energies

Article

Data-Driven EV Load Profiles Generation Using a Variational Auto-Encoder

Zhixin Pan [1], Jianming Wang [2], Wenlong Liao [3,*], Haiwen Chen [3], Dong Yuan [1], Weiping Zhu [1], Xin Fang [2] and Zhen Zhu [4]

[1] State Grid Jiangsu Electric Power Co., Ltd., Nanjing 210029, China; zhxpan@js.sgcc.com.cn (Z.P.); william20180803@outlook.com (D.Y.); 15600647598@163.com (W.Z.)
[2] Electric Power Research Institute of State Grid Jiangsu Electric Power Co., Ltd., Nanjing 211103, China; jm@js.sgcc.com.cn (J.W.); 15251867073@126.com (X.F.)
[3] Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin 300072, China; chenhaiwen@tju.edu.cn
[4] State Grid Wuxi Power Supply Company, Wuxi 214062, China; vltavaz@163.com
* Correspondence: wenlongliao@cau.edu.cn

Received: 3 January 2019; Accepted: 20 February 2019; Published: 5 March 2019

Abstract: Although the penetration of electric vehicles (EVs) in distribution networks can improve the energy saving and emission reduction effects, its random and uncertain nature limits the ability of distribution networks to accept the load of EVs. To this end, establishing a load profile model of EV charging stations accurately and reasonably is of great significance to the planning, operation and scheduling of power system. Traditional generation methods for EV load profiles rely too much on experience, and need to set up a power load probability distribution in advance. In this paper, we propose a data-driven approach for load profiles of EV generation using a variational automatic encoder. Firstly, an encoder composed of deep convolution networks and a decoder composed of transposed convolution networks are trained using the original load profiles. Then, the new load profiles are obtained by decoding the random number which obeys a normal distribution. The simulation results show that EV load profiles generated by the deep convolution variational auto-encoder can not only retain the temporal correlation and probability distribution nature of the original load profiles, but also have a good restorative effect on the time distribution and fluctuation nature of the original power load.

Keywords: electric vehicles; load profiles; data-driven; variational automatic encoder

1. Introduction

With the increasingly serious environmental and energy problems, electric vehicles (EVs) have attracted more and more attention because of their enormous advantages in energy saving and emission reduction. The stochastic nature of EVs has brought new challenges to the scheduling, operation, and planning of distribution management systems. Since EV loads are strongly stochastic, how to accurately model the uncertainty of EV loads has become a hot issue. Stochastic scenario generation is a widely used approach to describe the uncertainties of EV loads. By considering a series of different EV load profiles, the distribution management system can take uncertainty into account when making decisions, such as safe operation of distribution network, optimal economic trade and cooperative scheduling strategy.

Although scholars have proposed many methods for load profile generation of EV loads, there are still many problems to be solved. For example, most of the proposed methods are based on probabilistic models [1–4]. These models are not able to accurately describe the true data distribution of the EV loads. The sampling from the high-dimensional distribution leads to a complex process. They are not able to take into account the time-varying, dynamic character and the complex temporal correlation of EV loads. In addition, probabilistic models require a large number of sampling to produce various load profiles. Common sampling methods include Monte Carlo sampling and Latin hypercube sampling. In order to improve the sampling efficiency, some scenario reduction algorithms such as backward reduction, forward selection, and scenario tree construction have been proposed [5]. However, these methods not only exacerbate the complexity of probabilistic models, but also limit the diversity of generated load profiles.

Recently, as an important subsidiary of machine learning, the deep learning has become one of the most active technologies in many research fields, including image recognition, fault detection, load forecasting, data generation and so on. Especially, the variational auto-encoder (VAE), which was originally introduced by Kingma et al [6], has received enormous attention in the realm of data generation. Effective applications based on VAE have been reported in many areas such as noise reduction [7], dimensionality reduction [8–10] and data generation [11,12]. At present, the application of VAE in data generation mainly focuses on generating MNIST handwritten digits and face images. To the best of our knowledge, there is no report about stochastic scenario generation of EV loads profiles using VAE.

In this paper, it is aimed to overcome these difficulties of generating EV load profiles. This approach has two key advantages: Firstly, the new approach is entirely data-driven. They don't need to sample the data in large quantities to fit the probability distribution, which greatly improves the efficiency of EV load profiles generation. The generated EV loads is diverse, and they preserve the temporal correlation between EV loads. Secondly, VAE belongs to unsupervised learning, which does not require labelled large data manually. The rest of this paper is organized as follows: Section 2 provides the background about EV load profile generation. Section 3 proposes and describes the VAE model. Section 4 discusses the simulations and results. The conclusions are described in Section 5.

2. Literature Review

In the past, many scholars have proposed different probability distributions to generate EV loads based on some reasonable assumption. For example, Zhou et al. analyzed the charging characteristics of different types of EVs and proposed a probability model to simulate charging demand [13]. The proposed methods can take amount of charging time may affect the charging start time. Simulation showed that the load on weekends is smaller than that on working days during the day and night EV charging peaks. In [14], the authors assumed that the distance travelled by all EVs was fixed every day. In [15], all EVs were allocated a certain percentage and a probability distribution was built to simulate the power consumption of light-duty plug-in EVs under uncontrolled and controlled charging scenarios. In [16,17], the proposed methods simulate load profiles according to different driving patterns and vehicle usage data. Lopes et al. assumed that each EV travels a certain distance and consumes the corresponding power load [18,19]. However, it is hard to capture the time-varying and dynamic character of EV loads by first and second order statistics alone. Generally speaking, these methods require that the travel habits of electric vehicles specify or fit the probability distribution of EV load profiles. Whether the hypothesis of probabilistic model is reasonable or not is directly related to the usability of the generated EV load profiles. Since daily routines and the lifestyle are varied [20–22], there is generally no unique or canonical distribution type suitable for modeling EV loads.

Another popular way to generate EV loads profile is to use time series. Amini et al. proposed the probabilistic hierarchical EVs' parking lot demand modeling and autoregressive moving average model to forecast the EV loads [23]. A Time Series Model is applied to approximate the load profile for a parking lot in [24]. In [25], time-series seasonal autoregressive integrated moving average (ARIMA) models are used for modeling and generating EVs electrical loads. Despite simple to implement, these methods are easy to overfitting. Since they simply memorize the training data, the generated EV loads is highly similar to historical data, making it difficult to generate many different types of load data.

