Electric Vehicle Charging via Machine-Learning Pattern Recognition

Theron Smith, Ph.D.[1]; Joseph Garcia, Ph.D.[2]; and Gregory Washington, Ph.D.[3]

Abstract: This paper presents machine learning valley-filling (MLVF) to enhance plug-in electric vehicle (PEV) charging at the local power level while minimizing the effects of uncontrolled PEV charging. This study investigated whether a neural network algorithm could learn to identify when to begin charging a PEV by distinguishing low and high demand sections in the forecasted baseload. The results indicate that a neural network algorithm can indeed identify low and high demand sections in the forecasted baseload and learn when to begin charging electric vehicles to decrease demand loads. MLVF achieved a microaveraged $F1$ score, an indicator of a classifier's overall accuracy, of 92.84% of selecting the correct timeslot to initiate charging. **DOI: 10.1061/(ASCE)EY.1943-7897.0000778.**

Author keywords: Machine learning; Neural network; Plug-in electric vehicle (PEV); Valley filling; Distribution transformer; PEV charging.

Introduction

Limited fossil fuels and regulations to improve air quality are creating unprecedented changes in transportation, influencing customers to purchase zero-emission vehicles (ZEVs) instead of traditional internal combustion engine vehicles. The Governor of California established legislation to lower the greenhouse gas emissions due to transportation and promote ZEVs in California, setting a goal of reaching 1.5 million ZEVs in California by 2025 (GO-Biz 2013). ZEVs are classified as plug-in electric vehicles (PEVs), battery electric vehicles (BEVs), and fuel cell electric vehicles (FCEVs). However, PEVs and BEV's draw power from the grid, and as these vehicles become more prevalent, multiple studies have demonstrated that they will increase the electric load and overwhelm the grid (González Vayá and Andersson 2012; Masoum et al. 2011). The issue arises from customers generally arriving home and plugging-in their vehicles at the same time, producing a large electric load for which the current infrastructure was not designed, causing residential distribution transformers to operate above their rated limits and decreasing their lifespan (Razeghi et al. 2014).

This concern led to the development of smart charging protocols that shift PEV charging based on grid loads and the vehicle's owner's needs (Sharma et al. 2014). Generally, valley-filling, is a strategy used to smart-charge vehicles by shifting electric charging to hours in which the demand on the electric grid is lowest, filling valleys in the load curve (Zhang et al. 2014; Gan et al. 2013). In previous studies, valley-filling has been performed using many different methods which include but are not limited to linear programming (González Vayá et al. 2012; Zhang et al. 2014; Faddel et al. 2017; Ahn et al. 2011; Ren et al. 2021; El-Zonkoly 2019), dynamic programming (Clement-Nyns et al. 2010; Zhang and Li 2017), quadratic programming (Clement-Nyns et al. 2010), stochastic programming (Clement et al. 2009; Gan et al. 2012), particle swarm optimization and genetic algorithms (Qi et al. 2018; Kang et al. 2016), maximum sensitivity selection (Masoum et al. 2015; Deilami et al. 2011), Nash equilibrium (Ma et al. 2013), and fuzzy logic (Geng et al. 2013; Masoum et al. 2015; Singh et al. 2015). All these methods rely on complex optimization, iterative programming, or lengthy calculations for implementation. In contrast, machine learning, a technique used to develop mathematical models of data by identifying patterns through generalization and inference, has been shown to perform similarly to the listed methods while requiring less computational power once trained. However, machine learning has not been explored extensively in valley-fill algorithms.

In reference to PEV charging, machine learning has been used for load forecasting. Panahi et al. (2015) showed that an artificial neural network is capable of predicting the arrival time of PEVs with high accuracy, the distance a vehicle travels, and transformer load profiles. Li et al. (2018) compared five machine learning algorithms and showed that a convolutional neural network (CNN) integrated with the niche immune lion logarithm (NILA) is a promising technique for short-term load forecasting of EV charging stations. Zhu et al. (2019) conducted a comprehensive comparative study using six algorithms applied to three scenarios and determined that a long-short-term memory (LSTM) model is superior to the other methods and is competent to forecast short term PEV charging loads. Muzaffar and Afshari (2019) completed a similar comparison study and showed that LSTM can model and forecast transformer loads well, aligning with the results from Zhu et al. (2019). Muzaffar and Afshari (2019) and Zhu et al. (2019) did not include a CNN in their comparison.

Beyond load forecasting, machine learning has been used as a reinforcement tool for learning an optimum cost-reducing charging policy from a batch of transition samples and making cost-reducing charging decisions in new situations (Chiş et al. 2017).

[1]Engineer, Henry Samueli School of Engineering, Univ. of California, Irvine, 5200 Engineering Hall, Irvine, CA 92697-2700 (corresponding author). Email: smithtf@uci.edu

[2]Engineer, Henry Samueli School of Engineering, Univ. of California, Irvine, 5200 Engineering Hall, Irvine, CA 92697-2700. Email: josephg2@uci.edu

[3]President, George Mason Univ., Merten Hall Suite 5100, 4400 University Dr, MSN3A1, Fairfax, VA 22030. ORCID: https://orcid.org/0000-0002-3011-3159. Email: gwashin@gmu.edu

Note. This manuscript was submitted on October 6, 2020; approved on April 1, 2021; published online on August 2, 2021. Discussion period open until January 2, 2022; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Energy Engineering***, © ASCE, ISSN 0733-9402.

Majidpour et al. (2014) compared four algorithms in an analysis to discover the best algorithm for predicting the expected charging finishing time at a charging station for use as a cellphone application. The analysis showed that the nearest neighbor algorithm (k-nearest neighbor with $k = 1$) performed best. Yi and Scoffield (2018) showed that it is possible to estimate residential EV charging load profiles using historical residential charging behavior data and kernel density estimation.

The aforementioned numerous optimization techniques have shown great valley-filling capabilities using forecasted baseloads. In addition, the previously mentioned studies demonstrated that machine learning can enhance load forecasting and predict the completion time of vehicle charging, which improves how well optimization techniques can valley-fill. However, this requires two separate algorithms to work together: the machine learning algorithm creates the enhanced forecasted load, which is given to the optimization algorithm to schedule PEV charging. A machine learning algorithm can condense this process by using inputs that describe the baseload and vehicle to load forecast and schedule PEV charging simultaneously, potentially decreasing computational cost and increasing effectiveness. Before this is considered, the capacity of utilizing machine learning to valley-fill needs to be studied because it has not been assessed or defined.

