energies

MDPI

Article

Data Augmentation for Electricity Theft Detection Using Conditional Variational Auto-Encoder

Xuejiao Gong [1], Bo Tang [1], Ruijin Zhu [1], Wenlong Liao [2,*] and Like Song [3]

[1] Electric Engineering College, Tibet Agriculture and Animal Husbandry University, Nyingchi 860000, China; gxj2018@hhu.edu.cn (X.G.); tangbo@xza.edu.cn (B.T.); zhuruijin@xza.edu.cn (R.Z.)

[2] Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin 300072, China

[3] Maintenance Branch of State Grid Jibei Electric Power Co., Ltd., Beijing 102488, China; like_song@tju.edu.cn

* Correspondence: wenlongliao@cau.edu.cn; Tel.: +86-17320211455

Received: 19 July 2020; Accepted: 17 August 2020; Published: 19 August 2020

Abstract: Due to the strong concealment of electricity theft and the limitation of inspection resources, the number of power theft samples mastered by the power department is insufficient, which limits the accuracy of power theft detection. Therefore, a data augmentation method for electricity theft detection based on the conditional variational auto-encoder (CVAE) is proposed. Firstly, the stealing power curves are mapped into low dimensional latent variables by using the encoder composed of convolutional layers, and the new stealing power curves are reconstructed by the decoder composed of deconvolutional layers. Then, five typical attack models are proposed, and the convolutional neural network is constructed as a classifier according to the data characteristics of stealing power curves. Finally, the effectiveness and adaptability of the proposed method is verified by a smart meters' data set from London. The simulation results show that the CVAE can take into account the shapes and distribution characteristics of samples at the same time, and the generated stealing power curves have the best effect on the performance improvement of the classifier than the traditional augmentation methods such as the random oversampling method, synthetic minority over-sampling technique, and conditional generative adversarial network. Moreover, it is suitable for different classifiers.

Keywords: power theft detection; data augmentation; conditional variational auto-encoder; convolutional neural network; deep learning

1. Introduction

The electrical loss includes non-technical loss and technical loss. Technical loss is an unavoidable loss in the process of power transmission, which is determined by power loads and parameters of power supply equipment. Non-technical loss is caused by wrong measurement, electricity theft, and non-payment by consumers [1]. In recent years, the U.S. has lost USD 6 billion every year due to electricity theft, according to a report by Forbes magazine [2]. Therefore, the detection of electricity theft is of great significance to reduce non-technical loss.

The existing methods for electricity theft detection can be divided into supervised classification and unsupervised regression. The unsupervised regression method is to determine the electricity theft by comparing the deviation between the actual value and the predicted value of the power load [3]. This kind of method does not need a labeled data set to train the model, but it is difficult to set the threshold and the detection accuracy is low [4,5]. Supervised classification methods mainly include traditional data mining models such as support vector machine (SVM), multi-layer perceptron (MLP), Bayesian network, extreme gradient boosting tree (XGBoost) [6–10], and new deep learning technologies such as the deep belief network and convolutional neural network (CNN) [11–13]. Specifically, SVM is very suitable for binary classification. For n types of stealing power curves, it needs to train n SVM, which consumes a lot of computing time for the data sets with a large number of samples [14,15]. The Bayesian network is sensitive to the form of input data, and it needs to assume a prior distribution for samples, which may lead to poor accuracy for detection due to the inaccurate prior model [16]. MLP has a powerful non-linear mapping ability. In theory, it can fit arbitrary continuous functions theoretically. However, it has the problem of over-fitting [17]. The XGBoost improves the performance by using multiple classifiers, but it has too many parameters, which makes it difficult to adjust parameters [18,19]. In general, these traditional data mining methods are easy to implement, and are suitable for electricity theft detection with small samples. However, they have problems of low feature extraction ability and limited detection accuracy. Relatively, deep neural networks not only have strong ability of feature extraction, but also can map complex nonlinear relationships, which gives them a higher detection accuracy than traditional methods [20,21].

A sufficient number of stealing power curves in the data set is the basis to ensure that the deep neural networks have strong generalization ability. However, it is difficult to detect electricity theft due to the strong concealment of thieves and limited audit resources. In practical engineering, the number of stealing power curves found are limited, which is not enough to train deep neural networks. Therefore, it is necessary to use the limited stealing power curves for data augmentation, so as to improve the accuracy of detection. In reference [22], the random oversampling (ROS) is proposed to reproduce the samples. Although the number of samples is increased, the classifier is prone to over fitting, since new sample lacks diversity. To solve this problem, the synthetic minority over-sampling technique (SMOTE) is proposed in reference [23–25]. However, the SMOTE does not take into account the probability distribution characteristics of electricity stealing curves, so the improvement of accuracy is limited. Reference [26] uses the conditional generative adversarial network (CGAN) to model the stealing power curves, which has higher accuracy than the traditional oversampling methods, but it is difficult to adjust parameters, and the training process is unstable.

