Learning Assisted Demand Charge Mitigation for Workplace Electric Vehicle Charging

KALPESH CHAUDHARI, PATRICK CONNOR, JOSHUA COMDEN AND JENNIFER KING.
National Renewable Energy Laboratory, Golden, CO 80401 USA
Corresponding author: Kalpesh Chaudhari (e-mail: kalpesh.chaudhari@nrel.gov).

ABSTRACT Uncontrolled electric vehicle (EV) charging loads at a workplace could result in hefty demand charges. Several studies in the literature have focused on mitigating these demand charges at workplaces using aggregator-based centralized control and consensus-based distributed control. Though each model has successfully mitigated the demand charges they all show some limitations. Aggregator-based centralized solver is computation and time intensive while consensus-based distributed control is communication intensive. Other works with algorithms exploring demand charge mitigation also often rely on large amounts communications. These algorithms include water filling algorithms, multiobjective optimization, and a fuzzy logic controller. The machine learning-based (ML) approach investigated in this study seeks to eliminate both drawbacks using the model trained on optimization results obtained with existing methods. The investigation of ML-based controller in this study is not only found to be superior in performance due to speed and accuracy metrics compared to existing control approaches but has also provided the promising resilient approach to deal with communication failure and cybersecurity threats.

INDEX TERMS Demand charge, electric vehicles (EVs), machine learning, smart charging, transportation

I. INTRODUCTION

ELECTRIC Vehicles (EV) have been seeing a rise in demand worldwide. In 2010, about 17,000 EVs were on the road worldwide. In 2019, that number rose to about 7.2 million [1]. An average EV requires about 30 kilowatt hours to travel 100 miles which is the same amount of electricity an average American home uses each day [2]. With the continued growth of EVs, their charging could be a significant strain on the power grid [3]. Not only that, consumers could see a rise in their electricity prices [4]. With the increase in demand, an increase in demand charges will be seen as well. Demand charges are fees added to electricity bills of consumers, mainly commercial and industrial, based on the peak energy usage that month [5].

With continued growth of EVs, there will be a greater demand for charging at public charging stations, homes, and commercial properties such as parking garages or shopping centers [4]. More charging can result in higher demand peaks during the month. One issue that arises from increased...
demand is the demand charge. This is typically a monthly fee that large scale utility customers have to pay as part of maintaining the infrastructure needed to deliver high volumes of power to them. The exact value of the demand charge is based on the peak kilowatt usage during the month. Thus, the higher peak demand the higher amount a customer will have to pay. To prevent these issues related to increased demand, control strategies for charge mitigation are needed. These strategies are designed specifically to flatten spikes in demand and minimizes the costs associated with them.

National Renewable Energy Laboratory (NREL) researchers are investigating various smart-charging control strategies to solve this problem [6]. One strategy is to control electric vehicle charging using optimization methods. These methods can use a centralized solver or a distributed consensus-based solver. This strategy allows for a peak reduction, but relies heavily on communication and computation time [7]. A machine learning (ML) based charge controller can serve as an effective compliment to these optimization charge controllers and other demand charge mitigation charge controllers. The ML charge controller explored in this paper does not need data that is frequently unavailable such as the EV’s state of charge that consensus and other smart chargers use a parameter [8] [5]. The ML algorithm also does not need communication to operate.

A. RELATED WORKS

The focus of this work is on the load profile of individual charging electric vehicles which directly impacts the load profile of workplace charging infrastructure. Supervised machine learning prediction of charging behaviour includes other categories such as the prediction of EV battery state of charge [9], the prediction of session duration and energy consumption [10], the prediction of future energy needs to meet forecasted EV growth [11], and the prediction of charging speed [12].

Prediction of electric vehicle loads has been explored in other works. One example of this is the forecasting EV loads using a Bass model to predict an amount of EVs and then using a Monte Carlo algorithm to predict the load of each vehicle [13]. Another Monte Carlo simulation was done in [14] to predict uncontrolled EV loading. The Monte Carlo simulations while precise, lack the depth of a more learned model. This can be seen in [15] where a learned support vector machine (SVM) was compared against a statistical Monte Carlo prediction. The SVM had both a better mean absolute percent error and root mean square error.