This study explored how machine learning can improve PEV charging by investigating whether a machine learning algorithm can identify valleys in a forecasted load and charge vehicles as a preliminary effort to use machine learning to load forecast and scheduling PEV in the same instance. This study applied a neural network algorithm and examined its capabilities for learning how to identify when to begin charging a PEV by distinguishing between low and high demand sections in the forecasted baseload. Furthermore, the goal was to create an algorithm that is computationally fast, operates in real-time, is scalable, satisfies customer needs, reduces transformer loss of life, and contributes to smart transportation. These goals provided motivation to investigate machine learning for this application because of its robust capabilities to reproduce and model nonlinear processes. A classification algorithm, machine learning valley-filling (MLVF), is proposed, and it was trained to identify the best interval to begin charging a vehicle given the vehicle's charge needs, dwell time, and the forecasted baseload. It was assumed that a PEV can accept any charging rate between 0 and 11.5 kW, and that the maximum charging capacity of a charging station is 11.5 kW. Additionally, it was assumed that charging occurs using a constant selected rate and is delivered continuously until the vehicle reaches the desired charge level or is unplugged.

Problem Formation

This paper presents a machine learning algorithm, machine learning valley-filling, to determine the best time to begin charging vehicles based on their need and the forecasted baseload values. In practice, users would connect their vehicle to a smart charger and provide their vehicle's dwelling time and requested charge. The algorithm would be embedded in a smart charger and would utilize the user's inputs and the forecasted baseload to determine when to charge the vehicle. The smart charger would be connected to a central distribution transformer and would receive real-time updates of the forecasted baseload.

To simplify design, arranging MLVF to accept the same number of inputs for all vehicles increases its robustness and limit complexity. The number of inputs associated with the user, the requested charge, and dwell time do not change from vehicle to vehicle; however, the length of the forecasted baseload does. To account for this, the baseload is fixed in length and described to MVLF in discrete sections, regardless of the dwell time of the vehicle. This study performed an analysis to determine the best fixed length of time and duration of a section that should be used to describe the baseload as an input to MVLF. Two values were considered for the potential fixed length of time in which the baseload is described, 8 and 12 h. Three values were considered for the potential duration into which the baseload is sectioned, 15, 30, and 60 min. These sections are represented as values that are found by sectioning the baseload into time-intervals and calculating the average of each interval.

Table 1. Cases assessed in MLVF Study

Case	Forecasted baseload length (h)	Time interval duration (min)
1	8	60
2	8	30
3	8	15
4	12	60
5	12	30
6	12	15

Six cases were investigated to determine the best length of forecasted baseload and duration of the time intervals that should be used to describe the baseload as an input to MLVF (Table 1). These cases assessed MLVF's ability to select the correct charge initiation time and reduce the average load during charging. In each case, the duration was represented by the variable m.

The vehicle data described in the section "Transformer and PEV Data" contained 20,295 data examples and were overlaid onto three transformer baseloads, producing a data set of 60,885 examples. In each example, vehicle i's arrival time a_i, dwell time d_i, requested charge r_i, and forecasted baseload L were known. The raw data set was organized in a matrix in which the rows were data entries and the columns were features of each data example. The raw data set was processed and used to train and test MLVF. The first step of the processing procedure was to find the priority ratio, p_i, for each vehicle in the data set [Eq. (1)]. The priority ratio is a relation between needed charge and dwelling time; needing a significant amount of charge and having a small dwelling time creates high priority

$$p_i = \frac{r_i}{d_i}, \qquad i = 1, 2, \ldots, 60{,}885 \tag{1}$$

The priority ratio was compared to five hourly rates to determine the charging rate for each vehicle, R_i (Table 2).

The requested charge was divided by the charging rate, R_i, which was selected for vehicle i using Table 2 to calculate the vehicle's charging time, C_i

$$C_i = \frac{r_i}{R_i} \tag{2}$$

Each vehicle's load, $\mathbf{l}_i$ was determined by multiplying C_i and R_i. This creates a rectangle that is R_i kilowatts high and C_i hours long

Table 2. Selected charging rate table (kW)

Priority ratio, p_i	Charging rate, R_i
<1.9	1.9
<3.3	3.3
≥3.3	7.2
≥7.2	9.6
≥11.5	11.5

$$\mathbf{l}_i = C_i \times R_i \tag{3}$$

All possible charging profiles, j, for the ith vehicle are represented by $L(t)_i^j$. Eq. (4) is used to find the maximum number of charging profiles, q_i, for each vehicle i. The variable q_i, is the number of times $\mathbf{l}_i$ can be shifted within the total dwelling period for vehicle i; $\mathbf{L}(t)_i^j$ is formed by analyzing vehicle i's plug-in time and shifting the charge initiation time by j time intervals

$$q_i = \frac{\text{dwell time}_i}{m \text{ minutes}} - \frac{\text{requested charge}_i}{R_i \times m \text{ minutes}} \tag{4}$$

All the possible charging profiles in $\mathbf{L}(t)_i^j$ were added to the baseload, $\mathbf{B}(t)$. For each vehicle i, the j that minimizes F in Eq. (5) was stored into the ith entry of the column vector $\mathbf{Y}$. This value was used as the truth value to train the machine learning model. This optimization process simulated vehicle i's load, $\mathbf{l}_i$, shifting along the total dwelling period for vehicle i and finding the start time that produced the lowest average load value

$$\min F = \frac{1}{N}\sum_{t=0}^{N} \mathbf{B}(t) + \mathbf{L}(t)_i^j, \quad j = 0, \ldots, q_i \tag{5}$$