Deep learning technology has been widely used to generate data in recent years. The generative adversarial network (GAN) is designed to generate wind power data in [26]. The simulation result shows that the generative adversarial network is able to generate realistic wind and photovoltaic power profiles with full diversity of behaviors. Similarly, the Bayesian generative adversarial network is applied to generate solar scenarios by using Bayesian probabilities in [27]. Wan et al. proposed a VAE-based synthetic data generation method for imbalanced learning. The VAE has better performance than the traditional synthetic sampling methods [28]. To extend reproduction of demonstration motion, the VAE is applied to generate time-series data in [29,30]. Although GAN has shown great success in the realistic data generation, the training is not easy. For example, it's hard to achieve Nash equilibrium. Compared with GAN, the training process of VAE is easier. In addition, VAE can directly compare the difference between generated data and original data, but GAN cannot do so. VAE has been widely used to generate two-dimensional data such as MNIST handwritten digits and face images, but the applications on one- dimensional data generation are relatively limited.

Taking the above analysis into consideration, it is clear that though the predecessors have made great success for load profiles generation of EVs. There are still some problems to be solved. For example, how to design a new approach which can capture time-varying and dynamic characteristics without specifying probability distribution manually? How to avoid constant sampling and improve the efficiency of the algorithm? How to generate different types of EV loads instead of simply remembering historical data?

In order to address these issues, a novel approach based on VAE framework is proposed to generate stochastic scenario of EV loads profiles. The key contributions of this paper can be summarized as follows:

(1) New technique: It is the first time in this paper to explore the feasibility of VAE in the application of load profiles generation for EVs. By employing VAE, we can generate load profiles which capture the time-varying and dynamic nature of EV loads.

(2) Simple implement: Unlike traditional methods, the proposed approach can automatically learn the inherent nature of the EV loads, without the need to manually specify the probability distribution, which is suitable for generating EV loads of different time and space. In addition, it does not require a large number of samples to fit the probability distribution, which improves the efficiency of the algorithm.

(3) Conditional load profiles generation: The proposed approaches provide two ways to generate conditional load profiles with specific characteristics. The first method uses the specified load data as the VAE input to generate the conditional load profiles. Another method takes the normal distribution data as the input of the decoder, and the decoder generates a large number of load profiles. Then, the load profiles are classified according to the characteristics, so as to get the required conditional load profiles.

3. Methodology

This part mainly introduces the following four sections: the principle of VAE, the method of generating conditional load profiles, the basic steps of the approaches and the indicators for evaluating results.

3.1. Variational Auto-Encoder

As one of the automatic encoders, the variational auto-encoder obtains a generator which can generate similar data to the original load through limited training samples. The conventional automatic encoder takes the original data as an input and compresses the input data by the encoder. Then, the generated new data is acquired by the decoder. In other words, if a conventional automatic encoder wants to generate N load profiles, N original load profiles are required as inputs. In this case, the style and number of generated data by the traditional automatic encoder are limited by the original data, so it is difficult to generate different styles of data, and the structure of the generated data is single. To overcome the shortcomings of traditional automatic encoder, the variational auto-encoder adds noise data that satisfy normal distribution to the latent variable of the encoder, and then takes the two variables as the input of the decoder to generate the new data.

Normally, the VAE is designed to learn the data distribution $p_\theta(X)$ of original load profiles according to power load time series $X = \{x^1, \ldots, x^n\}$. Typically, this data distribution of EV loads is decomposed as follows:

$$p_\theta(X) = \prod_{i=1}^{n} p_\theta(x^i) \tag{1}$$

The log function is utilized to address the numerical issues, and the mathematical formula is as follows:

$$\log \prod_{i=1}^{n} p_\theta(x^i) = \prod_{i=1}^{n} \log p_\theta(x^i) \tag{2}$$

Each time series profile of EVs includes the so-called latent variable z, which explains the generative process. We can rewrite Equation (1) for a single point as:

$$p_\theta(x) = \int p_\theta(x, z)dz = \int p_\theta(z) p_\theta(x|z)dz \tag{3}$$

The generation process for EV loads profiles includes various steps. Firstly, the latent variable z is drawn following the prior probability $p_{\theta^*}(z)$. Secondly, an EV loads is generated based on the posterior probability $p_{\theta^*}(x|z)$. However, we don't know the prior $p_{\theta^*}(z)$ and the likelihood $p_{\theta^*}(x|z)$. In order to solve this problem, we need to know:

$$p_\theta(z|x) = \frac{p_\theta(x|z) p_\theta(z)}{p_\theta(x)} \tag{4}$$

Therefore, inference is difficult to carry out. We have to estimate this function approximately by $q_\phi(z|x)$ and set of parameters ϕ.

We need to approximate $\log p_\theta(x^i)$, because the data that follow this distribution cannot be sampled directly. To this end, we can combine the variational lower bound and the Kullback-Leibler divergence:

$$\log p_\theta(x) = D_{KL}\big(q_\phi(z|x) \| p_\theta(z|x)\big) + L(\theta, \phi; x) \tag{5}$$

Because we are calculating the difference between the true posterior $p_\theta(z|x)$ and the approximate $q_\phi(z|x)$, considering that this divergence is bigger than 0, then term $L(\theta, \phi; x)$ acts as a lower bound of the log likelihood:

$$\log p_\theta(x) \geq L(\theta, \phi; x) \tag{6}$$

$$L(\theta, \phi; x) = E_{q_\phi(z|x)}\left[-\log q_\phi(z|x) + \log p_\phi(x|z)\right] \tag{7}$$

That could be written as:

$$L(\theta, \phi; x) = -D_{KL}\big(q_\phi(z|x) \| p_\theta(z)\big) + E_{q_\phi(z|x)}[\log p_\theta(x|z)] \tag{8}$$

where the first term related to the KL-divergence constrains the function $q_\phi(z|x)$ to the shape of the $p_\theta(z)$. The second term can reconstruct the input with a given latent variable z that follows $p_\theta(x|z)$.

Using this loss function, we are able to parameterize the model as follows:

$$q_\phi(z|x) = q(z; f(x, \phi)) \tag{9}$$

$$p_\theta(x|z) = p(x; g(z, \theta)) \tag{10}$$

where f and g are deep neural networks with the set of parameters θ and ϕ, respectively. More details can be found in the [31]. The back propagation algorithm can be used to re-parameterize the generation of the vector z.

