal{A}|}$ ;
- (3) Avg[(G̃<sup>a</sup>)<sub>a∈qi</sub>] The average G̃<sup>a</sup> (as defined above) of all the users in group q<sub>i</sub>;
- (4)  $Avg[(\tilde{R}^a)_{a \in q_i}]$  The average  $\tilde{R}^a \triangleq \frac{R^a}{\max_{a \in (q_i)_{i=1,\dots,Q}} R^a}$  of

the users in  $q_i$ , where  $\tilde{R}^a$  is user *a*'s conditional gain-topower ratio normalized by the maximum ratio among all the users in the corresponding CS.

In addition to the aforementioned input entities, we also include for each charging station  $c \in C$  the normalized number of charging requests in each time slot. The normalization was done with respect to the maximum number of requests at any time slot in the corresponding CS. The adopted DNN can benefit from this information by learning the temporal dynamics of the charging requests, which can aid in identifying the peak charging periods. We note that input normalization can play a crucial role in the efficiency of the learning process as well as aid in preventing overfitting. As transpires from the simulation results presented in Sec. 5, taken together, these input entities allowed the trained sample DCLEV-ERNET model to extract transferable knowledge about the structure and parameters of EVCRP.

One of the key design considerations when encoding the EV requests as proposed is determining the number of groups *O*. The latter corresponds to the input data's granularity; thus, increasing the number of groups will result in a more detailed representation of the input. This can be beneficial for applications for which accuracy is of crucial importance. However, to ensure efficient training, the size of the training data should be increased. Conversely, decreasing the number of groups compresses the input resulting in a more general representation of the input. The advantage of this is that the DNN will identify the most salient features of the input data, and the compressed representation will highlight the most important information. Aside from the representation, the number of groups governs the number of parameters in the model to be trained, hence directly affecting the required computational resources. Therefore, the input dimensionality should be adjusted considering the trade-off between the desired accuracy and the available computational resources.

#### 4.3 Network Architecture

As illustrated in Fig.3, we adopt a multi-layer feed-forward neural network architecture of the form:

$$Y_0 = \left[\tilde{d}(t)_{t\in\mathcal{T}}, \chi^{CS^1}, \dots, \chi^{CS^{|C|}}\right], \tag{7}$$

$$Y_i = \sigma\Big(W_i Y_{i-1} + b_i\Big), \quad \forall \ i = 1, 2, \dots, H$$
(8)

$$\hat{Y} = \sigma' (W_{H+1}Y_H + b_{H+1}),$$
 (9)

where  $\chi^{CS^1}$  encodes the input entities for CS 1, as depicted in Fig. 3 and detailed above,  $Y_0$  and  $\hat{Y}$  are the input and output layers, respectively, while  $Y_i$  is the output vector of the *i*-th hidden layer (out of *H* in total) which depends on the weights  $W_i$ , biases  $b_i$ , and the output of the previous layer  $Y_{i-1}$ . In Eqns. (8) and (9),  $\sigma(.)$  and  $\sigma'(.)$  stand for the Rectified Linear Unit (ReLU) activation functions used in the hidden layers and the linear activation function for the output layer, respectively. ReLU allows the network to learn non-linear relationships in the data, and the linear activation function allows the output layer to produce continuous values that represent the number of users in each group. In the present study, we construct a DNN with nine hidden layers, thus H = 9. The number of neurons in the first hidden layer is 800, gradually decreasing to 50 as the layers become deeper. As a loss function, we utilize the mean squared error (MSE):

$$\mathcal{L}_{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left( Y^{i} - \hat{Y}^{i} \right)^{2}, \tag{10}$$

where *n* is the number of samples,  $\hat{Y}^i$  is the prediction of the network for sample *i*, and  $Y^i$  is the corresponding ground truth value. The selection of MSE as a loss function rests on extensive experimentation with various alternatives, including Huber Loss, mean percentage error, and mean absolute error. In particular, the experimentation results revealed that MSE provided the most accurate and reliable performance.