The conditional variational auto-encoder (CVAE) is a novel deep generative network which uses output vectors to reconstruct input features. At present, the CVAE has been widely used in different fields such as image augmentation, dimensionality reduction, and data generation [27–29], and has shown good performance, but its application in data augmentation for stealing power curves is still in its infancy. In theory, the CVAE effectively extracts the potential features of stealing power curves by using the encoder with strong learning ability, and reconstructs the stealing power curves by the decoder, which can provide enough data for the deep neural network. Specifically, it is necessary to redesign the structure of the CVAE to make it suitable for generating stealing power curves, since the existing structures are only suitable for processing 2-dimensional data, such as images and videos.

To improve the accuracy of electricity theft detection, a data augmentation method for stealing power curves based on conditional variational auto-encoder is proposed in this paper. The key contributions of this paper can be summarized as follows:

1. The CVAE proposed has strong generalization ability and can generate many stealing power curves similar to that from the test set through unsupervised learning. As long as Gaussian noises are input to the decoder of CVAE, any number of samples of stealing power curves can be generated to train the deep neural network.
2. Compared with ROS and SMOTE, the samples generated by CVAE not only have diversity, but also capture the probability distribution characteristics of stealing power curves. In addition, the training process of CVAE is more stable than that of CGAN and can generate new samples with higher quality.
3. After data augmentation for the training set by CVAE, the detection accuracy of deep neural network can be significantly improved, and it is suitable for different classifiers.

The rest of this paper is organized as follows: Section 2 proposes the conditional variational auto-encoder for data augmentation. Section 3 introduces the process of electricity theft detection base on CNN. The simulation and results are shown in Section 4. The conclusions are described in Section 5.

2. Conditional Variational Auto-Encoder for Data Augmentation

2.1. Conditional Variational Auto-Encoder

Formally, the variational auto-encoder is to learn the data distribution $p_\theta(X)$ of stealing power curves according to the historical data $X = \left\{x^1, x^2, \cdots x^n\right\}$. Typically, this data distribution of stealing power curves can be decomposed as follows [30]:

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

where Π is the capital pi that is a product of all values in range of series.

In order to solve numerical problems, the log function is applied to obtain the following results:

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

Each data point of steal power curves includes the latent variable z that explains the generative process. The Equation (1) can be rewritten for a single point as:

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

where $\int$ denotes the sign for definite integrals.

The generation procedure for stealing power curves includes various steps. First, the prior probability $p_{\theta^*}(z)$ is sampled to obtain the latent variable z. Then, the stealing power curve x is generated accordingly to the posterior probability $p_{\theta^*}(x|z)$. Unfortunately, the prior probability $p_{\theta^*}(z)$ and the posterior probability $p_{\theta^*}(x|z)$ are not available. In order to estimate them, the posterior probability $p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)}$ needs to be known. Hence, the inference is very difficult. Since the posterior probability is often very complex, a simple distribution $q_\phi(z|x)$ and parameter ϕ are needed to approximate it.

The distribution $\log p_\theta(x^i)$ needs to be estimated, because it is impossible to directly sample the distribution of the stealing power curves. Therefore, the Kullback–Leibler divergence can be combined with the variational lower bound:

$$\log p_\theta(x) = D_{KL}\Big(q_\phi(z|x) || p_\theta(z|x)\Big) + L(\theta, \phi; x) \tag{4}$$

$$L(\theta, \phi; x) = \int q_\phi(z|x) \log \frac{p_\phi(x|z)}{q_\phi(z|x)} = E_{q_\phi(z|x)}\Big[-\log q_\phi(z|x) + \log p_\phi(x|z)\Big] \tag{5}$$

where $D_{KL}\Big(q_\phi(z|x) || p_\theta(z|x)\Big)$ is the Kullback–Leibler divergence between the distribution $q_\phi(z|x)$ and the distribution $p_\theta(z|x)$. q_ϕ is the probability distribution to be learned and p_ϕ is the prior distribution of latent variables. Obviously, this Kullback–Leibler divergence is greater than 0. The term acts as a lower bound of the log-likelihood:

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

In this case, the term $L(\theta, \phi; x)$ could be written as:

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

where the first term $-D_{KL}\left(q_\phi(z|x)||p_\theta(z)\right)$ constrains the function $q_\phi(z|x)$ to the shape of the $p_\theta(z)$. The second term $E_{q_\phi(z|x)}[\log p_\theta(x|z)]$ reconstructs the input data with the given latent variable z that follows $p_\theta(x|z)$. With this optimization goal L, the model can be parameterized as follows:

$$\begin{cases} q_\phi(z|x) = q(z; f(x,\phi)) \\ p_\theta(x|z) = p(x; g(z,\theta)) \end{cases} \tag{8}$$

where f and g are deep neural networks with a set of parameters, respectively. A more detailed derivation about VAE can be found in [27].