The key advantage of VAE that we can use some samples to train it in an unsupervised way, and then encode EV load profiles in the latent space without any effort. Once the origin load profiles are encoded, we can modify them in this space, and then reconstruct or decode the altered vector to get some new load profiles. A standard structure of variational auto-encoder is shown in Figure 1.

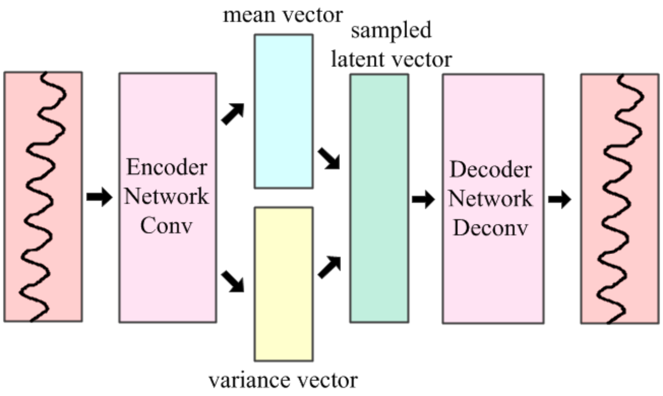


Figure 1. Structure of the VAE.

The encoder encodes the load profiles into the latent space and outputs two vectors describing the mean and variance of the latent state distributions. Then, the decoder model will generate a latent vector by sampling from these defined distributions and proceed to develop a reconstruction of the original input.

3.2. K-nearest Neighbor Algorithms

The VAE trains the depth convolution network using the original power load profiles, and then takes the random number obeying the normal distribution as the input of the decoder to get the new power load. The types of load profiles generated are various, so we need to process them further to obtain the label load profiles. In this paper, we will use the k-nearest neighbor algorithm (KNN) to classify the generated load profiles, so as to get the conditional load profiles.

The k-nearest neighbor algorithm is a non-parametric method widely used for classification. The input of KNN includes k closest training examples in the feature space and the output of KNN is a class membership. An object is classified by a majority vote of its neighbors and the object will be assigned to the class most common among its k nearest neighbors. The KNN can be implemented by following the below steps:

(1) Load the data and initialize the value of k.
(2) Calculate the Euclidean distance between test load profiles and each row of training load profiles.
(3) Sort the calculated distances in ascending order based on distance values.
(4) Get top k rows from the sorted array.
(5) Get the most frequent class of these rows.

3.3. Indicators for Evaluating Results

Unlike load power forecasting, the generated power load is not required to be exactly the same as the actual power load in each time interval. In order to evaluate the similarity between the generated load profiles and the actual load profiles, this paper will verify the validity of the generated load profiles from four aspects: probability distribution of power loads, the temporal correlation of power loads, duration of power loads and volatility of power loads:

(1) Probability distribution of power loads: The power load is divided into several intervals, and the number of load data falling in each interval is calculated. Then, the ratio of the number of loads in each interval to the total load is calculated, and the corresponding probability histogram or the probability distribution of power loads is obtained by connecting the scattered points.

(2) The temporal correlation of power loads: Here autocorrelation function will be utilized to evaluate the temporal correlation of power loads since it's the most popular method. It is required that the temporal correlation between the generated power load profiles and the original power load profiles be consistent. The mathematical formula of autocorrelation function is as follows:

$$R(\tau) = \frac{E[(X_t - \mu)(X_{t+\tau} - \mu)}{\sigma^2} \tag{11}$$

where μ is the mean of X_t; σ is standard deviation; $\tau$is the lag time.

(3) Duration of power loads: The duration of load power indicates the running time when the power load is greater than a certain value. The duration of power load can be expressed as follows:

$$q_{ij} = \begin{cases} 1, P_i \geq P_j, \text{if } P_j > 0 \\ 0, P_i < P_j, \text{if } P_j > 0 \\ 1, P_i < P_j, \text{if } P_j \leq 0 \\ 0, P_i \geq P_j, \text{if } P_j \leq 0 \end{cases} \quad i = 1 \sim n, j = 1 \sim m \tag{12}$$

$$t_j = \sum_{i=1}^{n} q_{i,j} \tag{13}$$

where: P_i represents item i of the power load time series; n represents the total length of load time series; m is the number of intervals of load profiles; $q_{i,j}$ is a variable used to count variables; t_j is the total time when the load is greater than P_j.

(4) Volatility of power loads: In order to evaluate the volatility of power load, the absolute values of the two adjacent power load differences are calculated first, and then the probability distributions of these absolute values are calculated.

3.4. Procedures of Generating Load Profiles Based on VAE

According to the above analysis, the produce of EV load profiles generation using variational auto-encoder is as follows:

(1) Data normalization: The data of load profiles needs to be normalized before the data is assigned to the encoder, otherwise the loss function may not converge. In this paper, the min-max normalization method will be used to transform the input data into the interval [0,1].

(2) Coding the load profiles: The encoder constructed by deep convolution network maps the input data to the low dimensional vector space.

(3) Sampling data: According to the mean and variance of the encoder output, some random number obeying the normal distribution is generated, and they will be used as the input of the decoder.

(4) Decoding data: Taking the data of the sampling layer as input, the new load profiles can be obtained by the decoder composed of the deep transposed convolution network.

(5) Gradient back propagation: The loss function is calculated using the output data from the decoder and the original load data. Then, the back-propagation method will be utilized to continuously adjust the weights and thresholds.

(6) Generating load profiles: After the iteration, we can use the trained VAE to generate new load profiles. The proposed approaches provide two ways to generate new load profiles. The first