In [16], multiple learners are compared in their ability to predict energy consumption of individual EV charging outlets using real world data from UCLA campus. Compared against the k-Nearest Neighbor, Support Vector Regression, and Random Forest, the Modified Pattern-based Sequence Forecasting outperformed in its predictive abilities. Although, it may come at a cost of computation burden.

Reinforcement-Learning (RL) is another type of predictive tool that has been used for EV load prediction. RL techniques such as Q-learning help in real-time learning which can be particularly useful for EV load prediction as seen in [17].

Another focus of this work is demand charge mitigation (DCM). Work has been done in different areas with the goal of DCM. For example in [18], a fuzzy logic controller is used to control EV loading for residential demand side management. In [19], multiobjective optimization is used to create an optimal charging pattern for a plug-in hybrid EV; although, this relies heavily on communication. [20] examines non-commercial DCM. In the work, a water filling algorithm is used for real-time EV smart charging. EV demand charges can be reduced by about 20-30% but the algorithm also relies heavily on communication. In [21] explores a charge management system to control EVs to reduce the cost faced by employers. The system varies the EV load based on the current building load. Although, the system requires input from users whose information may not always be correct. Finally in [22] the authors explore using an adaptive charging network for DCM. Although, this charging network also relies on intense communication.

B. MAIN CONTRIBUTIONS AND ORGANIZATION

The works above provide an overview of the world of using ML to predict EV charging behavior and of using algorithms for demand charge mitigation. The novelty of this work is in their overlap. This paper explores ML-based charge controllers for demand charge mitigation with the following contributions:

- A novel approach to predict the optimized charging profile of individual EVs is proposed. (Section III)
- The methodology of generating the data to train and test the algorithms is explored along with how the algorithms were trained and tested. The assessment of the algorithms is explored as well. (Section III)

The remaining of this paper is organized as follows. In Section II the background of the machine learning and optimization algorithms used in this paper are discussed. The methodology and novel approach behind predicting the optimized charging profiles is described in section III. The outcomes and results of testing the algorithms are discussed along with comparing the algorithms against other controllers and no control in Section IV. Finally, conclusions are drawn and future work is theorized in Section V.

II. BACKGROUND

This section discusses the machine learning and optimization algorithms used in this work. The metrics used for evaluation of the results are also described.

A. MACHINE LEARNING ALGORITHMS

ML algorithms take data and try to create a learning framework that can improve itself overtime with exposure to more data. Within ML, there are two different branches. There is supervised machine learning and unsupervised machine learning [23]. In unsupervised ML, there are no labels in
the training data set and the goal of the algorithm is to find similarity between data points and group them together. On the other hand in supervised ML, there are labels or values in the training data set. The goal of the supervised ML algorithm is to find a function that can accurately predict values or labels given input variables. The predicted value from a supervised ML algorithm can either be a discrete label typically referred to as classification or a continuous value typically referred to as regression [23].

This work, supervised ML regression models are used. This is because the charging profile for an EV is being predicted given the input variables of old start charge time, old end charge time, and energy consumed. These variables were chosen for this work due their real-world ease of attaining as opposed to other variables such as state of charge. The two models used in this work are random forest (RF) and Linear Regression (LR).

Before describing a RF, a decision tree (DT) algorithm must first be discussed. A DT tries to create an algorithm that makes a prediction by making different comparisons related to the input variables. It can be thought of a tree that continues through its decisions all the way down to the leaf nodes where no further decisions can be made. In regression, the final prediction is made by averaging the values at the leaf nodes. The DT is quite prone to overfitting the training data which results in not useful predictions [24]. A way to overcome this is to aggregate a good amount of randomized DTs and have each of them train and make a prediction on their own. This is essentially what a RF algorithm does [25]. The RF was chosen for this work based on its ability to tackle regression problems and for its built-in bagging.