For the stealing power curve, it may have different shapes due to different attack methods, such as physical attack, data attack, and communication attack. In order to make the variational auto-encoder generate the stealing power curves of the specified attack method, the labels should be added in the training stage of the variational auto-encoder. Normally, the conditional distribution $p_\theta(x|y)$ can be used to replace the original distribution $p_\theta(x)$. The term $L(\theta,\phi;x)$ of CVAE could be written as [31]:

$$L(\theta,\phi;x|y) = -D_{KL}\left(q_\phi(z|(x|y))||p_\theta(z)\right) + E_{q_\phi(z|x)}\left[\log p_\theta((x|y)|z)\right] \tag{9}$$

2.2. Data Augmentation for Stealing Power Curves

The main advantage of CVAE is that it does not need to assume the probability distribution of the stealing power curves, and only a few samples are needed to train the model, which can generate samples similar to the original stealing power curves. A summary of this process for generating stealing power curves is represented in Figure 1.

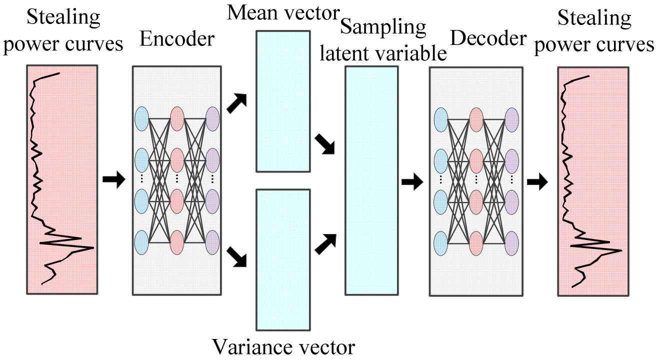

Figure 1. Process of generating stealing power curves by the conditional variational auto-encoder (CVAE).

Step 1: The input data of CVAE are the stealing power curves and labels. Before inputting the data into the CVAE, it is necessary to normalize the data of power stealing curves, otherwise CVAE may not converge. In this paper, the min-max normalization method is used to transform the input data into values between 0 and 1.

Step 2: The deep convolutional network with a strong ability of feature extraction is used to construct encoder that maps input data to the low dimensional latent variables. Then, the mean and variance of the output data of the encoder are calculated, which are used to generate corresponding Gaussian noises as the input data of the decoder.

Step 3: Gaussian noises are fed to the decoder composed of the deep transposed convolutional network to generate new stealing power curves. Then, the output data of the decoder and the actual data are utilized to calculate the loss function, which is used to update the weight of the encoder and decoder by the back-propagation method.

Step 4: After training the CVAE, the Gaussian noises are fed to the decoder to generate the stealing power curves under the specified attack model. Furthermore, the generated stealing power curves and the original samples from the training set will be used to train a classifier (e.g., CNN), which is used to distinguish whether the unknown sample is a stealing power curve or a normal power curve.

3. Electricity Theft Detection Based on Data Augmentation

3.1. Attack Models for Generation of Stealing Power Curves

In previous works, most of the stealing power curves are obtained by simulation, because it is difficult to detect electricity theft due to the strong concealment of thieves and limited audit resources. In this paper, the different attack models (e.g., physical attack, communication attack, and data attack) are utilized to obtain the samples with labels [2,13]. Table 1 shows the stealing power curves under different attack models. In the Table 1, some types of attack models will cause denial of service. In this case, the meter will stop reporting consumer information. This is the case of attack models such as alter routing table, drop packets, and disconnect meter [32]. The more problematic attack models are those that allow generating fake consumption records that imitate a user with a legitimate low power profile. This is the case of session hijacking, other types of attack models that permit privileged access to the firmware misconfiguration and power meter [33,34].

Table 1. Mathematical formulas of different attack models.