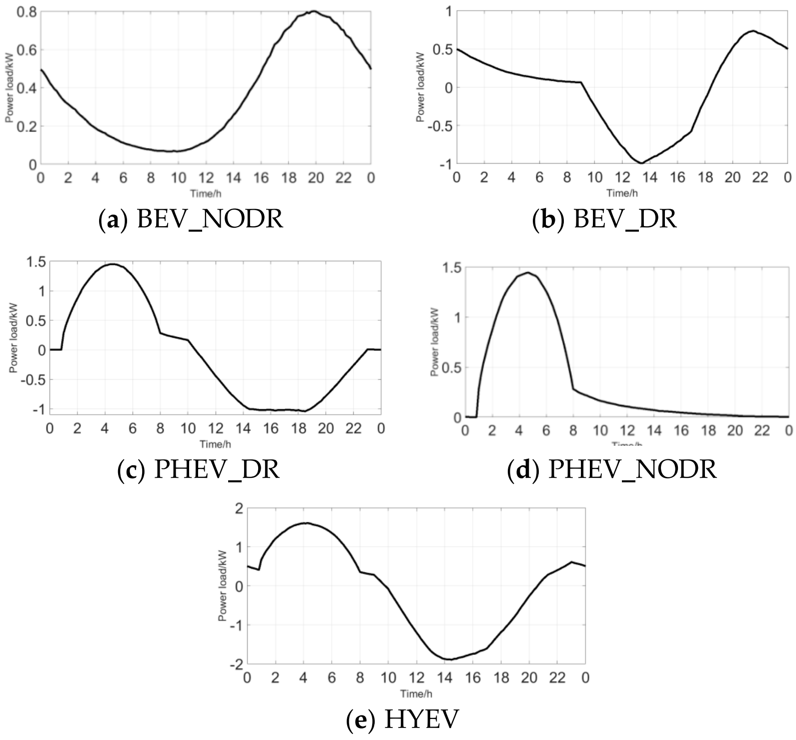
method uses the specified load data as the VAE input to generate the conditional load profiles. Another way takes the normal distribution data as the input of the decoder, and the decoder generates a large number of load profiles. Then, the load profiles are classified by KNN, so as to get the required conditional load profiles.

4. Study Case

4.1. Parameters and Structure

It assumes that EVs mainly include the following five classes: Battery EV without demand response (BEV_NODR), Battery EV with demand response (BEV_DR), Plug-in hybrid EV without demand response (PHEV_NODR), Plug-in hybrid EV with demand response (PHEV_DR) and the above four hybrid EVs (HYEV).

The modeling details of the load curves for these EVs are shown in [32–36]. It assumes that they charge and discharge in a charging station. The typical load profiles of the 5 classes' EVs are shown in Figure 2. The load profiles are recorded every 10 minutes, so there are 144 points per load profiles. All approaches are implemented on a laptop with a 2.4 GHz Intel i3 processor and 6 GB of memory using the Keras library with Theano backend. All weights of network are initialized from a zero-centered normal distribution. The VAE is trained with mini-batch Adam with a mini-batch size of 50. After testing the parameters of different structures and parameters, the optimal structure and parameters are as follows:



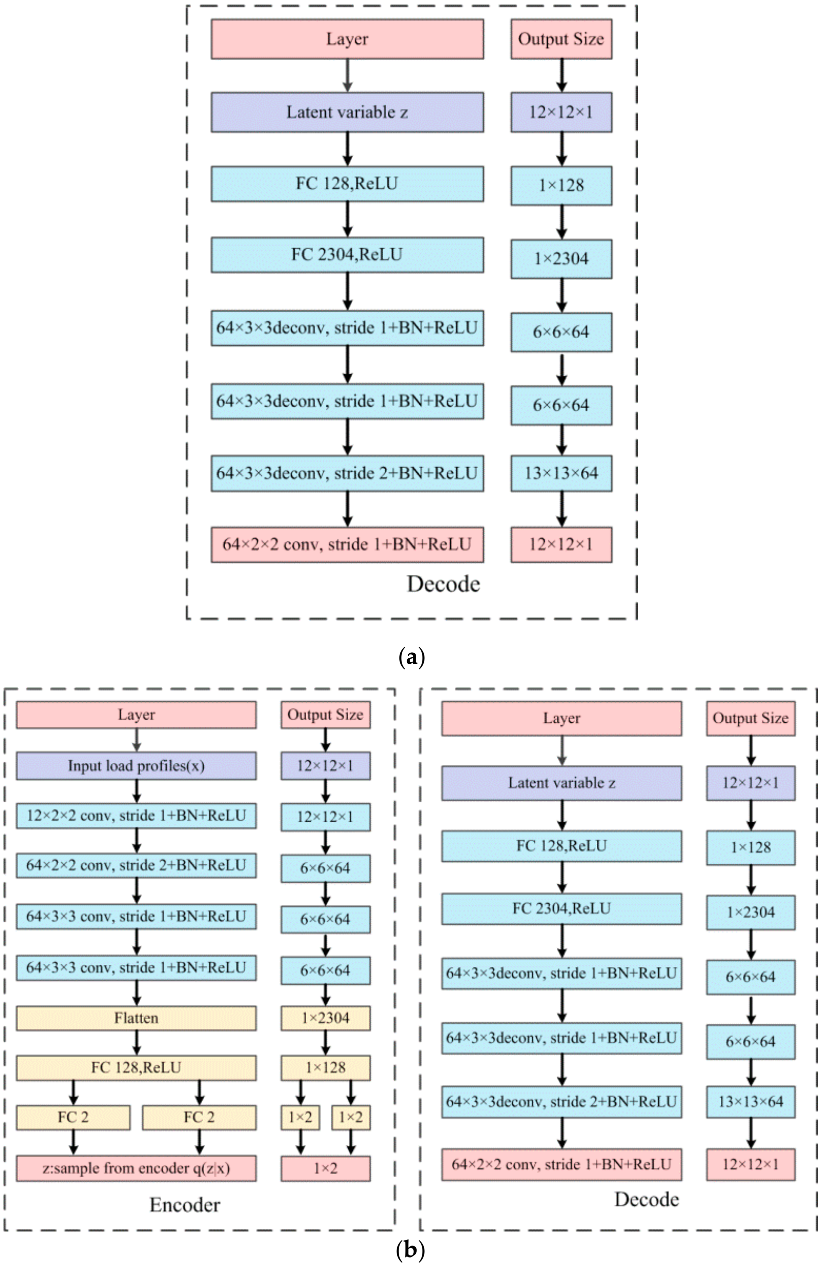
(**a**) BEV_NODR (**b**) BEV_DR

(**c**) PHEV_DR (**d**) PHEV_NODR

(**e**) HYEV

Figure 2. Typical load profiles of five classes EV.

The input load profiles consist of 144 values reconstructed into a 12 × 12 × 1 data matrix. We construct four convolutional layers in the encoder network. The convolution kernel size of convolution layer 1 is 2 × 2 and convolution stride is 1. The convolution kernel size of convolution layer 2 is 2 × 2 and convolution stride is 2. The convolution kernel size of convolution layer 3 and layer 4 is 3 × 3 and convolution stride is 1. Each convolutional layer is followed by a batch normalization layer and a ReLU activation function. In order to sample latent variable z and calculate the KL divergence loss, two fully-connected layers for mean and variance are added to encoder. For decoder, we use 4 transposed convolutional layers and a convolutional layer. The transposed convolution kernel size of convolution layer 1 and layer 2 is 3 × 3 and convolution stride is 1. The transposed convolution kernel size of convolution layer 3 is 3 × 3 and convolution stride is 2. The convolution kernel size of convolution layer 1 is 2 × 2. We also use ReLU as the activation function and use batch normalization to help stabilize training. The details of architecture are shown in Figure 3.