The LR model is a straightforward basic ML algorithm from the field of statistics. The model is strictly for regression and it takes the input variables from the training set and tries to fit a linear function through the points. The function is then used to predict continuous values [26].

**B. OPTIMIZATION ALGORITHMS**

An optimization algorithm tries to maximize or minimize functions by choosing different input values within a set and computing the output of the function. One type of optimization problems is convex optimization which tries to minimize convex functions over convex sets. Since there is a problem of scalability with general convex optimization solvers, scalable solvers were used in this work [27].

The optimization algorithm used in this paper is the alternating direction method of multipliers (ADMM) algorithm. ADMM solves convex optimization problems by breaking the centralized problem into smaller distributed problems that are solved individually. This makes it easier to solve and handle the optimization problem [28], [29]. The controller solving the optimization problem with ADMM is referred to as the distributed consensus-based controller. In this work the goal of the centralized problem is to decide the charging rate of the EVs so that the peak among all time steps is reduced. In solving the centralized problem, it is assumed that all the information of the EVs are known such as the start times, end times, and energy consumed. The controller solving the optimization problem using centralized optimization is referred to as the centralized controller throughout the paper.

**III. METHODOLOGY**

In this section, the approach used for the prediction of charging profiles is defined. The problem is defined, the dataset is discussed along with the engineering of the data, and the training of the ML models described.

### A. EV CHARGING PROFILE

Let $e_t$ represent the power used to charge the vehicle at time $t$, $t_{\text{start}}$ represents the time when a vehicle plugs in to charge, and $t_{\text{end}}$ represents the end charging time, thus an individual vehicle’s charging profile for a session is defined below in (1):

$$P_{\text{session}} \triangleq (e_{t_{\text{start}}}, \ldots, e_{t_{\text{end}}})$$

The total complete charging profile of an individual charging station, $P_{\text{total}}$, can thus be defined below in (2):

$$P_{\text{total}} \triangleq \sum P_{\text{session}}$$

where the summation aligns the associated times.

The ML model predicts the individual optimized charging profile for each EV assuming the start time, end time, and energy is known. The aggregate predicted profile is obtained by summing each individual charging profile.

### B. EXPLORING AND PREPROCESSING THE DATASET

Two datasets were used in this work. Both datasets are described below.

This is one of the main areas of novelty with the work. The specifics of the data and the way it is preprocessed is what results in the ability to use our trained controller on real world parking garage data and obtain acceptable peak reductions.

The first dataset used is the EVI-Pro dataset. This dataset contained about 300,000 events. Obtaining a large amount of EV charging data is in the real world is incredibly difficult. EVs are currently not widely adopted and most datasets involving EV charging are small-scale. EVI-Pro can generate the large-scale data necessary to train the ML algorithm.

EVI-Pro has been developed through a collaboration between the National Renewable Energy Laboratory (NREL) and the California Energy Commission, with additional support from the U.S. Department of Energy’s Vehicle Technologies Office. It uses detailed data on personal vehicle travel patterns, electric vehicle attributes, and charging station characteristics in bottom-up simulations to estimate the quantity and type of charging infrastructure necessary to support regional adoption of electric vehicles [30].

The second data set is from NREL’s parking garage on its campus in Golden, CO. This dataset contains data from actual EVs charging in the garage during workplace hours. This data
Algorithm 1 Method for Training and Testing the ML