Similar to the grouping in the input layer, the EV requests in the output are grouped by the requested charging station into |C|groups. For each station  $c \in C$ , the requests are partitioned into Lgroups based on  $\tilde{G}^a$  in descending order. Let  $l_i$  be the set of users in the *i*-th group, then

$$l_i \triangleq \left\{ a : \frac{i-1}{L} < \tilde{G}^a \le \frac{i}{L}, a \in \mathcal{A} \right\}, \quad \forall \ i = 1, \dots, L.$$
 (11)

For further granularity, in each group  $l_i$ , the power demand is normalized by dividing over  $\overline{P}$ , and the requests are partitioned into V groups based on their normalized power demand in ascending order. Denote by  $v_i$  the set of users in the *i*-th group such that

$$q_i \triangleq \left\{ a : (1 - \frac{i}{V}) < \tilde{P}^a \le \frac{V - i + 1}{V}, a \in \mathcal{A} \right\}, \quad \forall \ i = 1, \dots, V,$$
(12)

where  $\tilde{P}^a$  denotes the normalized power demand of user  $a \in \mathcal{A}$ .

The groups have been arranged so that the first group comprises users with the highest conditional gain and the lowest power demand. The output layer denoted by  $\hat{Y}$  consists of  $|\hat{Y}| = (|C| \cdot L \cdot V)$ neurons. Each output neuron in  $\hat{Y}$  represents the number of users to include in the final solution from each group.

# 4.4 Post-processing

As previously noted, the network learns the distribution of user categories in the optimal solution thus requiring post-processing to extract the final solution of EVCRP. For the purposes of the current study we employ a greedy approach explained in Alg. 1.

The output of the DNN,  $\hat{Y}$ , representing the number of users to be assigned to each group j, serves as an input to Alg. 1. The algorithm proceeds by rounding down the elements of  $\hat{Y}$  to their nearest integer values. For each group j, we define  $\mathcal{M}$  as the set of prospective users in group *j* sorted in descending order of their  $G^a$ . Then, the algorithm iterates over  $\mathcal{M}$ , selecting the users unless the solution is deemed infeasible. The feasibility of the solution is contingent upon constraints (2) to (5). Constr. (2) is violated if the maximum network capacity is exceeded at any point in time. Constr. (3) requires a user to be assigned to only one charging station. Constr.(4) is violated if the number of assigned users in any station exceeds  $N_c$ . Lastly, as the procedure either accepts or rejects users, constraint (5) is ensured to be met. In the event that incorporating a user leads to an infeasible solution, their charging request will be declined, and the algorithm continues to the next user in  $\mathcal{M}$ . The algorithm terminates when either the proposed number of users to be added to group *j* is zero or all the users in the group have been evaluated.

The number of groups  $|\hat{Y}|$  (i.e., the number of neurons in the output layer) is an important hyperparameter that can have a significant effect on the performance and accuracy of the model. Recall that  $|\hat{Y}|$  can be manipulated by varying *L* and/or *V*. The accuracy of the model can be positively affected by the increase in the number of groups, since the search space of Alg. 1 is reduced. Yet, an excessive number of groups (e.g., comparable to the number of EV users), hence output neurons, will likely result in highly sparse predictions leading to increased complexity and a higher chance of overfitting. These aspects combined can degrade the performance severely. Likewise, reducing the number of groups expands the search space thus requiring more computational power

e-Energy '23, June 20-23, 2023, Orlando, FL, USA

Algorithm	Post-processing	Routine
-----------	-----------------	---------

Input:  $\hat{Y}$ Output: Solution X to EVCRP 1:  $\hat{Y} = |\hat{Y}|$ 2: for  $j \in \hat{Y}$  do  $\mathcal{M} \leftarrow$  Users in group *j* sorted in descending order of their 3: utility values. 4: NOU  $(j) = |\mathcal{M}|$  $\beta \leftarrow \hat{Y}(j)$ 5:  $\eta \leftarrow 0$ 6: while  $\beta \neq 0$  AND  $\eta \leq$  NOU (j) - 1 do 7: 8:  $X(\mathcal{M}_{\eta}) = 1$ if X is not feasible then 9: 10:  $X(\mathcal{M}_n) = 0$ 11: else  $\beta = \beta - 1$ 12: end if 13: 14:  $\eta = \eta + 1$ end while 15: 16: end for

during post-processing. Consequently, adjusting this number can improve the predictions made by the model, but it is important to consider the trade-offs between computational efficiency and performance. Ultimately, the optimal number of neurons in the output layer should be determined through empirical experimentation and evaluation using appropriate evaluation metrics.