Attack Models	Mathematical Model	Attack Models	Mathematical Model
Type 1	$x'_t = \alpha x_t, 0.1 \leq \alpha \leq 0.8$	Type 4	$x'_t = \gamma_t \text{mean}(x), 0.1 \leq \gamma_t \leq 0.8$
Type 2	$x'_t = \beta_t x_t, \beta_t = \begin{cases} 0, t_s < t < t_e \\ 1, \text{else} \end{cases}$ $0 \leq t_s \leq 46 - t_c, t_e = t_s + t_d, t_c \leq t_d \leq 48$	Type 5	$x'_t = x_{48-t}$
Type 3	$x'_t = \gamma_t x_t, 0.1 \leq \gamma_t \leq 0.8$		

In type 1, the normal power curve, is multiplied by a random number in the range of 0.1 to 0.8 to get the stealing power curve. In type 2, the recorders of consumption are replaced by zeroes during a random period of every day. In type 3, every point of the normal power curve is multiplied by a random number in the range of 0.1 to 0.8. In type 4, the recorder of consumption is the product between the mean of the normal power curve and a random noise in the range of 0.1 to 0.8. In type 5, the recorders of consumption between low electrovalence period and high electrovalence period are exchanged.

3.2. Electricity Theft Detection Based on CNN

The input variables of the classifier for electricity theft detection are the power curves, and the output variables are the types of power curves shown in Table 2.

Table 2. The way of one-hot coding for power curves.

Curves	Output Code	Curves	Output Code	Curves	Output Code
Normal	000000	Type 2	001000	Type 4	000010
Type 1	010000	Type 3	000100	Type 5	000001

As one of the representative algorithms of deep learning technologies, CNN has been widely used in image classification, fault diagnosis, and time series prediction due to its powerful feature extraction ability and has achieved remarkable results [35,36]. Compared with the traditional classification methods, CNN can not only map more complex nonlinear relationships, but also has good generalization ability. Therefore, this paper selects CNN as the classifier for electricity theft detection.

CNN is composed of convolutional layers, pooling layers, flatten layers, dropout layers, and fully connected layers. Specifically, convolutional layers and pooling layers are responsible for extracting the features of stealing power curves. Their mathematical formula is as follows:

$$y_i = f_i(x_i * w_i + b_i) \tag{10}$$

$$y' = \max_{i,j\in R}\left(y_{i,j}\right) \tag{11}$$

where x_i denotes input data of i-th convolutional layer and y_i denotes output data of i-th convolutional layer. y' denotes output data of i-th max-pooling layer. f_i is the activation function. b_i and w_i denote the offset vector and the weights of the i-th convolutional layer, respectively.

To alleviate the over-fitting, the dropout layer can make some neurons lose efficacy with a certain probability. The flatten layer is used as the bridge between the pooling layer and fully connected layer, which plays the role of format transformation for data. The mathematical formula of a fully connected layer is as follows:

$$y''_i = g_i\left(y'_i w''_i + b''_i\right) \tag{12}$$

where g_i is the activation function. b''_i and w''_i denote the offset vector and the weights of the i-th fully connected layer, respectively. y''_i and y'_i denote output and input data of i-th fully connected layer, respectively.

According to the characteristics of the power curves, the optimal structure and parameters of CNN are obtained after many experiments, as shown in Figure 2.

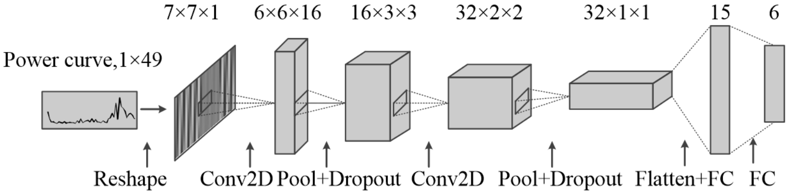

Figure 2. The structure and parameters of the convolutional neural network (CNN) for electricity theft detection.

In order to process the data conveniently, a 0 element is added at the end of the power curve, and the 1 × 49 vector is reconstituted into a 3-dimensional tensor of 5 × 5 scales as the input data of the convolutional layer. Then, two convolutional layers and max-pooling layers are used to extract the key features of the power curves. The number of filters in two convolutional layers is 16 and 32, respectively. The convolutional size and pooling size are 2 × 2. There is a dropout layer behind the pooling layer, which makes neurons lose efficacy with a probability of 0.25. After the flatten layer, there are two fully connected layers with 15 and 6 neurons, respectively. Activation functions are mathematical equations that determine the output of a neural network. The function is attached to each neuron in the network, and determines whether it should be activated or not, based on whether each neuron's input is relevant for the model's prediction. Common activation functions include the Sigmoid function, Tanh function, Softmax, and ReLU. Specifically, the Sigmoid function is usually used to normalize the output of the last layer of the neural network for forecasting tasks. The Softmax function is usually used as a classifier of the neural network for multi-classification. For the Tanh function, previous works show that it has the problems of vanishing gradient and is computationally expensive [11]. Therefore, except for the last layer, it uses softmax function as activation function, and the remaining layers use the ReLU function as the activation function. The loss function is categorical cross-entropy, and the optimizer is the Adadelta algorithm.