The portion of the baseload corresponding to timespan of the profile within $\mathbf{L}(t)_i^j$ that satisfied Eq. (5), the vehicle's dwell time, and requested charge were organized into a vector, $\mathbf{X}_i$ [Eq. (6)]. The baseload was expressed by k number of elements, where k is the total length of provided forecasted baseload divided by the length of the time intervals used in the analysis. Eq. (7) sets the j value that minimized Eq. (5) to $\mathbf{Y}_i$

$$\mathbf{X}_i = \begin{bmatrix} \text{Requested sharge} \\ \text{Dwell time} \\ \text{Averaged forecasted baseload}_1 \\ \text{Averaged forecasted baseload}_2 \\ \vdots \\ \text{Averaged forecasted baseload}_k \end{bmatrix}^T \tag{6}$$

$$\mathbf{Y}_i = [\text{Interval to begin charging}] \tag{7}$$

The 60,885 examples were evaluated, and the final structure of the input and output data sets are represented by Eqs. (8) and (9)

$$\overline{\mathbf{X}} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_{60,885} \end{bmatrix} \tag{8}$$

$$\overline{\mathbf{Y}} = \begin{bmatrix} \boldsymbol{Y}_1 \\ \boldsymbol{Y}_2 \\ \vdots \\ \boldsymbol{Y}_{60,885} \end{bmatrix} \tag{9}$$

The Z-score [Eq. (10)] was used to standardize each column within $\overline{\mathbf{X}}$; Z-score standardization removes the mean, μ, and scales to unit variance, σ, making data normally distributed

$$Z\text{–score}(\overline{\mathbf{X}}) = \frac{(\overline{\boldsymbol{X}} - \mu)}{\sigma} \tag{10}$$

To create the MLVF, data were administered to the algorithm in three sets: training, validation, and testing. The 60,885 examples were divided into sets by designating 80% of the data for design and 20% for testing. Then the design data was divided again, allocating 80% for training and 20% for validation. This resulted in 64% for training, 16% for validating, and 20% for testing, translating to 38,961 training, 9,747 validation, and 12,177 testing examples.

Training data are the initial set of data that are given to the algorithm, allowing the model to develop correlations. Validation data are used to assess how well the training data trained the algorithm. After each iteration of training, validation data are inserted into the trained model, and the output is compared with the true output associated with the validation point. The model continued to fit to the training data until the validation accuracy did not increase for 25 consecutive iterations. After training was complete, MVLF's network weights were equated to the weights from iteration with the highest validation accuracy, and the algorithm was fully constructed (Géron 2018).

A brief description of neural networks is provided here for clarity; a more detailed explanation was given by Géron (2018). MLVF is composed of six layers: an input, three hidden layers, a softmax layer, and an output layer (Géron 2018). The process begins by importing inputs, x, into the nodes of the input layer of the neural network. The inputs then are multiplied by a vector of weights, $\boldsymbol{\theta}$, and are sent to the nodes in the hidden layer

$$\mathbf{z} = \boldsymbol{\theta}^{\mathbf{T}}\mathbf{x} \tag{11}$$

Each node hypothesizes what its output should be using the information it received from the previous node using an activation function, $g(z)$

$$\mathbf{H} = g(z) = \mathbf{g}(\boldsymbol{\theta}^{\mathbf{T}}\boldsymbol{x}) \tag{12}$$

There are three hidden layers, each of which uses the rectified linear unit (ReLU) activation function [Eq. (13)]. The output produced by the hypothesis of nodes in the first layer becomes the input to the nodes in the second layer. The process continues until the final hidden layer is evaluated, and the hypothesis of each node in the final hidden layer is sent to the nodes in Layer 5, the softmax layer

$$g(\boldsymbol{\theta}^{\boldsymbol{T}}\boldsymbol{x}) = \max(0, \boldsymbol{\theta}^{\boldsymbol{T}}\boldsymbol{x}) \tag{13}$$

The softmax layer uses the softmax activation function [Eq. (14)]. This function outputs a vector that represents the probability distributions of each potential starting interval being correct

$$\mathbf{S}(\mathbf{y}_i) = \frac{\boldsymbol{y_i}}{\sum \boldsymbol{e}^{\boldsymbol{y_j}}} \tag{14}$$

The interval with the highest probability then is selected as the charge initiation interval

$$y = \max(\mathbf{S}(\mathbf{y_i})) \tag{15}$$

MLVF learns patterns within data through a process called backpropagation. The algorithm runs over many iterations trying to optimize a cost function, $J(\theta)$, which is expressed by the cross-entropy function

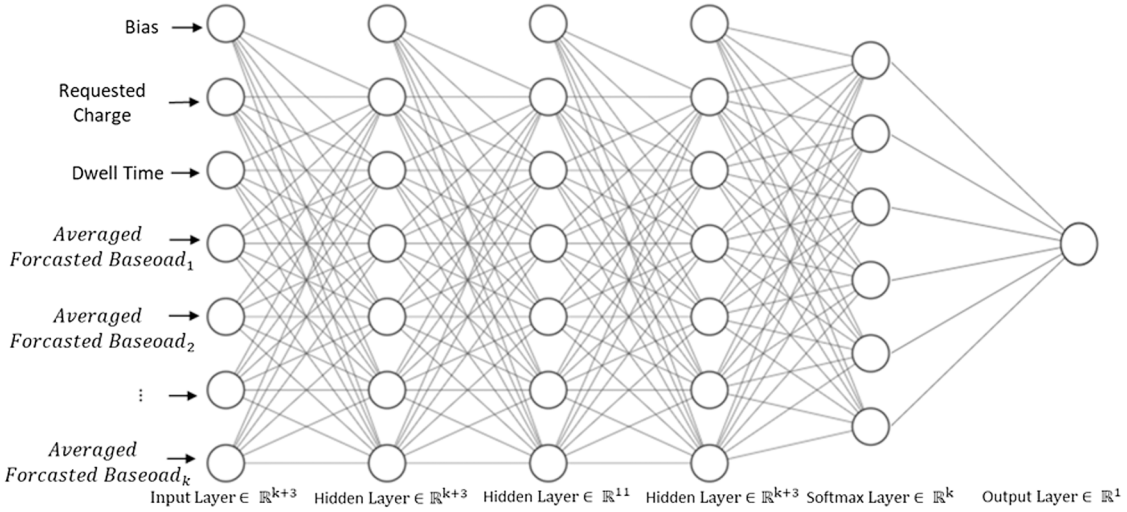

Fig. 1. MLVF's neural network architecture.

$$J(\theta) = \frac{-1}{m}\sum_{i=1}^{m}[\boldsymbol{y}^{\boldsymbol{i}}\log(\boldsymbol{H}^{\boldsymbol{i}}) + (1-\boldsymbol{y}^{\boldsymbol{i}})\log(1-\boldsymbol{H}^{\boldsymbol{i}}))] \tag{16}$$

The desired value or truth is Y, and during each iteration, the steepest descent is used to modify $\boldsymbol{\theta}$ [Eq. (17)], making H closer to the desired value. A more detailed explanation of neural networks was given by Géron (2018)

$$\boldsymbol{\theta}_j := \boldsymbol{\theta}_j - \propto \frac{\partial}{\partial \boldsymbol{\theta}_j}\mathbf{J}(\boldsymbol{\theta}) \tag{17}$$

A final illustration of MLVF is shown in Fig. 1. The network varies per case because there are $k+3$ inputs, where k is the total length of provided forecasted baseload divided by the time intervals.