To guarantee a feasible solution to EVCRP, Alg. 1 verifies the constraints' violations in line 9. The computational complexity of this post-processing routine depends on the number of groups and the sorting procedure within each group. Assuming that the number of users in each group is constant (i.e.,  $<< |\mathcal{A}|$ ), the computational time amounts to  $O(|\hat{Y}|+|C|\cdot|\mathcal{T}|)$ . In the unlikely worst-case scenario where DCLEVERNET forecasts a solution instance with all the users partitioned into a single group, the computational complexity would be  $O(|\mathcal{A}| \log |\mathcal{A}| + |\mathcal{A}| \cdot |C| \cdot |\mathcal{T}|)$ .

## **5 EVALUATION STUDIES**

In this section, the performance of the proposed approach is evaluated by simulations. The considered criteria are the approximation ratio and the running time.

## 5.1 Simulation Setup and Settings

Distribution network setting: The capacity of the power substation, denoted by  $\overline{P}$ , is set to be 1MW. The base-load, denoted by d(t), varies and is dependent on the time of the day. In our simulations, we adopt three base-load profiles as depicted in Fig. 4; each profile represents the daily average household load profile in the service area of South California from 00:00, January 1, 2011, to 23:59, January 3, 2011 [37]. Each EV has the option to charge in 3 charging stations with charging rates of 1.5kW, 7Kw, and 50kW, respectively. In each station, the number of installed chargers, denoted by  $N_c$ , is set to 200.



Figure 4: The considered three versions of base-load profiles based on 650 households.

*Scheduling horizon:* We consider a 24-hour time horizon divided into 96 slots of 15 minutes, i.e.,  $|\mathcal{T}| = 96$ .

Training and Testing data: For training the presented DCLEVER-NET model, 50, 000input samples were produced, to which an 80/20 split was applied to divide the data into training and test sets. The data is equally comprised of synthetic and real-world charging requests. The synthetic data are based on real-world patterns. According to [33], most EV users start charging when return home at 18:00, and more than 90% of EVs start charging between 13:00 and 23:00. Therefore, the start time can be modeled as a normal distribution with a mean of 18:00 and a standard deviation of 5 hours. We set the charging start time at random (following a normal distribution) and its length to the time needed to fulfill the energy requirement. The energy required for user  $a \in \mathcal{A}$  was formulated as follows:

$$P^a = \left(1 - \frac{\text{SOC}^a}{100}\right) \cdot B^a,\tag{13}$$

where SOC<sup>*a*</sup> and  $B^a$  are the state of charge and the battery size for user  $a \in \mathcal{A}$ .

The SOC is modeled as a truncated normal distribution with a mean of 0.5 and a standard deviation of 0.3. For each EV, the SOC is limited between 20% to 80%. Similarly,  $B^a$  is modeled as a normal distribution with a mean of 24kW and a standard deviation of 10kW, for  $\forall a \in \mathcal{A}$ . For real-world data, we are utilizing the Caltech Adaptive Charging Network dataset [22], which is a comprehensive collection of data related to the energy consumption patterns and charging behaviors of electric vehicles within a network environment. The dataset contains rich information about EV driving and charging patterns. We adopt the arrival time and the power demand of EVs on Tuesdays.

For inclusivity, this study evaluates two distinct settings for utility values. The first setting involves a linear utility, which can be expressed as:

$$u^a = 0.36 \cdot P^a \,, \tag{14}$$

where 0.36 is the peak electricity price. The second setting involves a random selection of utilities within the range from 5000 to 8000.

In the case studies performed, the energy cost is based on the time of use rate in South California [37]. For both the training and testing data, the non-EV load is chosen to be the load profile 1 pictured in Fig 4.

#### Benchmark algorithms:



(a) The average approximation ratio and running time of PTAS\*, DCLEVERNET [Q=4, L=10,V=10], and GreedyU on 250 EVCRP instances randomly taken from the synthetic testing dataset.



(b) The average approximation ratio and running time of PTAS\*, DCLEVERNET [Q=4, L=10,V=10], and GreedyU on 250 EVCRP instances randomly taken from the ACN testing dataset.

Figure 5: Performance comparison of DCLEVERNET, GreedyU, and PTAS\* algorithms on the synthetic and ACN datasets.