Layer
Output Size
Latent variable z
12×12×1
FC 128,ReLU
1×128
FC 2304,ReLU
1×2304
64×3×3deconv, stride 1+BN+ReLU
6×6×64
64×3×3deconv, stride 1+BN+ReLU
6×6×64
64×3×3deconv, stride 2+BN+ReLU
13×13×64
64×2×2 conv, stride 1+BN+ReLU
12×12×1
Decode
Layer
Output Size
Input load profiles(x)
12×12×1
12×2×2 conv, stride 1+BN+ReLU
12×12×1
64×2×2 conv, stride 2+BN+ReLU
6×6×64
64×3×3 conv, stride 1+BN+ReLU
6×6×64
64×3×3 conv, stride 1+BN+ReLU
6×6×64
Flatten
1×2304
FC 128,ReLU
1×128
FC 2
FC 2
1×2
1×2
z:sample from encoder q(z|x)
1×2
Encoder
Layer
Output Size
Latent variable z
12×12×1
FC 128,ReLU
1×128
FC 2304,ReLU
1×2304
64×3×3deconv, stride 1+BN+ReLU
6×6×64
64×3×3deconv, stride 1+BN+ReLU
6×6×64
64×3×3deconv, stride 2+BN+ReLU
13×13×64
64×2×2 conv, stride 1+BN+ReLU
12×12×1
Decode

(**a**)

(**b**)

Figure 3. Architecture and parameters of auto-encoder network. (**a**) Architecture and parameters of encoder; and, (**b**) Architecture and parameters of decoder.

This section mainly consists of the following three points: (1) Visualizing the natural load profiles based on the similarity of their latent representations. (2) The load profiles are generated, and the similarity between the new load profiles and the original load profiles are evaluated by the proposed indicators. (3) The conditional load profiles are generated by two ways, and the performance is evaluated.

4.2. Visualization of Latent Vectors

In addition to directly observing the results of the generated load profiles by using the proposed four indicators, there is another way to evaluate the new load profiles. We can observe the empirical distribution of generated load profiles by sampling latent vectors. The latent vectors are the encoding representation of the natural load profiles. According the similarity of latent representations, we are able to visualize the natural load profiles. The principle of the approaches is shown in Figure 4, and the two steps to implement the approaches are as follows:

(1) It assumes that the dimension of the latent vectors is 2. Then, the generated load profiles are mapped to 2-dimensional spaces, which make it more convenient to visualize.

(2) Since the dimension of the latent vectors is 2, we can sample a batch of data at equal intervals from the two-dimensional space, which can represent the empirical distribution of the latent vectors in the whole two-dimensional space. Then, the sampled data is processed by the decoder, and the new load profiles can be obtained.

The visualization of 15 × 15 load profiles is shown in Figure 5.

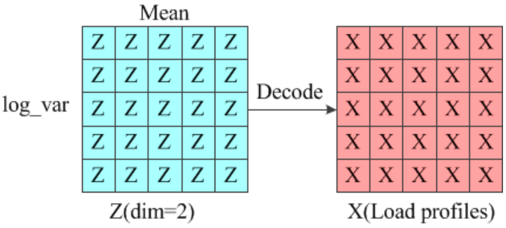


Figure 4. Visualization of latent vectors.

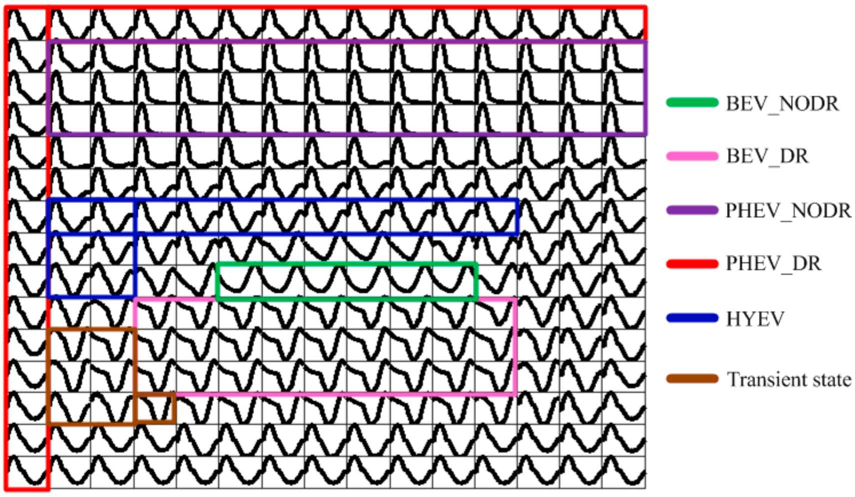


Figure 5. Visualization of 15 × 15 load profiles based on latent vectors.

We can see that load profiles with similar shapes tend to be clustered as a group. A variety of different classes of load profiles appear in this two-dimensional plane, and there are some transitional regions between different classes of load profiles. The load profiles in the transition area have a variety of load characteristics, and the profiles of these power loads are similar.

4.3. Generating Load Profiles

To verify the effectiveness of the proposed algorithm, we assume that there are 20-60 HYEV per day at a charging station. Nine thousand (9000) load profiles are obtained through simulation. Among them, 8000 load profiles are randomly selected as the training set, and the remaining data is used as the test set. To illustrate the effectiveness of the proposed approaches, we will compare the performance of VAE and GAN networks. The parameters of GAN are as follows: the input of the generator is random noise with dimension 50. The generator consists of three full-connected layers, and the number of neurons is 48, 96 and 144, respectively. The discriminator also consists of three layers of full-junction layers, with 94, 48 and one neuron, respectively. The performance of original load profiles and the generated load profiles are shown in Figures 6–10.