Transformer and PEV Data

Transformer Data

The data used to obtain the baseloads for the simulations were measured from a 75-kW residential transformer located in Irvine, California, on three different days. The high and low temperatures of each day are listed in Table 3. The baseload was the load on the transformer before PEV charging was added. The measured 75 kW transformer served 20 homes, 8 of which had air conditioning. The size of these homes ranged from 1,900 to 2,900 ft^2. The baseload transformer data used throughout this study did not include any electric vehicle charging. The transformer data had a sampling time of 5 min. This sampling time could exaggerate changes in the load.

The baseload data used in this study were recorded from midnight to midnight. To analyze overnight charging, each day's load profile was extended from 24 to 48 hours, and the middle region (12–36 hours) was examined, thereby creating an overnight interval spanning noon to noon. The transformer data used in this study were the same as those used by Ramos Muñoz et al. (2016), allowing for direct comparison. Whereas Ramos Muñoz et al. (2016) used only Thursday, September 25, 2014, this study followed Smith (2019) and additionally used Monday, August 25, 2014, and Tuesday, September 16, 2014. In this study, August 25, 2014; September 16, 2014; and September 25, 2014 are referred to as Days 1, 2, and 3 respectively. The baseload from each day is plotted in Fig. 2, and the standard deviation of the baseloads on Days 1, 2, and 3 were 11.15, 37.82, and 16.75, respectively.

Table 3. High and low temperatures of each day

Day	Low temperature	High temperature
August 25, 2014	21.1°C (70.0°F)	27.2°C (81.0°F)
September 16, 2014	25°C (77.0°F)	37.2°C (99.0°F)
September 25, 2014	22.2°C (72.0°F)	31.1°C (88.0°F)

PEV Data

Similar to the studies outlined by Zhang et al. (2014), Ramos Muñoz et al. (2016), and Smith (2019), data from the 2009 National Household Travel Survey (NHTS) were used to simulate vehicle travel behavior. The processing steps utilized by Ramos Muñoz et al. (2016) were used. The following alterations to the data set resulted in travel data for 20,295 vehicles:

- trips without a personally owned vehicle were removed;
- person-chain data were converted to vehicle-chain data;

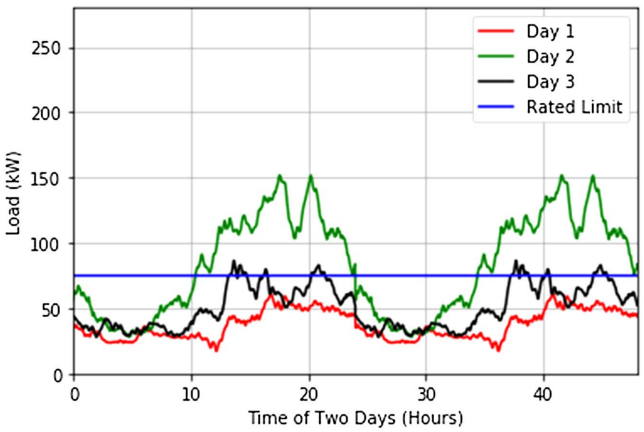

Fig. 2. Transformer baseload data for August 25, 2014, September 15, 2014, and September 25, 2014.

- daily trip data with unlinked destinations or significant overspeed were removed; and
- tours were organized to start and end at home.

Using assumptions from Zhang et al. (2014), 20,295 PEVs were assigned randomly to the 2,255 transformers, maintaining a ratio of 9 PEVs/transformer. The original random assignment was maintained throughout all simulations.

The PEV data were overlaid onto the transformer baseloads to simulate uncontrolled charging in this study. In the uncontrolled scenario, all vehicles immediately began charging at a constant rate of 7.2 kW when plugged in. Uncontrolled charging was applied to each day, and served as the datum in this study. It was referenced consistently and used to measure the performance of controlled charging.

Results and Discussion

Uncontrolled Charging

Fig. 3 shows the results for uncontrolled charging applied to each baseload, and all PEVs using 7.2 kW charging. Fig. 3 shows the load on all 2,255 randomly assigned transformers and the baseload (i.e., the load on the transformer without any PEV charging). As each vehicle charges, the power use is added to the baseload to obtain the new demand for the transformer.

When uncontrolled charging was applied to the baseloads, the average load during charging for each day was 57.87, 127.52, and 74.36 kW, respectively (Table 4). The absolute maximum peak power reached among all transformers and the average maximum peak power reached by each transformer for each day increased considerably compared with the maximum peak from each day's baseload because of the added demand from the PEVs. The maximum peak power from each day's baseload was 60.85, 151.93, and 86.58 kW, respectively. The addition of PEV charging increased the absolute maximum peak power reached among all transformers to 103.30, 195.13, and 120.58 kW, respectively. The average maximum peak power reached by each transformer was 78.36, 168.07, and 97.79 kW, respectively. This signifies that for all baseloads, the source of the absolute maximum peak power and average maximum peak power was PEV charging.

MLVF Parameter Study

The purpose of the machine learning algorithm outlined in this study is to identify when to begin charging a PEV by distinguishing between low and high demand sections in the forecasted baseload. The purpose of this parameter study was to explore six cases, shown in Table 5, and determine which case produced the best prediction capabilities for MLVF. Two parameters within the algorithm were evaluated: the length of the forecasted baseload provided to the algorithm, and the duration of the intervals into which the forecasted baseload was averaged. The analysis examined the results from providing 8- and 12-h forecasted baseloads in 15-, 30-, and 60-min averaged time intervals. Each case used the same split percentages for training, validating, and testing, i.e., 64%, 16%, and 20%, respectively. These percentages translated to 38,961 training, 9,747 validation, and 12,177 testing examples.