- PTAS<sup>\*</sup>: A PTAS is an algorithm that for any  $\epsilon > 0$  is guaranteed to produce a  $(1 - \epsilon)$ -approximation to a given maximization problem. The running time of a PTAS is polynomial in the input size for every fixed  $\epsilon$ , but the exponent of the polynomial might depend on  $\frac{1}{\epsilon}$ . In other words, a PTAS allows trading approximation ratio for the running time. As one benchmark, we implement an adapted version of the PTAS for EVCRP proposed in [19]. The adapted variant, denoted by PTAS\*, proceeds as follows. The algorithm considers randomly generated initial guesses for a small subset of users. For each guess, the remaining subproblem of EVCRP is solved with a numerical solver by relaxing the other discrete control variables to be continuous, then rounding these variables to obtain a feasible solution. The key difference between PTAS and PTAS\* is that the former iterates over all possible guess combinations in the search for the best approximation ratio, whereas the latter considers only a limited number of guesses. Even with a few users, the running time of PTAS in practice could be computationally prohibitive. Therefore, in PTAS\*, we restrict the number of guesses to 250.
- Greedy Utility Algorithm (GreedyU): As a baseline, we adopt a commonly utilized greedy strategy that sorts the EV users in A = {1,..., |A|} according to their conditional gain in descending order, such that

$$G^1 \ge G^2 \ge \ldots \ge G^{|\mathcal{A}|},\tag{15}$$

then selects the customers sequentially in that order, subject to feasibility constraints. Since GreedyU sorts and iterates over the entire customer set  $\mathcal{A}$ , the running time complexity is  $O(|\mathcal{A}| \log |\mathcal{A}| + |\mathcal{A}| \cdot |\mathcal{C}| \cdot |\mathcal{T}|)$ .

*Implementation of the DNN model:* The presented DCLEVERNET model was constructed on the basis of the Keras platform. The model was trained over 200 epochs with a batch size of 32. The learning rate was set to 0.001.

The simulations were evaluated on a desktop machine with Intel i7-8750 CPU 2.2GHz processor and 16 GB of RAM. The algorithms were coded in Python 2.7 programming language with SciPy and NumPy libraries for scientific computation.



Figure 6: The evolution of training and validation losses over the number of epochs trained.

## 5.2 Evaluation Results

To thoroughly evaluate the effectiveness of the assembled DCLEV-ERNET model, we performed a series of experiments utilizing outof-sample data from both synthetic and real-world data sets. In addition, in our evaluations, we examine the generalizability of the trained model by testing its performance on out-of-distribution data. This involved manipulating various factors, such as reducing the number of chargers  $N_c$ , increasing the number of EV charging requests, and varying the load profile. Through these experiments, we aim to assess the method's ability to adapt and make accurate predictions in new and unseen situations.



Figure 7: The average approximation ratio and running time of DCLEVERNET [Q=4, L=10,V=10] and GreedyU on 250 EVCRP instances randomly taken from the synthetic and ACN testing datasets.



Figure 8: The average approximation ratio and running time of DCLEVERNET [Q=4, L=10,V=10], and GreedyU on 200 randomly generated synthetic EVCRP instances with gradually increasing input size from 3000 to 10,000.

First, we analyze the behavior of the loss function over multiple epochs, as depicted in Fig.6. The results show a consistent decrease in the loss function for both the training and testing data. This indicates that the model is effectively learning to produce increasingly similar predictions to the corresponding labels. Next, we examine the approximation ratio (i.e., how close the predicted solution is to the optimal solution) and the running time of the model. The performance of DCLEVERNET was evaluated over 250 testing instances from the synthetic and ACN data sets, respectively. The comparative results depicted in Fig.7, clearly demonstrate that DCLEVERNET outperforms the GreedyU algorithm in terms of both approximation ratio and computational time. In addition, a comparison was made between DCLEVERNET and PTAS\*, and the results, plotted in Fig.5, indicate that the average case performance of DCLEVERNET is nearly on par with that of PTAS\* in terms of solution quality for both synthetic and ACN instances. As evidenced by Figs.5 and 7, for the ACN dataset the worst-case approximation ratio achieved by DCLEVERNET was higher than those of PTAS\* and the GreedyU method. Furthermore, DCLEVER-NET demonstrated superior computational efficiency, on average being the fastest in all the case studies conducted.

To demonstrate the scalability of the proposed approach, we tested the trained DCLEVERNET model (on instances with 1500 customers) on higher-dimensional problem instances with the number of EV requests ranging from 3,000 to 10,000. Each case was evaluated with 200 synthetic instances with random utilities. The results are depicted in Fig.8, which compares the performance of the proposed approach to that of the GreedyU algorithm. As observed from the figure, DCLEVERNET demonstrates a superior approximation ratio, and it's significantly faster than GreedyU across all instances. This indicates that the proposed approach can handle large-scale EV charging scheduling problems with acceptable performance without the need for re-training.