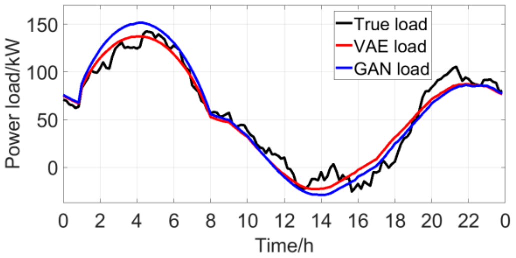


Figure 6. The true load profile and generated load profile.

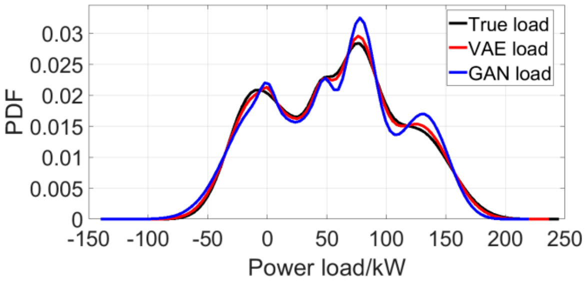


Figure 7. Probability density functions of true load and generated load.

The load profiles generated in Figure 6 are randomly selected. Then, the dynamic time warping distance between each load profile in the test set and the generated load profile is calculated. Finally, the true load profile with minimum dynamic time warping distance is selected. It can be seen from the Figure 6 that the generated load profiles are very similar to the true load profiles that did not participate in the training process, which shows that the generated data are very close to the true data. Figure 7 shows the probability density functions for 8000 training sets and the generated load profiles drawn by ksdensite function in MATLAB2018a. Obviously, the probability of each power load appearing is similar in the generated load profile and the original load profile, which shows that the proposed methods can well fit the probability distribution of the original power load. In addition, it can be found from above Figures that the load profiles generated by VAE is closer to the true profiles than that generated by GAN.

In order to analyze the time characteristics of the generated load profiles, Figure 8 shows the autocorrelation function of the load profile for a week that is randomly selected, and Figure 9 is the duration of the power loads.

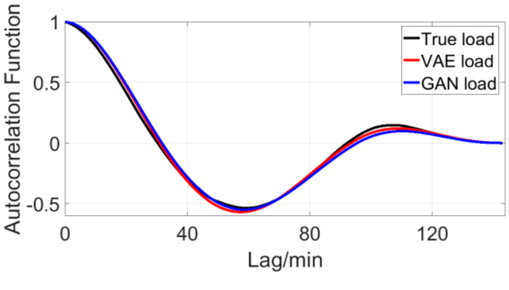


Figure 8. The autocorrelation function of load.

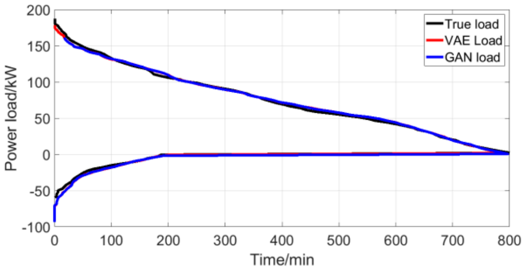


Figure 9. Duration of load profiles.

The autocorrelation functions of true load and generated loads are similar, which implies that the generated load profiles have the same temporal correlation as the original load profiles. It can be seen from Figure 9 that the generated load profile and the actual load profile and the area enclosed by the *y*-axis are approximately equal, which indicates that the VAE well restores the power demand of the EV charging station for one week.

Further, Figure 10 shows the probability density function of the load variation per hour. Obviously, the fluctuation of the simulated load profile is almost the same as the real situation, which further illustrates the effectiveness of the proposed approaches.

As far as the autocorrelation function, duration of load profiles, load variation and probability density function are concerned, VAE can imitate the real load profiles well, which is slightly better than GAN.

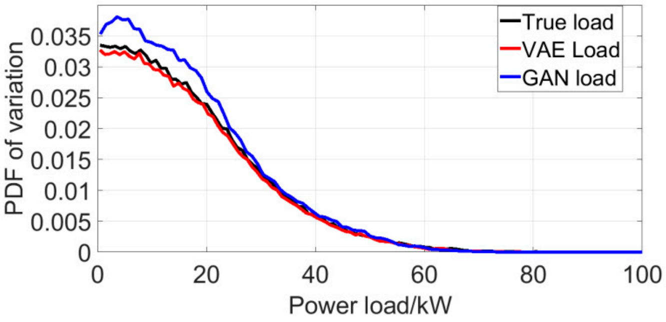


Figure 10. Probability density function of load variation.

4.4. Conditional Load Profiles Generation

The VAE offers two methods to generate conditional load profiles. The first method takes the original loads as input and obtains new load profiles through the VAE model. Another method is to take the data obeying the normal distribution as an input and obtain new load profiles through the decoder model. To test the performance of these two methods, we assume that there are four classes' electric cars per day at a charging station. In addition, it assumes that the number of BEV_NODR is a random number ranges from 40 to 70. The number of BEV_DR is a random number ranges from 30 to 50. The number of PHEV_NODR and PHEV_DR are a random number ranges from 20 to 50. 9000 load profiles are obtained through simulation. Among them, 8000 load profiles are randomly selected as the training set, and the remaining data is used as the test set. After many testing, the parameter k of the K-nearest neighbor algorithms is assumed to be 30. The performance of original load profiles and the generated load profiles are shown in Figures 11–14.

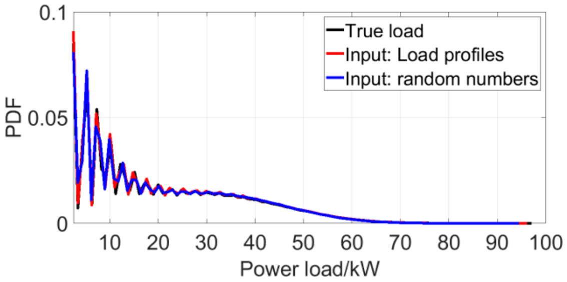


Figure 11. Probability density function of BEV_NODR.

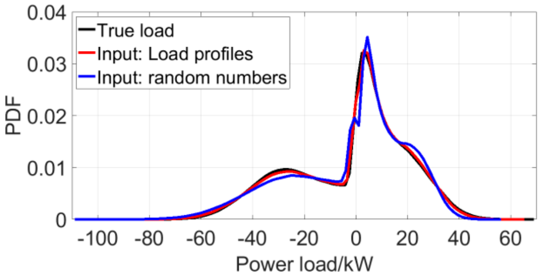


Figure 12. Probability density function of BEV_DR.