The multiclass metrics macroaveraged recall, weighted-averaged recall, macroaveraged precision, weighted-averaged precision, macroaveraged $F1$ score, weighted-averaged $F1$ score, and micro-averaged $F1$ score (Shmueli 2020) were used to evaluate MLVF's classification ability. The results from testing the MLVF's classification ability from each case are listed in Table 6. The macroaveraged recall was greater than the macroaveraged precision for every case. This indicates that the algorithm produced more false positives than false negatives. Generally, false positives are more desired than false negatives in classification algorithms. However, in this application, one is not inherently better than the other, because both misclassifications initiate charging in an interval that is not preferred.

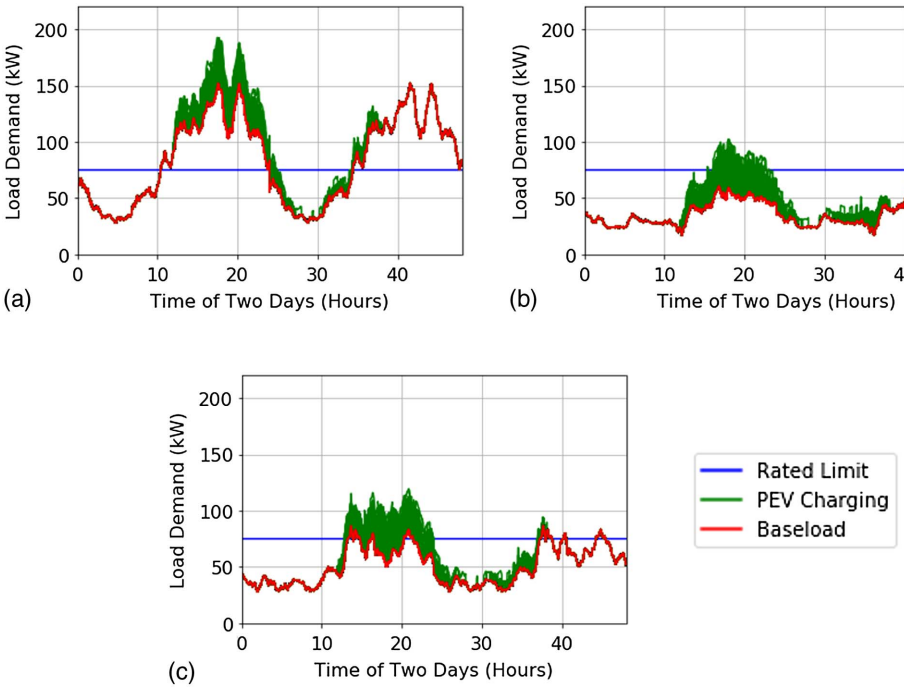

Fig. 3. Uncontrolled charging using 7.2 kW on (a) August 25, 2014 baseloads; (b) September 15, 2014 baseloads; and (c) September 25, 2014 baseload.

Table 4. Demand changes in each case

Case	Day	Absolute highest peak	Average highest peak	Average load during charging	Absolute highest peak percentage difference (%)	Average highest peak percentage difference (%)	Average load during charging percentage difference (%)
Uncontrolled	1	103.3	78.36	57.87	—	—	—
	2	195.23	168.07	127.52	—	—	—
	3	120.58	97.79	74.36	—	—	—
1	1	82.73	68.64	49.30	22.11	13.22	15.99
	2	175.53	161.81	109.49	10.63	3.80	15.21
	3	106.73	92.99	65.07	12.19	5.03	13.33
2	1	82.73	67.68	49.23	22.11	14.63	16.13
	2	173.91	161.67	106.28	11.55	3.88	18.17
	3	103.43	93.05	63.94	15.31	4.97	15.07
3	1	81.91	66.55	47.46	23.10	16.30	19.77
	2	172.01	160.23	101.53	12.65	4.78	22.69
	3	103.43	92.33	62.57	15.31	5.74	17.22
4	1	68.63	62.30	43.76	40.33	22.84	27.77
	2	163.11	154.68	83.21	17.93	8.30	42.05
	3	94.18	87.59	55.05	24.59	11.00	29.84
5	1	66.73	61.35	42.46	43.02	24.35	30.72
	2	161.21	154.11	77.95	19.09	8.67	48.25
	3	92.63	87.17	52.81	26.22	11.48	33.89
6	1	62.93	60.98	40.91	48.57	24.95	34.34
	2	157.41	152.89	70.55	21.45	9.46	57.53
	3	90.73	86.83	51.05	28.25	11.87	37.17

Table 5. Case inputs and outputs

Case	Forecast length input (h)	Interval input (min)	No. of inputs	No. of possible output classifications	No. of outputs
1	8	60	10	8	1
2	8	30	18	16	1
3	8	15	26	24	1
4	12	60	14	12	1
5	12	30	26	24	1
6	12	15	50	48	1

When averaged recall and precision were weighted, the results for all cases increased, improving up to 9% in Case 6. This indicates that the algorithm was penalized heavily for poorly predicting time intervals that have a relatively small number of vehicles in their class. The macroaveraged metrics were determined by calculating the mean value, which assigned equal weights to each class and allowed each class to influence the overall metric equally. In contrast, the weighted metrics were determined by weighting each class by the number of samples from that class, allowing classes with fewer numbers to have less influence.

Similar to macroaveraged precision and macroaveraged recall, the $F1$ score increased when changed from macroaveraged to weighted, providing more evidence that the algorithm was penalized for mislabeling vehicles belonging to a small class. The microaveraged $F1$ score represents the classifier's overall accuracy, which was a similar value to the weighted-average $F1$ score for all cases. This suggests that the sample sizes within the algorithm may be too different to use macro metrics and that the weighted-average $F1$ score may provide the most information for the algorithm's effectiveness, classifying the time at which charging should initiate at least 80% of the time for all cases. Compared with random selection, the odds of selecting the correct class within Case 1, the smallest set of classes, was 12.5%, and that within Case 6, the largest set of classes, was 2.08%. However, MLVF achieved 92.96% and 80.71% for the same cases, respectively. This proves that the algorithm learned to identify patterns and determined how to charge vehicles efficiently.