The load profile is an important parameter in the EV charging scheduling problem as it affects the distribution of the energy demand, and thus it is crucial to evaluate the model's performance under different load profiles. The performance of DCLEVERNET was evaluated on 200 instances with load profiles 2 and 3, which were not included in the training dataset. The results of this evaluation are presented in Fig.10. While the observed performance of the DCLEVERNET is lower than in the instances with load profile 1 (on



Figure 9: The average approximation ratio and running time of DCLEVERNET [Q=4, L=10,V=10] and GreedyU on 100 EVCRP instances considering modified number of charging slots ( $N_c = 100$ ).



Figure 10: The average approximation ratio and running time of DCLEVERNET [Q=4, L=10,V=10] and GreedyU on 200 EVCRP instances generated considering load profiles previously unseen during the training.

which it was trained), it still significantly outperforms the GreedyU algorithm in both solution quality and running time. Lastly, to further assess the robustness and generalizability of the constructed model, we test the performance of the trained DCLEVERNET model on 100 instances with a modified number of charging slots in each station:  $N_c = 100$  for  $\forall c \in C$ . As Fig 9 demonstrates, DCLEVERNET continued to perform superior to the GreedyU algorithm in terms of approximation ratio and computational efficiency. This signifies that the proposed approach is robust and can adapt to different load profiles while retaining satisfactory performance.

# 6 DISCUSSION AND PROSPECTIVE EXTENSIONS

While the performed extensive simulations demonstrate the effectiveness of the trained sample DCLEVERNET model, a number of further improvements can be attained to solidify the overall framework. We identify the following immediate extensions as well as prospective avenues for further exploration.

(1) One promising immediate extension is to augment the proposed framework by incorporating more complex neural network architectures. Specifically, the utilization of Transformer based architectures, which employ the self-attention mechanism, has demonstrated remarkable performance in a wide range of applications. The self-attention mechanism enables the network to focus on specific parts of the input by assigning different weights to different parts of the input, this allows the network to focus on the most relevant

parts of the input and improve the performance of the network. Another technique that can be considered is the integration of skip connections. These connections enable the flow of information to bypass certain layers in the network, which can improve the efficiency of the flow of information and improve the performance of the network. Combining these two and integrating them into DclEVERNET will allow training a substantially deeper network than the currently presented one, likely yielding drastically improved generalisability and performance.

- (2) Another immediate extension is to substitute the employed post-processing routine. Depending on the desired outcome, one can improve the computational efficiency or the performance quality. For example, towards the latter, one can adopt a variant of the Beam Search algorithm. Beam Search is a heuristic exploration algorithm that maintains a set of candidate solutions based on different algorithms and then returns the one with the highest objective value. For example, instead of relying only on one sorting procedure, the post-processing can simultaneously employ several sorting criteria based on power demand and/or gain-to-power ratio, then return the selection of users with the highest objective value. This would allow to explore the predicted solution space more extensively. On the other hand, Alg. 1 can be replaced by a faster vet possibly less efficient procedure. One example of such a method is the selection of users probabilistically via a random process. In this worst-case scenario, this would consume only linear running time.
- (3) In future work, the extension of this framework to include covering and mixed covering and packing problems will be explored. Covering problems involve finding a subset of items that can completely cover a set of resources, while mixed covering and packing problems involve both covering resources and packing items. The proposed framework can be extended to include these types of problems by creating a similar dataset as described in Sec. 4.2.
- (4) Added to the latter, another promising direction is to completely eliminate the need for post-processing, thus providing an end-to-end DL optimizer for combinatorial optimization problems. Though profoundly difficult, we believe that it is plausible to devise a standalone DL solver (possibly a two-stage approach combining two different DL techniques) which on expectation may return near-optimal approximately feasible (with a bounded violation error) solutions.

# 7 CONCLUSION

This paper studied the problem of maximizing the total welfare gain of EV users participating in scheduling reservation programs in *large-scale, networked* EV charging facilities. Aiming to inform and advance the design of scalable and efficient reservation management strategies, we introduced and empirically verified a Deep Combinatorial Learning pipeline that can attain near-optimal scheduling decisions within sub/near-linear time. To provide further scrutiny, we tested the constructed model's generalisability and robustness against out-of-sample and out-of-distribution inputs based on previously unseen settings and parameters of the problem. The findings signify the potential of the proposed approach to pave the way towards more efficient means of tackling combinatorial optimization problems with tens of thousands of decision variables.