**Input:** $e_t, t_{start}, t_{end}, P_{session}$

**Output:** $a_{mapeLR}, a_{mapeRF}, t_{excLR}, t_{excRF}, P_{sessionRF}, P_{sessionLR}, a_{rmseLR}, a_{rmseRF}$

1: $e_t = e_t > 1$
2: $t_{start} = t_{start} > 360$
3: $t_{end} = t_{end} < 1200$
4: for $i = 1$ to 1000 do
5: $Rf.train(e_t, t_{start}, t_{end}, P_{session})$
6: $Lr.train(e_t, t_{start}, t_{end}, P_{session})$
7: end for
8: for $i = 1$ to 250 do
9: $t_{before} = t$
10: $P_{sessionRF}(i) = Rf.predict(e_{t_{test}}, t_{start_{test}}, t_{end_{test}})$
11: $t_{excRF}(i) = t - t_{before}$
12: $a_{mapeRF}(i) = \frac{1}{n} \sum_{i=1}^{n} (P_{sessionRF}(i) - P_{session_{actual}}(i))$
13: $a_{rmseRF}(i) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (P_{sessionRF}(i) - P_{session_{actual}}(i))^2}$
14: $P_{sessionLR}(i) = Lr.predict(e_{t_{test}}, t_{start_{test}}, t_{end_{test}})$
15: $t_{excLR}(i) = t - t_{before}$
16: $a_{mapeLR}(i) = \frac{1}{n} \sum_{i=1}^{n} (P_{sessionLR}(i) - P_{session_{actual}}(i))$
17: $a_{rmseLR}(i) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (P_{sessionLR}(i) - P_{session_{actual}}(i))^2}$
18: end for
19: return $P_{sessionRF}, P_{sessionLR}, t_{excRF}, t_{excLR}, a_{mapeRF}, a_{mapeLR}, a_{rmseLR}, a_{rmseRF}$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3172384, IEEE Access

FIGURE 1: Diagram of Process

C. MODEL SELECTION AND EXPERIMENTAL SETUP

This section describes the process that was gone through to obtain results. The first subsection describes how the data was generated that was then used to train and test the ML algorithms. The second subsection mentions the process of training and testing the algorithms along with assessment criteria. Fig. 1 will serve as a reference throughout this section.

1) Generating Data

For a ML algorithm to be able to learn and make predictions, it first needs data to train it. Since, algorithm should be able to mimic both the distributed and centralized algorithm it needed to train on the output of the optimal charge controllers. The EVI-Pro dataset was taken and run through both central and distributed charging controller codes setting the number of charging stations to be one and the number of EVs to be a hundred. The charging code takes the input data with parameters described above and calculates an optimized charging profile for each charging event in order to shave the peak of non-controlled charging. 1000 trials consisting of 100 random charging events for both distributed and centralized optimization algorithms were generated to train the algorithm. 250 trials consisting of 100 new EV charging events for both distributed and central optimization algorithms were generated to test the ML algorithms. The process just described can be seen in Fig. 1 with the arrow pointing from input data to testing dataset and the arrows pointing from input data to charge controller using the consensus-based distributed or the centralized solver to optimized charge profiles and finally to the training dataset. Histograms to help visualize the different differences between data sets can be seen in the figures below. The departure times, energy consumed, and dwell time data can be seen in Fig. 2, Fig. 3, and Fig. 4 respectively. The (a) figures represent the training data, the (b) figures represent the validation data, and the (c) figures represent the real-world vehicle data taken from the NREL parking garage.

2) Fitting and Testing the Algorithm

For this section, Algorithm 1’s pseudocode will serve as a guide throughout the process of generating and testing the
The algorithm first takes training and testing data inputs containing vehicle start time, vehicle departure time, energy needed to charge in kilo-watt hours, and finally the optimized charge profile obtained from optimization. These elements are already cleaned and processed, as described in the previous subsection, to ensure only good data is being used to train the algorithm. This cleaning of the data can be seen in lines 1, 2, and 3 of Algorithm 1. After obtaining the data sets the algorithm was trained using the scikit-learn: machine learning package. Random forest and linear regression objects were created and looped through 1000 trial sets so that they were trained. On each pass the object weights were updated training the algorithm to be more accurate. This step can be seen in lines 4-7 of Algorithm 1.