MLVF Algorithm Performance

The MLVF relies on a neural network to distribute rates, ranging between 0 and 11.5 kW. The maximum peak power from the baseload, the absolute maximum peak power reached among all transformers, and the average maximum peak power reached by each transformer when each case was applied to the baseloads of each day were 60.85, 151.93, and 86.58 kW, respectively (Table 5). Unlike the uncontrolled charging case, the maximum peak for all three baseloads was produced by the baseload rather than by PEV charging. In addition, this shows that all the cases

Table 6. Multiclass performance metrics

Case	Macroaveraged recall	Weighted-average recall	Macroaveraged precision	Weighted average precision	Macroaveraged $F1$ score	Weighted-average $F1$ score	Microaveraged $F1$ score
1	0.9258	0.9300	0.9233	0.9308	0.9244	0.9302	0.9296
2	0.8962	0.9116	0.8897	0.9124	0.8921	0.9119	0.9116
3	0.8556	0.8960	0.8539	0.8975	0.8539	0.8964	0.8960
4	0.8882	0.9001	0.8843	0.9010	0.8859	0.9002	0.9001
5	0.8528	0.8796	0.8314	0.8833	0.8381	0.8803	0.8796
6	0.7316	0.8071	0.7174	0.8207	0.7158	0.8088	0.8071

Table 7. The performance of each case

Case	Average load during charging percentage difference (%)
1	14.84
2	16.46
3	19.89
4	33.22
5	37.62
6	43.01

performed equally, relative to the reduction in the absolute maximum peak power reached among all transformers and the reduction of the average maximum peak power reached by each transformer.

The overall effectiveness of each case was determined by averaging the reduction of the average load during uncontrolled charging for all three tested days (Table 7). Cases 4–6 all performed better than Cases 1–3, indicating that the algorithm valley-fills more effectively when more hours of the forecasted loads are used in the analysis. This occurs because the algorithm has more options to select for when the vehicle will charge, increasing the chances of lowering the average load during charging.

In addition, the results showed that providing the length of forecasted baseload in smaller length intervals increases the algorithm's ability to valley-fill. Smaller intervals provide more clarity to baseload's behavior and allow the algorithm to have better insight into where valleys are located. Larger intervals overgeneralize the baseloads shape, making it more difficult to accurately determine when a vehicle should charge.

Fig. 4 shows the charging profiles for the Day 2 baseload from each case. Figs. 4(a–c) show Cases 1–3, which used 8 h of forecasted baseload. Figs. 4(d–f) show Cases 4–6, which received 12 h of forecasted baseload. Figs. 4(a and d), 4(b and e), and 4(c and f) used 60-, 30-, and 15-min time intervals, respectively. Cases 4–6 executed much more charging at low-demand times than did Cases 1–3, demonstrating that receiving 12 h of forecasted baseload is better than 8 h. In addition, averaging the baseload in smaller time intervals increased the valley-filling capabilities.

The algorithm produced more-accurate results when the length of the forecasted baseload provided to the algorithm was short, and the duration of time intervals was long. When the parameters are structured in this form, the number of possible classes the algorithm must distinguish between decreases, increasing accuracy. However, MVLF valley-filled better when the length of the forecasted baseload provided to the algorithm was long and the duration of time intervals was short (Fig. 4). This discrepancy occurs because increasing the duration of time intervals skews the interpretation of accuracy, i.e., an incorrect classification using shorter time intervals may be a correct classification using longer time intervals. This occurs when an incorrect and a correct classification of a smaller time interval are both within a larger time interval. For example, assume that a vehicle, and its needed charge, the dwell time, and the forecasted baseload are known and given to two algorithms. The first algorithm was trained on Case 1's data set using 60-min intervals, and should begin charging the vehicle in the 5:00–6:00 p.m. interval. The second algorithm was trained on Case 6's data set using 15-min intervals, and should begin charging in the 5:30–5:45 p.m. interval.

If the first algorithm begins charging at 5:00 p.m., and the second algorithm begins charging the vehicle at 5:15 p.m., the first algorithm has the correct classification whereas the second algorithm's classification is incorrect. However, when the vehicle was added to the baseload, the Case 1 algorithm had a higher average load during charging than did the Case 6 algorithm. This occurred because the Case 1 algorithm has a lower-resolution view of the baseload than the Case 6 algorithm, creating an overgeneralization problem. The Case 1 algorithm classifies more accurately, but it classifies on a baseload that is more generalized, which may be less effective. The higher-resolution view of the baseload that the Case 6 algorithm uses gives the algorithm the ability, in some instances, to be wrong by up to three intervals while still reducing the average load during charging more than the correct classification does of the Case 1 algorithm. This proves that giving the algorithm more information about the baseload can be more vital for increasing effectiveness than for increasing the algorithm prediction accuracy.

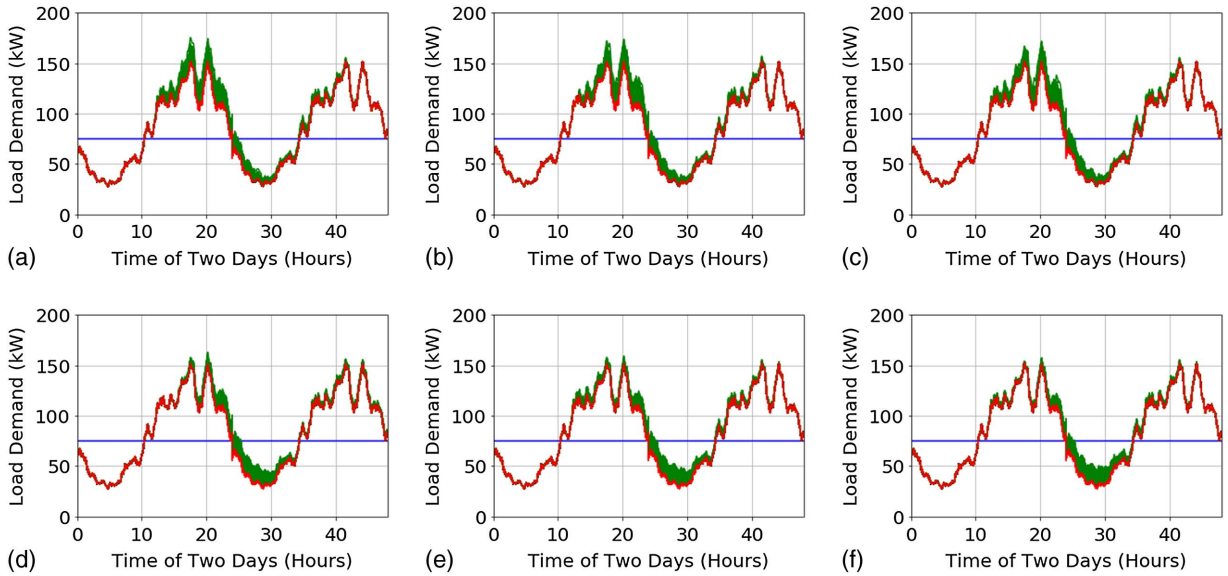

Fig. 4. Charging profiles on Day 2's baseload: (a) Case 1; (b) Case 2; (c) Case 3; (d) Case 4; (e) Case 5; and (f) Case 6.