# ACKNOWLEDGMENTS

We are very grateful to the anonymous referees for their time and effort spent on reviewing this work. Their valuable feedback and constructive comments helped to improve the paper's presentation quality and clarity. Also, we would like to extend our sincere appreciation to Anahit Sargsyan (a Ph.D. student at the Technical University of Munich) for reading, commenting on, and helping to polish the preliminary variant of this paper.

## REFERENCES

- Abdullah Al Zishan, Moosa Moghimi Haji, and Omid Ardakanian. 2020. Adaptive Control of Plug-in Electric Vehicle Charging with Reinforcement Learning. In Proceedings of ACM e-Energy '20. 116–120.
- [2] Bahram Alinia, Mohammad H Hajiesmaili, Zachary J Lee, Noel Crespi, and Enrique Mallada. 2020. Online EV scheduling algorithms for adaptive charging networks with global peak constraints. *IEEE Transactions on Sustainable Computing* (2020).
- [3] Belqasem Aljafari, Pandia Rajan Jeyaraj, Aravind Chellachi Kathiresan, and Sudhakar Babu Thanikanti. 2023. Electric vehicle optimum charging-discharging scheduling with dynamic pricing employing multi agent deep neural network. *Computers and Electrical Engineering* 105 (2023), 108555.
- [4] Muhammad Amjad, Ayaz Ahmad, Mubashir Husain Rehmani, and Tariq Umer. 2018. A review of EVs charging: From the perspective of energy optimization, optimization approaches, and charging techniques. *Transportation Research Part D: Transport and Environment* 62 (2018), 386–417.
- [5] Kristien Clement-Nyns, Edwin Haesen, and Johan Driesen. 2010. The Impact of Charging Plug-In Hybrid Electric Vehicles on a Residential Distribution Grid. IEEE Transactions on Power Systems 25, 1 (2010), 371–380. https://doi.org/10.1109/TPWRS.2009.2036481
- [6] S. Deilami, A. S. Masoum, P. S. Moses, and M. A. S. Masoum. 2011. Real-Time Coordination of Plug-In Electric Vehicle Charging in Smart Grids to Minimize Power Losses and Improve Voltage Profile. *IEEE Trans. Smart Grid* 2, 3 (2011), 456–467. https://doi.org/10.1109/TSG.2011.2159816
- [7] Roel Dobbe, Oscar Sondermeijer, David Fridovich-Keil, Daniel Arnold, Duncan Callaway, and Claire Tomlin. 2019. Toward distributed energy services: Decentralizing optimal power flow with machine learning. *IEEE Transactions on Smart Grid* 11, 2 (2019), 1296–1306.
- [8] Wenqian Dong, Zhen Xie, Gokcen Kestor, and Dong Li. 2020. Smart-PGSim: Using neural network to accelerate AC-OPF power grid simulation. In SC20: International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 1–15.
- [9] Khaled Elbassioni, Areg Karapetyan, and Trung Thanh Nguyen. 2019. Approximation schemes for r-weighted Minimization Knapsack problems. Annals of Operations Research 279, 1 (01 Aug 2019), 367–386.
- [10] Ferdinando Fioretto, Terrence W.K. Mak, and Pascal Van Hentenryck. 2020. Predicting AC Optimal Power Flows: Combining Deep Learning and Lagrangian Dual Methods. Proceedings of the AAAI Conference on Artificial Intelligence 34, 01 (Apr. 2020), 630–637.
- [11] O. Frendo, N. Gaertner, and H. Stuckenschmidt. 2019. Real-Time Smart Charging Based on Precomputed Schedules. *IEEE Trans. Smart Grid* 10, 6 (2019), 6921–6932. https://doi.org/10.1109/TSG.2019.2914274
- [12] L. Gan, U. Topcu, and S. H. Low. 2013. Optimal decentralized protocol for electric vehicle charging. *IEEE Trans. Power Syst.* 28, 2 (2013), 940–951. https://doi.org/10.1109/TPWRS.2012.2210288
- [13] Maxime Gasse, Didier Chetelat, Nicola Ferroni, Laurent Charlin, and Andrea Lodi. 2019. Exact Combinatorial Optimization with Graph Convolutional Neural Networks. In Advances in Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc.
- [14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep learning. MIT press.
- [15] Victor J Gutierrez-Martinez, Claudio A Cañizares, Claudio R Fuerte-Esquivel, Alejandro Pizano-Martinez, and Xueping Gu. 2010. Neural-network securityboundary constrained optimal power flow. *IEEE Transactions on Power Systems* 26, 1 (2010), 63–72.