After the algorithm was trained, it was tested in a similar capacity. The models were tested on 250 iterations with each iteration testing the models performance. Line 10 and 14 in Algorithm 1 describes how the ML algorithm would make a prediction of the optimal charging profile for each event. To assess accuracy, the mean absolute percent error (MAPE) was used as seen in (3) along with the root mean square error (RMSE) seen in (4). MAPE was used due to its ease of understanding and typical use in forecasting. The lower the MAPE the better the result [31]. The final accuracy of each iteration was calculated by subtracting the respective MAPE, calculated in lines 12 and 18, from 100% as seen in lines 13 and 19 of Algorithm 1. RMSE was used due
its high weighting of larger errors. It also emphasizes lower error values and is commonly used in regression problems. Its calculation can be seen in lines 14 and 20 of Algorithm 1

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$  \hspace{1cm} (3)

$A_t$ represents the actual value and $F_t$ represents the forecasted value.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$  \hspace{1cm} (4)

$y_i$ represents the actual value and $\hat{y}_i$ represents the forecasted value.

The execution time for the algorithm to be able to make predictions was also calculated in each iteration. Before predicting the optimized profile the runtime was stored which can be seen in lines 9 and 15 of Algorithm 1. After predicting the optimized profile, the time to execute was calculated by taking the difference between current time and the previously stored time as seen in lines 13 and 17 of Algorithm 1.

Once done with all 250 trials the outputs returned are the individual optimized charging profiles of each EV from the random forest object and linear regression object, the execution times for each object for each iteration, and the MAPE and RMSE for each object in each iteration. The average execution time was calculated. Aggregate load profiles an electric vehicle charger might see were also calculated by summing the individual EV profiles for each iteration as seen in (2). Finally, the best accuracy ML aggregate profile for both distributed and central testing data was plotted alongside the aggregate graph of no control charging and the optimized charging algorithm that it trained on either distributed or centralized. This whole process was done twice once for the random forest ML algorithm and then for the linear regression ML algorithm. This whole process can be seen in Fig. 1 with the machine learning algorithm box to ML-based Charge controller to Predicted Charge Control Profiles to Error/Mismatch.

IV. RESULTS AND DISCUSSION

This section dissects the results of the two ML algorithms trying to mimic the distributed algorithm and the two ML algorithms trying to mimic the central algorithm. First the distributed results are explored then the centralized results. Finally its performance on unseen NREL garage data is discussed. Overall in this section the superior performance of the ML-based controller is emphasized.

A. CONSENSUS-BASED DISTRIBUTED RESULTS

The random forest algorithm performed quite well when testing with the unseen data. The most accurate run achieved an accuracy of about 97.6%. The least accurate run had an accuracy of about 83.3%. The most accurate run also achieved an RMSE of 3.9 and the least accurate of 10.7. The average time it took for the trained random forest algorithm to make a prediction was about 0.281 s. The aggregate random forest predicted charging profile with the best accuracy can be seen below in Fig. 7a being compared with the distributed charging profile and the no control profile.

The linear regression achieved a similar high accuracy of about 96.6%. Although, its worst accuracy was much lower than random forest’s. The lowest accuracy was about 69.6%. The most accurate run also achieved an RMSE of 4.2 and the least accurate of 14.8. The average prediction time of the linear regression algorithm was about 0.672 ms. It was much faster but its range of accuracy is cause for concern. The aggregate linear regression predicted charging profile with the best accuracy can be seen below in Fig. 7b being compared with the distributed charging profile and the no control profile. The range of peaks for each predicted distributed profile for random forest and linear regression can be seen in Fig. 5 and Fig. 6. These figures allow for a more complete view of the predicted profiles for each ML.

The random forest algorithm is shown to be superior to that of the consensus-based distributed algorithm. Its speed is much faster compared to a 3-minute run time for the optimization algorithm. It can also achieve quite comparable optimized charging profiles with similar peak reductions as seen in Fig. 7a.
B. CENTRALIZED RESULTS

When trying to mimic the centralized algorithm, both ML algorithms achieved much better ranges of accuracies than when trying to mimic the distributed. For random forest, the best accuracy obtained was about 99.6%. The least accurate run the accuracy was about 97.1%. The most accurate run also achieved an RMSE of 2.1 and the least accurate of 7.6. The average execution time for predicting the profiles of 100 events was about 0.277 s. The graph of the best predicted aggregate profile can be seen compared with the other profiles in Fig. 10a.