Conclusion

The results indicate that a machine learning algorithm can learn to identify when to begin charging a PEV by distinguishing between low and high demand sections in the forecasted baseload. The results show that the algorithm can achieve a microaveraged $F1$ score, an indicator of a classifier's overall accuracy, of 0.9284; hence, the algorithm can select the correct time to initiate charging 92.84% of the time.

The analysis revealed that the algorithm valley-fills more effectively when more hours of the forecasted loads are provided as an input. In addition, its valley-filling capability increases when the length of forecasted baseload is given in smaller length intervals. It also was discovered that accuracy does not necessarily equate to valley-filling performance. Providing a lower-resolution view of the baseload increases classification accuracy, but results in a heavy generalization of the baseload that decreases the effectiveness.

More research should be conducted to investigate how other variations in the length of the forecasted baseload provided to the algorithm and its duration of time intervals affect the algorithm's ability to select the correct charge initiation time. Only two variations in the length of the forecasted baseload provided to the algorithm and three variations of the duration of the time interval were considered; more should be evaluated. Furthermore, to decrease complexity, it was assumed that charging occurs using a constant rate and is delivered continuously until the vehicle reaches the desired charge level or unplugs. Further research should consider removing these constraints and expanding the problem space. Lastly, research should investigate how introducing input variables such as weather, holidays, and observed events can teach the algorithm how these entities affect the forecasted load. These inputs allow the algorithm to learn how historical baseloads can be modified to create more accurate forecasted baseload and improve valley-filling performance.

Data Availability Statement

Code generated during this study to simulate and assess the proposed algorithm is proprietary and confidential in nature and may be provided only with restrictions.

Notation

The following symbols are used in this paper:

a_i = arrival time;
$B(t)$ = localized baseload;
C_i = charging time;
d_i = dwell time;
$g(z)$ = activation function;
H = hypothesis;
i = vehicle number;
$J(\theta)$ = cost function;
j = all possible charging profiles;
L = forecasted baseload;
l_i = each vehicle's load;
m = time interval duration;
p_i = priority ratio;
q_i = maximum number of charging profiles;
R_i = charging rate;
r_i = requested charge;
$S(y_i)$ = softmax layer;
$\overline{\mathbf{X}}$ = input data;
$\mathbf{X}_i$ = input vector;
$\overline{\mathbf{Y}}$ = output data;
$\mathbf{Y}_i$ = output vector;
z = node input;
θ = weights;
μ = mean;
σ = standard deviation; and
$\propto$ = learning rate.

References

Ahn, C., C.-T. Li, and H. Peng. 2011. "Decentralized charging algorithm for electrified vehicles connected to smart grid." In *Proc., 2011 American Control Conf.*, 3924–3929. New Orleans: American Control Conference. https://doi.org/10.1109/ACC.2011.5990895.

Chiş, A., J. Lundén, and V. Koivunen. 2017. "Reinforcement learning-based plug-in electric vehicle charging with forecasted price." *IEEE Trans. Veh. Technol.* 66 (5): 3674–3684. https://doi.org/10.1109/TVT.2016.2603536.

Clement, K., E. Haesen, and J. Driesen. 2009. "Coordinated charging of multiple plug-in hybrid electric vehicles in residential distribution grids." In *Proc., 2009 IEEE/PES Power Systems Conf. and Exposition*, 1–7. https://doi.org/10.1109/PSCE.2009.4839973. New York: IEEE.

Clement-Nyns, K., E. Haesen, and J. Driesen. 2010. "The impact of charging plug-in hybrid electric vehicles on a residential distribution grid." *IEEE Trans. Power Syst.* 25 (1): 371–380. https://doi.org/10.1109/TPWRS.2009.2036481.

Deilami, S., 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): 456–467. https://doi.org/10.1109/TSG.2011.2159816.

El-Zonkoly, A. M. 2019. "Optimal energy management in smart grids including different types of aggregated flexible loads." *J. Energy Eng.* 145 (5): 04019015. https://doi.org/10.1061/(ASCE)EY.1943-7897.0000613.

Faddel, S., A. A. S. Mohamed, and O. Mohammed. 2017. "Linear autonomous control of electric vehicles charging in distribution systems." In *Proc., 2017 IEEE Power Energy Society General Meeting*, 1–5. New York: IEEE. https://doi.org/10.1109/PESGM.2017.8274033.

Gan, L., U. Topcu, and S. H. Low. 2012. "Stochastic distributed protocol for electric vehicle charging with discrete charging rate." In *Proc., 2012 IEEE Power and Energy Society General Meeting*, 1–8. New York: IEEE. https://doi.org/10.1109/PESGM.2012.6344847.

Gan, L., U. Topcu, and S. H. Low. 2013. "Optimal decentralized protocol for electric vehicle charging." *IEEE Trans. Power Syst.* 28 (2): 940–951. https://doi.org/10.1109/TPWRS.2012.2210288.

Geng, B., J. K. Mills, and D. Sun. 2013. "Two-stage charging strategy for plug-in electric vehicles at the residential transformer level." *IEEE Trans. Smart Grid* 4 (3): 1442–1452. https://doi.org/10.1109/TSG.2013.2246198.

Géron, A. 2018. "Neural networks and deep learning." Accessed October 29, 2019. http://proquest.safaribooksonline.com/9781492037354.

GO-Biz. 2013. "6. ZEV Action Plan A roadmap toward 1.5 million zero-emission vehicles on California roadways by 2025." Accessed October 14, 2019. http://www.business.ca.gov/ZEV-Action-Plan.