e-Energy '23, June 20-23, 2023, Orlando, FL, USA

- [16] Kurt Hornik. 1991. Approximation capabilities of multilayer feedforward networks. Neural Networks 4, 2 (1991), 251–257.
- [17] Wanjun Huang, Xiang Pan, Minghua Chen, and Steven H Low. 2021. DeepOPF-V: Solving AC-OPF Problems Efficiently. *IEEE Transactions on Power Systems* 37, 1 (2021), 800–803.
- [18] Areg Karapetyan, Khaled Elbassioni, Majid Khonji, and Sid Chi-Kin Chau. 2023. Approximations for generalized unsplittable flow on paths with application to power systems optimization. *Annals of Operations Research* 320, 1 (01 Jan 2023), 173–204.
- [19] Majid Khonji, Sid Chi-Kin Chau, and Khaled Elbassioni. 2018. Approximation scheduling algorithms for electric vehicle charging with discrete charging options. In Proceedings of the Ninth International Conference on Future Energy Systems. 579–585.
- [20] Stavros G. Kolliopoulos and Clifford Stein. 2001. Approximation Algorithms for Single-Source Unsplittable Flow. SIAM J. Comput. 31, 3 (2001), 919–946.
- [21] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. Nature 521, 7553 (01 May 2015), 436–444.
- [22] Zachary J. Lee, George Lee, Ted Lee, Cheng Jin, Rand Lee, Zhi Low, Daniel Chang, Christine Ortega, and Steven H. Low. 2021. Adaptive Charging Networks: A Framework for Smart Electric Vehicle Charging. *IEEE Transactions on Smart Grid* 12, 5 (2021), 4339–4350. https://doi.org/10.1109/TSG.2021.3074437
- [23] Hepeng Li, Zhiqiang Wan, and Haibo He. 2020. Constrained EV Charging Scheduling Based on Safe Deep Reinforcement Learning. *IEEE Transactions on Smart Grid* 11, 3 (2020), 2427–2439. https://doi.org/10.1109/TSG.2019.2955437
- [24] Wei-Li Liu, Yue-Jiao Gong, Wei-Neng Chen, Zhiqin Liu, Hua Wang, and Jun Zhang. 2019. Coordinated charging scheduling of electric vehicles: a mixedvariable differential evolution approach. *IEEE Transactions on Intelligent Transportation Systems* 21, 12 (2019), 5094–5109.
- [25] Tania B. Lopez-Garcia, Alberto Coronado-Mendoza, and José A. Domínguez-Navarro. 2020. Artificial neural networks in microgrids: A review. Engineering Applications of Artificial Intelligence 95 (2020), 103894.
- [26] Karol Lina López, Christian Gagné, and Marc-André Gardner. 2019. Demand-Side Management Using Deep Learning for Smart Charging of Electric Vehicles. *IEEE Transactions on Smart Grid* 10, 3 (2019), 2683–2691. https://doi.org/10.1109/TSG.2018.2808247
- [27] Colin McKerracher, Aleksandra O'Donovan, Nikolas Soulopoulos, Andrew Grant, Siyi Mi, David Doherty, Ryan Fisher, Corey Cantor, Jinghong Lyu, Kwasi Ampofo, Andy Leach, Yayoi Sekine, Laura Malo Yague, William Edmonds, Komal Kareer, and Takehiro Kawahara. 2022. Electric Vehicle Outlook 2022 (BloombergNEF). https://about.bnef.com/electric-vehicle-outlook/
- [28] Zeinab Moghaddam, Iftekhar Ahmad, Daryoush Habibi, and Quoc Viet Phung. 2018. Smart Charging Strategy for Electric Vehicle Charging Stations. *IEEE Transactions on Transportation Electrification* 4, 1 (2018), 76–88. https://doi.org/10.1109/TTE.2017.2753403
- [29] Joy Chandra Mukherjee and Arobinda Gupta. 2014. A review of charge scheduling of electric vehicles in smart grid. *IEEE Systems Journal* 9, 4 (2014), 1541–1553.
- [30] Xiang Pan, Minghua Chen, Tianyu Zhao, and Steven H Low. 2022. DeepOPF: A feasibility-optimized deep neural network approach for AC optimal power flow problems. *IEEE Systems Journal* (2022).
- [31] Xiang Pan, Tianyu Zhao, Minghua Chen, and Shengyu Zhang. 2020. Deepopf: A deep neural network approach for security-constrained dc optimal power flow. *IEEE Transactions on Power Systems* 36, 3 (2020), 1725–1735.
- [32] L. Pieltain Fernández, T. Gomez San Roman, R. Cossent, C. Mateo Domingo, and P. Frías. 2011. Assessment of the Impact of Plug-in Electric Vehicles on Distribution Networks. *IEEE Transactions on Power Systems* 26, 1 (2011), 206– 213. https://doi.org/10.1109/TPWRS.2010.2049133
- [33] Kejun Qian, Chengke Zhou, Malcolm Allan, and Yue Yuan. 2010. Modeling of load demand due to EV battery charging in distribution systems. *IEEE transactions on power systems* 26, 2 (2010), 802–810.
- [34] E Riva Sanseverino, ML Di Silvestre, L Mineo, S Favuzza, NQ Nguyen, and QTT Tran. 2016. A multi-agent system reinforcement learning based optimal power flow for islanded microgrids. In 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC). IEEE, 1–6.
- [35] Mostafa Shibl, Loay Ismail, and Ahmed Massoud. 2020. Machine learning-based management of electric vehicles charging: Towards highly-dispersed fast chargers. Energies 13, 20 (2020), 5429.
- [36] Mostafa Shibl, Loay Ismail, and Ahmed Massoud. 2021. Electric vehicles charging management using machine learning considering fast charging and vehicleto-grid operation. *Energies* 14, 19 (2021), 6199.
- [37] Southern California Edison. 2021. Demand-Side Management Programs - Energy Conservation Assistance. https://www.sce.com/005\_regul\_info/eca/DOMSM11.DLP.
- [38] Bo Sun, Zhe Huang, Xiaoqi Tan, and Danny H. K. Tsang. 2018. Optimal Scheduling for Electric Vehicle Charging With Discrete Charging Levels in Distribution Grid. *IEEE Trans. Smart Grid* 9, 2 (2018), 624–634. https://doi.org/10.1109/TSG.2016.2558585