The linear regression best accuracy achieved was about 98.9% and the worst accuracy was about 91.2%. The most accurate run also achieved an RMSE of 4.2 and the least accurate of 8.7. It slightly under performed compared to the random forest algorithm. The average execution time was about 0.640 ms. The best aggregate linear regression profile compared to the other types of control profiles can be seen in Fig. 10b.

C. NREL GARAGE RESULTS

In order to see if the ML would be able to generalize well on unseen data, it was fed real world data from the NREL parking garage after being trained. Due to the better performance of RF in the previous tests, it was the only ML to test on the garage data. The RF algorithm successfully reduced the peak of the garage data for both the centralized and distributed solvers. The respective peak reductions were 29.9% and 42.6%. The centralized trained RF predicted profile is in Fig. 13a and the distributed trained RF predicted profile is in Fig. 13b. This successfully demonstrates that a RF ML algorithm trained on a generic data set from EVI-pro can successful optimize real-world load profiles. The range of peaks for each predicted distributed and central random forest profile can be seen in Fig. 11 and Fig. 12.

As before, these figures allow for a more complete view of the predicted profiles for each ML.

Again, the random forest algorithm is shown to be superior. Its speed is much faster when comparing it to a minute run time for the centralized algorithm. It also has a high range of accuracy and obtains a similar peak reduction to the centralized algorithm as seen in Fig. 10a.
allow for a more complete view of the predicted profiles for each ML. To maintain flexibility, the algorithm once fully implemented and online will continue to learn and update in real-time. This way it can remain accommodate any changes in operation whether it be permanent like a new charging station or seasonal like decreased traffic flow. This can be seen Fig. 1 with the area coming from NREL Garage data and going back to the input data.

V. CONCLUSION AND FUTURE WORK

This work proposed a novel ML-based solution to the increasing demand on the grid from EV charging loads. Unlike other work, the random forest ML algorithm required minimal variables while maintaining a high level of accuracy. It utilized only energy required, start-time, and departure-time. The ML trained on already optimized data obtained from NREL’s EVI-pro being ran through centralized and distributed consensus-based optimized charge controllers. After training both linear regression and random forest models, the ML-based charge controllers were given the testing data and they generated optimized charging profiles at fast speeds and high accuracies. The random forest ML-based controller outperformed the linear regression model and thus was used to demonstrate its ability to optimize on unseen NREL garage data. Without communication or time-intensive computation the random forest algorithm successful generated optimized charging profiles with reduced peak demands on the garage data for both the centralized and distributed optimization techniques, successfully demonstrating the novelty of this work.

Due to the high accuracy of the ML-based controller, in the future this work could be extended for detecting anomalies in EV charging. This application is particularly useful in cybersecurity threat mitigation. This would be useful in a scenario where a charging station is compromised. The compromised charging station could create a fake demand for a large amount of energy. The ML-based charge controller could recognize this demand as out of the ordinary and raise alarm.

Other future work involves using the ML-based controller to optimally dispatch DERs for EV charging scenario. The ML-based controller could serve as a guidance or a perfect forecast for demand on the grid. This in turn would allow
for sources like PV or storage to be managed and perfectly controlled to meet this hypothetical demand.