González Vayá, M., and G. Andersson. 2012. "Centralized and decentralized approaches to smart charging of plug-in vehicles." In *Proc., 2012 IEEE Power and Energy Society General Meeting*, 1–8. New York: IEEE. https://doi.org/10.1109/PESGM.2012.6344902.

González Vayá, M., M. D. Galus, R. A. Waraich, and G. Andersson. 2012. "On the interdependence of intelligent charging approaches for plug-in electric vehicles in transmission and distribution networks." In *Proc., 2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, 1–8. New York: IEEE. https://doi.org/10.1109/ISGTEurope.2012.6465891.

Kang, Q., J. Wang, M. Zhou, and A. C. Ammari. 2016. "Centralized charging strategy and scheduling algorithm for electric vehicles under a battery swapping scenario." *IEEE Trans. Intell. Transp. Syst.* 17 (3): 659–669. https://doi.org/10.1109/TITS.2015.2487323.

Li, Y., Y. Huang, and M. Zhang. 2018. "Short-term load forecasting for electric vehicle charging station based on niche immunity lion algorithm and convolutional neural network." *Energies* 11 (5): 1253. https://doi.org/10.3390/en11051253.

Ma, Z., D. S. Callaway, and I. A. Hiskens. 2013. "Decentralized charging control of large populations of plug-in electric vehicles." *IEEE Trans. Control Syst. Technol.* 21 (1): 67–78. https://doi.org/10.1109/TCST.2011.2174059.

Majidpour, M., C. Qiu, C.-Y. Chung, P. Chu, R. Gadh, and H. R. Pota. 2014. "Fast demand forecast of Electric Vehicle Charging Stations for cell phone application." In *Proc., 2014 IEEE PES General Meeting | Conf. Exposition*, 1–5. New York: IEEE. https://doi.org/10.1109/PESGM.2014.6938864.

Masoum, A. S., S. Deilami, A. Abu-Siada, and M. A. S. Masoum. 2015. "Fuzzy approach for online coordination of plug-in electric vehicle charging in smart grid." *IEEE Trans. Sustainable Energy* 6 (3): 1112–1121. https://doi.org/10.1109/TSTE.2014.2327640.

Masoum, A. S., S. Deilami, P. S. Moses, M. A. S. Masoum, and A. Abu-Siada. 2011. "Smart load management of plug-in electric vehicles in distribution and residential networks with charging stations for peak shaving and loss minimisation considering voltage regulation." *Transm. Distrib. IET Gener.* 5 (8): 877–888. https://doi.org/10.1049/iet-gtd.2010.0574.

Muzaffar, S., and A. Afshari. 2019. "Short-term load forecasts using LSTM networks." *Energy Procedia* 158 (Feb): 2922–2927. https://doi.org/10.1016/j.egypro.2019.01.952.

Panahi, D., S. Deilami, M. A. S. Masoum, and S. M. Islam. 2015. "Forecasting plug-in electric vehicles load profile using artificial neural networks." In *Proc., 2015 Australasian Universities Power Engineering Conf. (AUPEC)*, 1–6. Perth, Australia: Australasian Universities Power Engineering Conference. https://doi.org/10.1109/AUPEC.2015.7324879.

Qi, L., J. Yao, and X. Wang. 2018. "Improved particle swarm optimization algorithm to solve the problem of layout optimization of electric vehicle charging stations." *J. Highway Transp. Res. Dev. (English Ed.)* 12 (2): 96–103. https://doi.org/10.1061/JHTRCQ.0000631.

Ramos Muñoz, E., G. Razeghi, L. Zhang, and F. Jabbari. 2016. "Electric vehicle charging algorithms for coordination of the grid and distribution transformer levels." *Energy* 113 (Oct): 930–942. https://doi.org/10.1016/j.energy.2016.07.122.

Razeghi, G., L. Zhang, T. Brown, and S. Samuelsen. 2014. "Impacts of plug-in hybrid electric vehicles on a residential transformer using stochastic and empirical analysis." *J. Power Sources* 252 (Apr): 277–285. https://doi.org/10.1016/j.jpowsour.2013.11.089.

Ren, H., Q. Deng, F. Wen, J. Du, P. Yu, and J. Tian. 2021. "Joint planning of a distribution system and a charging network for electric vehicles." *J. Energy Eng.* 147 (1): 04020085. https://doi.org/10.1061/(ASCE)EY.1943-7897.0000734.

Sharma, I., C. Cañizares, and K. Bhattacharya. 2014. "Smart charging of PEVs penetrating into residential distribution systems." *IEEE Trans. Smart Grid* 5 (3): 1196–1209. https://doi.org/10.1109/TSG.2014.2303173.

Shmueli, B. 2020. "Multi-class metrics made simple, Part II: The F1-score." Accessed November 19, 2020. https://towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1.

Singh, M., K. Thirugnanam, P. Kumar, and I. Kar. 2015. "Real-time coordination of electric vehicles to support the grid at the distribution substation level." *IEEE Syst. J.* 9 (3): 1000–1010. https://doi.org/10.1109/JSYST.2013.2280821.

Smith, T. F. L. 2019. "Continuously varying valley filling smart charging techniques [UC Irvine]." Accessed August 12, 2019. https://escholarship.org/uc/item/5tq7k0q2.

Yi, Z., and D. Scoffield. 2018. "A data-driven framework for residential electric vehicle charging load profile generation." In *Proc., 2018 IEEE Transportation Electrification Conf. and Expo (ITEC)*, 519–524. New York: IEEE. https://doi.org/10.1109/ITEC.2018.8450228.

Zhang, L., F. Jabbari, T. Brown, and S. Samuelsen. 2014. "Coordinating plug-in electric vehicle charging with electric grid: Valley filling and target load following." *J. Power Sources* 267 (Dec): 584–597. https://doi.org/10.1016/j.jpowsour.2014.04.078.

Zhang, L., and Y. Li. 2017. "Optimal management for parking-lot electric vehicle charging by two-stage approximate dynamic programming." *IEEE Trans. Smart Grid* 8 (4): 1722–1730. https://doi.org/10.1109/TSG.2015.2505298.

Zhu, J., Z. Yang, M. Mourshed, Y. Guo, Y. Zhou, Y. Chang, Y. Wei, and S. Feng. 2019. "Electric vehicle charging load forecasting: A comparative study of deep learning approaches." *Energies* 12 (14): 2692. https://doi.org/10.3390/en12142692.