- and K Damashandaramusthy and Isa Ving Vang 2016 Into
- [39] Kang Miao Tan, Vigna K Ramachandaramurthy, and Jia Ying Yong. 2016. Integration of electric vehicles in smart grid: A review on vehicle to grid technologies and optimization techniques. *Renewable and Sustainable Energy Reviews* 53 (2016), 720–732.
- [40] Alfredo Vaccaro and Claudio A Cañizares. 2016. A knowledge-based framework for power flow and optimal power flow analyses. *IEEE Transactions on Smart Grid* 9, 1 (2016), 230–239.
- [41] Shenghe Xu, Shivendra S. Panwar, Murali Kodialam, and T.V. Lakshman. 2020. Deep Neural Network Approximated Dynamic Programming for Combinatorial Optimization. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 02 (Apr. 2020), 1684–1691.
- [42] Shaobing Yang, Shanshan Zhang, and Jingjing Ye. 2019. A Novel Online Scheduling Algorithm and Hierarchical Protocol for Large-Scale EV Charging Coordination. *IEEE Access* 7 (2019), 101376–101387. https://doi.org/10.1109/ACCESS.2019.2929626
- [43] Pooya Tahmasebi Zadeh, Maryam Joudaki, and Alireza Ansari. 2021. A Survey on Deep Learning Applications for Electric Vehicles in Micro Grids. In 2021 5th International Conference on Internet of Things and Applications (IoT). 1–6. https://doi.org/10.1109/IoT52625.2021.9469715
- [44] Tianyu Zhao, Xiang Pan, Minghua Chen, Andreas Venzke, and Steven H Low. 2020. DeepOPF+: A deep neural network approach for DC optimal power flow for ensuring feasibility. In 2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). IEEE, 1– 6.

This figure "fig2.png" is available in "png" format from:

http://arxiv.org/ps/2305.11195v2