REFERENCES

[1] International Energy Agency, “Global EV Outlook 2020 Entering the decade of electric drive?” Tech. Rep., 2020.
[2] Alex Brown, “Electric Cars Will Challenge State Power Grids,” 1 2020.
[3] A. Ghasempour, Advanced Metering Infrastructure in Smart Grid: Requirements, Challenges, Architectures, Technologies, and Optimizations, J. Lou, Ed. Nova Science Publishing, 2017.
[4] Nichola Groom and Tina Bellon, “EV rollout will require huge investments in strained U.S. power grids,” 3 2021.
[5] A. A. Mohamed, D. Day, A. Meintz, and J. Myungsoo, “Real-Time Implementation of Smart Wireless Charging of On-Demand Shuttle Service for Demand Charge Mitigation,” IEEE Transactions on Vehicular Technology, vol. 70, no. 1, pp. 59–68, 1 2021.
[6] Andrew Meintz, “Electric Vehicle Smart Charging at Scale.”
[7] A. S. Al-Ogaili, T. J. Tengku Hashim, N. A. Rahmat, A. K. Ramasamy, M. B. Marsadek, M. Faisal, and M. A. Hannan, “Review on scheduling, clustering, and forecasting strategies for controlling electric vehicle charging: Challenges and recommendations,” IEEE Access, vol. 7, pp. 128 353–128 371, 2019.
[8] I. B. Espedal, A. Jinasena, O. S. Burheim, and J. J. Lamb, “Current trends for state-of-charge (SoC) estimation in lithium-ion battery electric vehicles,” 6 2021.
[9] S. Huang, Z. He, and X. Li, “A Method of SOC Estimation for Electric Vehicle Based on Limited Information,” in 2020 IEEE International Conference on Networking, Sensing and Control, ICNSC 2020. Institute of Electrical and Electronics Engineers Inc., 10 2020.
[10] S. Shahrriari, A. R. Al-Ali, A. H. Osman, S. Dhou, and M. Nijim, “Prediction of EV charging behavior using machine learning,” IEEE Access, vol. 9, pp. 111 576–111 586, 2021.
[11] Q. Xing, Z. Chen, Z. Zhang, X. Huang, Z. Leng, K. Sun, Y. Chen, and H. Wang, “Charging demand forecasting model for electric vehicles based on online ride-hailing trip data,” IEEE Access, vol. 7, pp. 157 390–157 409, 2019.
[12] J. J. Mies, J. R. Helmus, and R. van den Hoed, “Estimating the charging profile of individual charge sessions of Electric Vehicles in The Netherlands,” World Electric Vehicle Journal, vol. 9, no. 2, 8 2018.
[13] D. Liu, Z. Li, J. Jiang, X. Cheng, and G. Wu, “Electric vehicle load forecast based on monte carlo algorithm,” 2020, pp. 1760–1763.
[14] Q. Dai, T. Cai, S. Duan, and F. Zhao, “Stochastic modeling and forecasting of load demand for electric bus battery-swap station,” IEEE Transactions on Power Delivery, vol. 29, no. 4, pp. 1909–1917, 2014.
[15] E. S. Xydas, C. E. Marmaras, L. M. Cipicigan, A. S. Hassan, and N. Jenkins, “Forecasting Electric Vehicle charging demand using Support Vector Machines,” in Proceedings of the Universities Power Engineering Conference, 2013.
[16] M. Majidpour, C. Qiu, P. Chu, R. Gadh, and H. R. Pota, “A Novel Forecasting Algorithm for Electric Vehicle Charging Stations,” 2014 International Conference on Connected Vehicles and Expo (ICCVE), 2014.
[17] M. Dabbaghjamanesh, A. Moenini, and A. Kavousi-Fard, “Reinforcement Learning-Based Load Forecasting of Electric Vehicle Charging Station Using Q-Learning Technique,” IEEE Transactions on Industrial Informatics, vol. 17, no. 6, pp. 4229–4237, 6 2021.
[18] S. Fadel and O. A. Mohammed, “Automated distributed electric vehicle controller for residential demand side management,” IEEE Transactions on Industry Applications, vol. 55, no. 1, pp. 16–25, l 2019.
[19] M. Moenini-Aghtaie, A. Abbaspour, M. Fotuhi-Firuzabad, and P. Dehghanian, “Optimized Probabilistic PHEVs Demand Management in the Context of Energy Hubs,” IEEE Transactions on Power Delivery, vol. 30, no. 2, pp. 996–1006, 4 2015.
[20] G. Zhang, S. T. Tan, and G. Gary Wang, “Real-Time Smart Charging of Electric Vehicles for Demand Charge Reduction at Non-Residential Sites,” IEEE Transactions on Smart Grid, vol. 9, no. 5, pp. 4027–4037, 9 2018.
[21] M. Jun and A. Meintz, “Workplace Charge Management with Aggregated Building Loads,” in 2018 IEEE Transportation and Electrification Conference and Expo, ITEC 2018. Institute of Electrical and Electronics Engineers Inc., 8 2018, pp. 519–524.
[22] W. B. Heredia, K. Chaudhari, A. Meintz, M. Jun, and S. Pless, “Evaluation of smart charging for electric vehicle-to-building integration: A case study,” Applied Energy, vol. 266, 5 2020.
[23] O. Simeone, “A Very Brief Introduction to Machine Learning With Applications to Communication Systems,” 8 2018. [Online]. Available: http://arxiv.org/abs/1808.02342
[24] J. R. Quinlan, “Induction of Decision Trees,” Tech. Rep., 1986.
[25] G. Biau and G. B. Fr, “Analysis of a Random Forest Models,” Tech. Rep., 2012.
[26] David A. Freedman, Statistical Models: Theory and Practice. Cambridge University Press, 2009.
[27] H. Hindi, “A Tutorial on Convex Optimization,” Tech. Rep.
[28] D. Hallac, J. Leskovec, and S. Boyd, “Network lasso: Clustering and forecasting strategies for controlling electric vehicle chargers,” IEEE Transactions on Vehicular Technology, vol. 68, no. 11, pp. 1879–1891, 11 2019.
[29] Eric Wood, “EVI-Pro: Electric Vehicle Infrastructure – Projection Tool.”
[30] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed Optimization and Statistical Learning via the Proximal Algorithm,” vol. 1, pp. 1–15, 1 2011.
[31] “MEAN ABSOLUTE PERCENTAGE ERROR (MAPE),” in Encyclopedia of Production and Manufacturing Management. Springer US.
KALPESH CHAUDHARI (Member, IEEE) received a B.E. in electrical engineering from Sardar Patel College of Engineering, Mumbai, India, in 2013 and Ph.D. in electrical engineering from Nanyang Technological University, Singapore, in 2019.