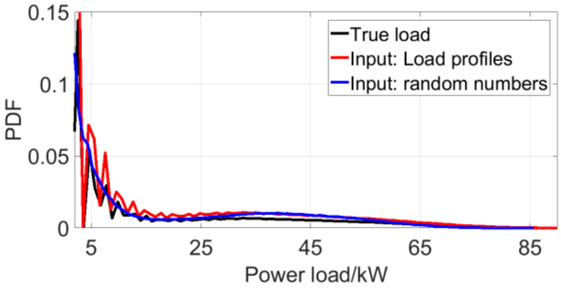


Figure 13. Probability density function of PHEV_NODR.

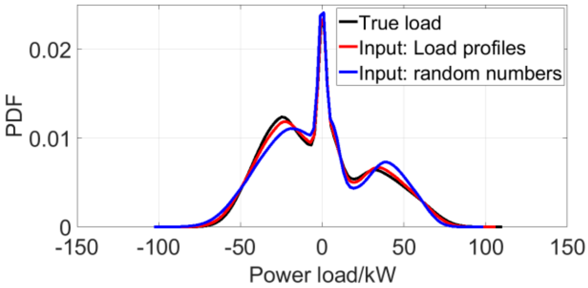


Figure 14. Probability density function of PHEV_DR.

Obviously, the probability density function of the load profiles generated by the first method is highly consistent with the probability density function of the true load profiles, which shows that the first method can produce the required label load profiles well. The number of new load profiles produced by the first method is limited by the number of real loads. In other words, the number of conditional load profiles generated is less than or equal to the number of original conditional load profiles. In addition, the load profiles generated by the first method are highly similar to the input load profiles, and the type of new load profiles is limited. There is a certain deviation between the probability density function of the load profiles generated by the second method and that of the original load profiles, because some transition state profiles are also divided into conditional load profiles. Compared with the first method, the type of conditional load profiles produced by the second method is more abundant because of the existence of load profiles in transition state.

5. Conclusions

It is of great practical significance for the operation, optimization and management of power systems to master the principle of potential load changes and generate data similar to historical loads. In this paper, a novel machine learning technique, the variational auto-encoder, is proposed to generate EV load profiles. The proposed approach can automatically learn the inherent nature of the EV loads, without the need to manually specify the probability distribution, which is suitable for generating EV loads of different time and space. Through simulation, we deduce the following conclusions:

(1) Traditional methods mainly fit the EV load distribution manually according to the driving habits of electric vehicles. Since daily routines and lifestyle vary, there is generally no unique or canonical distribution type suitable for modeling EV loads. The proposed approach can automatically learn the inherent nature of the EV loads, without the need to manually specify the probability distribution, which is suitable for generating EV loads of different time and space.

(2) By mapping the load profiles to 2-D space and sampling the latent vectors, it is found that similar load profiles are grouped into a class. There are some transition spaces between the two classes of load profiles, and the load profiles in the transition space have many characteristics of load profiles.

(3) With the noise data obeying normal distribution as input, the load profiles similar to the original data can be obtained by the trained decoder. By comparing the generated load profiles with the real load profiles, it is found that VAE can not only accurately capture the temporal correlation and probability distribution characteristics of the original load profiles, but also preserve the volatility of the original load profiles. In addition, the power consumption of the generated load profiles and that of the original load profiles are also consistent over a period of time.

(4) The proposed approaches provide two ways to generate conditional load profiles with specific characteristics. The probability density function of the load profiles generated by the first method is highly consistent with the probability density function of the true load profiles, which shows that the first method can produce the required label load profiles well. Compared with the first method, the second method can generate an infinite number of conditional load profiles, and the conditional load profiles are also richer in style.

Author Contributions: Conceptualization, W.L.; Data curation, J.W. and D.Y.; Investigation, W.Z.; Methodology, W.L.; Resources, Z.Z.; Software, Z.P. and X.F.; Supervision, H.C.

Funding: This research received no external funding.

Acknowledgments: The authors are grateful to Shouxiang Wang for his suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Badea, G.; Felseghi, R.-A.; Varlam, M.; Filote, C.; Culcer, M.; Iliescu, M.; Răboacă, M. Design and simulation of romanian solar energy charging station for electric vehicles. *Energies* **2018**, *12*, 74. [CrossRef]
2. Deb, S.; Tammi, K.; Kalita, K.; Mahanta, P. Impact of electric vehicle charging station load on distribution network. *Energies* **2018**, *11*, 178. [CrossRef]
3. De Gennaro, M.; Paffumi, E.; Scholz, H.; Martini, G. Gis-driven analysis of e-mobility in urban areas: An evaluation of the impact on the electric energy grid. *Appl. Energy* **2014**, *124*, 94–116. [CrossRef]
4. Harris, C.B.; Webber, M.E. An empirically-validated methodology to simulate electricity demand for electric vehicle charging. *Appl. Energy* **2014**, *126*, 172–181. [CrossRef]
5. Dupačová, J.; Gröwe-Kuska, N.; Römisch, W. Scenario reduction in stochastic programming. *Math. Program.* **2003**, *95*, 493–511. [CrossRef]

6. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
7. Zhang, J.; Zhang, Y.; Bai, L.; Han, J. Lossless-constraint denoising based auto-encoders. *Signal Process. Image Commun.* **2018**, *63*, 92–99. [CrossRef]
8. Wang, Y.; Yao, H.; Zhao, S. Auto-encoder based dimensionality reduction. *Neurocomputing* **2016**, *184*, 232–242. [CrossRef]
9. Taghanaki, S.A.; Kawahara, J.; Miles, B.; Hamarneh, G. Pareto-optimal multi-objective dimensionality reduction deep auto-encoder for mammography classification. *Comput. Methods Progr. Biomed.* **2017**, *145*, 85–93. [CrossRef] [PubMed]
10. Wang, X.; Kong, Y.; Cheng, Y. Dimensionality reduction for hyperspectral data based on sample-dependent repulsion graph regularized auto-encoder. *Chin. J. Electron.* **2017**, *26*, 1233–1238. [CrossRef]
11. Kiasari, M.A.; Moirangthem, D.S.; Lee, M. Coupled generative adversarial stacked auto-encoder: Cogasa. *Neural Netw.* **2018**, *100*, 1–9. [CrossRef] [PubMed]
12. Lin, S.-Y.; Chiang, C.-C.; Li, J.-B.; Hung, Z.-S.; Chao, K.-M. Dynamic fine-tuning stacked auto-encoder neural network for weather forecast. *Future Gener. Comput. Syst.* **2018**, *89*, 446–454. [CrossRef]
13. Zhou, N.; Xiong, X.; Wang, Q. Probability model and simulation method of electric vehicle charging load on distribution network. *Electr. Power Compon. Syst.* **2014**, *42*, 879–888. [CrossRef]
14. Kintner-Meyer, M.; Schneider, K.; Pratt, R. Impacts Assessment of plug-in Hybrid Vehicles on Electric Utilities and Regional us Power Grids, Part 1: Technical Analysis. Available online: https://scholar.google.com/scholar?hl=zh-CN&as_sdt=0%2C5&q=Impacts+assessment+of+plug-in+hybrid+vehicles+on+electric+utilities+and+regional+us+power+grids%2C+part+1%3A+Technical+analysis&btnG= (accessed on 3 January 2019).
15. Wu, D.; Aliprantis, D.C.; Gkritza, K. Electric energy and power consumption by light-duty plug-in electric vehicles. *IEEE Trans. Power Syst.* **2011**, *26*, 738–746. [CrossRef]
16. Yu, H.; Tseng, F.; McGee, R. Driving pattern identification for ev range estimation. In Proceedings of the 2012 IEEE International Electric Vehicle Conference, Greenville, South Carolina, 4–8 March 2012; pp. 1–7.
17. Ashtari, A.; Bibeau, E.; Shahidinejad, S.; Molinski, T. Pev charging profile prediction and analysis based on vehicle usage data. *IEEE Trans. Smart Grid* **2012**, *3*, 341–350. [CrossRef]
18. Lopes, J.A.P.; Soares, F.J.; Almeida, P.M.R. Integration of electric vehicles in the electric power system. *Proc. IEEE* **2011**, *99*, 168–183. [CrossRef]
19. Clement-Nyns, K.; Haesen, E.; Driesen, J. The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. *IEEE Trans. Power Syst.* **2010**, *25*, 371–380. [CrossRef]
20. Flath, C.M.; Gottwalt, S.; Ilg, J.P. A revenue management approach for efficient electric vehicle charging coordination. In Proceedings of the 2012 45th Hawaii International Conference on System Sciences, 4–7 January 2012; pp. 1888–1896.
21. Greene, D.L. Estimating daily vehicle usage distributions and the implications for limited-range vehicles. *Transp. Res. Part B Methodol.* **1985**, *19*, 347–358. [CrossRef]
22. Schäuble, J.; Kaschub, T.; Ensslen, A.; Jochem, P.; Fichtner, W. Generating electric vehicle load profiles from empirical data of three ev fleets in southwest germany. *J. Clean. Prod.* **2017**, *150*, 253–266. [CrossRef]
23. Amini, M.H.; Karabasoglu, O.; Ilić, M.D.; Boroojeni, K.G.; Iyengar, S.S. Arima-based demand forecasting method considering probabilistic model of electric vehicles' parking lots. In Proceedings of the 2015 IEEE Power & Energy Society General Meeting, Denver, CO, USA, 26–30 July 2015; pp. 1–5.
24. Paterakis, N.G.; Gibescu, M. A methodology to generate power profiles of electric vehicle parking lots under different operational strategies. *Appl. Energy* **2016**, *173*, 111–123. [CrossRef]
25. Louie, H.M. Time-series modeling of aggregated electric vehicle charging station load. *Electr. Power Compon. Syst.* **2017**, *45*, 1498–1511. [CrossRef]
26. Chen, Y.; Wang, Y.; Kirschen, D.; Zhang, B. Model-free renewable scenario generation using generative adversarial networks. *IEEE Trans. Power Syst.* **2018**, *33*, 3265–3275. [CrossRef]
27. Chen, Y.; Li, P.; Zhang, B. Bayesian renewables scenario generation via deep generative networks. In Proceedings of the 2018 52nd Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, 21–23 March 2018; pp. 1–6.
28. Wan, Z.; Zhang, Y.; He, H. Variational autoencoder based synthetic data generation for imbalanced learning. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–7.

29. Takahashi, D.; Katsura, S. Extended reproduction of demonstration motion using variational autoencoder. In Proceedings of the 2018 IEEE 27th International Symposium on Industrial Electronics (ISIE), Cairns, QLD, Australia, 13–15 June 2018; pp. 1057–1062.
30. Wand, S.; Chen, H.; Li, X.; Su, X. In Conditional Variational Automatic Encoder Method for Stochastic Scenario Generation of Wind Power and Photovoltaic System. *Power Syst. Technol.* **2018**, *42*, 1860–1867.
31. Kingma, D.P.; Welling, M. Stochastic Gradient VB and the Variational Auto-Encoder. Available online: https://scholar.google.com/scholar?q=Stochastic%20Gradient%20VB%20and%20the%20Variational%20Auto-Encoder%20Kingma%202013 (accessed on 3 January 2019).
32. Khorramdel, B.; Khorramdel, H.; Aghaei, J.; Heidari, A.; Agelidis, V.G. Voltage security considerations in optimal operation of bevs/phevs integrated microgrids. *IEEE Trans. Smart Grid* **2015**, *6*, 1575–1587. [CrossRef]
33. Dupont, B.; Dietrich, K.; De Jonghe, C.; Ramos, A.; Belmans, R. Impact of residential demand response on power system operation: A belgian case study. *Appl. Energy* **2014**, *122*, 1–10. [CrossRef]
34. Turker, H.; Florescu, A.; Bacha, S.; Chatroux, D. Load rates of low voltage transformers and medium voltage profile assessments on a real distribution electric grid based on average daily load profile (dlp) of a housing for a high penetration of plug-in hybrid electric vehicles (phevs). In Proceedings of the 2011 IEEE Vehicle Power and Propulsion Conference, Chicago, IL, USA, 6–9 September 2011; pp. 1–8.
35. Shao, S.; Zhang, T.; Pipattanasomporn, M.; Rahman, S. Impact of tou rates on distribution load shapes in a smart grid with phev penetration. In Proceedings of the IEEE PES T&D 2010, New Orleans, LA, USA, 19–22 April 2010; pp. 1–6.
36. Darabi, Z.; Ferdowsi, M. Aggregated impact of plug-in hybrid electric vehicles on electricity demand profile. *IEEE Trans. Sustain. Energy* **2011**, *2*, 501–508. [CrossRef]