He worked as a design engineer with Jacobs Engineering, Mumbai, India, between 2013 and 2014 specializing in substation and distribution grid designing for petrochemical refineries. Since 2018, he has been working as a researcher in electric vehicle grid integration and advanced control at the National Renewable Energy Laboratory in Golden, CO, USA, working on developing advanced EV control algorithms and optimization formulations for an autonomous energy grid.

His research interests include peer-to-peer energy trading, modeling of electric vehicle changing load, optimization of energy storage systems, and developing decentralized control algorithms for optimal operation of electric vehicle charging stations.

PATRICK M. CONNOR is from Pittsburgh, Pennsylvania, USA. He is currently in the process of receiving his B.S. in electrical engineering from the University of Pittsburgh which he is set to earn in 2022. Since the summer of 2021, he has been a researcher at the National Renewable Energy Laboratory. He is also involved in research in the power systems lab at the University of Pittsburgh.

His current research interests include optimizing power systems using machine learning, distribution system simulation, and modernization of the grid.

JOSHUA COMDEN is a Postdoctoral Researcher at the National Renewable Energy Laboratory, Golden, CO, USA. He received his Ph.D. (2019) and M.S. (2015) degrees in operations research in the Department of Applied Mathematics and Statistics at Stony Brook University, Stony Brook, NY, USA, and his B.E. (2009) degree in chemical engineering from the University of Delaware, Newark, DE, USA. He won the Woo Jong Kim Dissertation Award in 2020. His research interests include the design and analysis of online optimization and control algorithms for power systems.

JENNIFER KING is a research engineer at NREL working at the National Wind Technology Center primarily on wind farm controls and optimization, autonomous energy systems, optimal design and dispatch of hybrid energy systems. She is passionate about finding new ways to better integrate and operate renewable energy into the grid of the future. She obtained her Ph.D. in aerospace engineering and mechanics from the University of Minnesota where she developed her passion for renewable energy and controls, where her primary focus was on reduced-order modeling and control of wind farms. Her current research focus areas are in modeling and distributed control/optimization of large-scale hybrid systems.