3.3. The Process of the Proposed Methods

Summarizing the above analysis, the process of electricity theft detection based on data augmentation is shown in Figure 3. The specific steps are as follows:

Step 1: After importing the dataset, the dataset is divided into the training set, validation set, and test set. The one-hot codes method is used to represent seven types of power curves, and the min-max normalization method is used to normalize the raw data.

Step 2: In the coding stage, the stealing power curves are mapped into latent variables by encoder. In the decoding stage, the new stealing power curves are obtained by feeding Gaussian noises to decoder. Then, the loss function is calculated to update the weights of the network. After the training of CVAE, a large amount of Gaussian noises are fed to the decoder of CVAE to generate new samples for training CNN.

Step 3: The samples generated by CVAE and the original samples from training set are used to train CNN. In the training process, the features of input variables are extracted by convolutional layers and pooling layers, and the labels output by a fully connected layer are used to calculate the loss function. Finally, the back-propagation algorithm is used to update the weights of CNN. After training CNN, it will be used to distinguish whether the unknown sample is a stealing power curve or a normal power curve.

Step 4: For the multi classification problem, it is too simple to evaluate the performance of the model only by accuracy. In this paper, Macro F1 and G-mean are used to evaluate the performance of CNN for the test set [37,38].

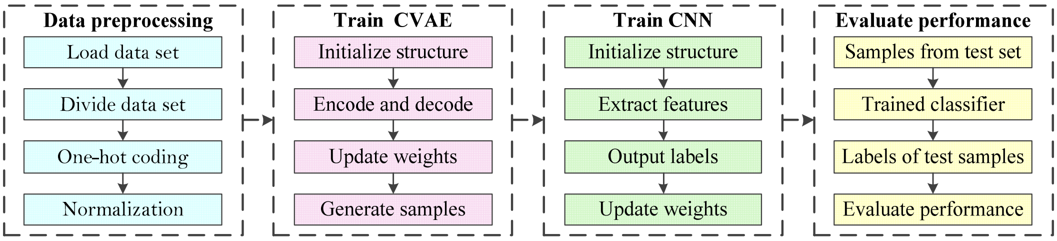

Figure 3. Process of electricity theft detection based on data augmentation.

4. Case Study

4.1. Data Description

To illustrate the effectiveness of the proposed methods, the data set of smart meters from London is used for simulation and analysis [39]. In this dataset, the time resolution of the power curve is 30 min, which means that each power curve has 48 points. Some samples are randomly selected to generate the stealing power curves based on the attack models proposed in Section 3.1. For example, in order to generate the stealing power curves in type 1, a normal curve is randomly selected from the data set. Then, this normal power curve is multiplied by a random number in the range of 0.1 to 0.8 to get the stealing power curve in type 1. In this case, the power curve with a label can be obtained through the attack models. Furthermore, the CVAE model is used to expand the number of training samples to twice the original number as shown in Table 3. Specifically, the samples in the validation set and test set do not change after data augmentation.

Except for SVM, the other algorithms (e.g., CVAE, CNN, MLP, XGBoost, ROS, SMOTE, and GAN) in this paper are running in Spyder (Python 3.7) with keras 2.2.4 and tensorflow 1.12.0. The parameters of the computer are as follows: 16 GB of memory, the processor is 3.8 GHz and Intel Core (TM) i5-7400 CPU.

Table 3. The number of samples in the training set, validation set, and test set.

Samples	Before Data Augmentation				After Data Augmentation			
	Sum	Training	Validation	Test	Sum	Training	Validation	Test
Normal	600	400	100	100	1000	800	100	100
Type 1	300	200	50	50	500	400	50	50
Type 2	300	200	50	50	500	400	50	50
Type 3	300	200	50	50	500	400	50	50
Type 4	300	200	50	50	500	400	50	50
Type 5	300	200	50	50	500	400	50	50

4.2. Performance of CVAE

Figure 4 shows the structure and parameters of CVAE. The input data of CVAE are vectors of 1 × 48 scales. A 0 is added at the end of these vectors and then they will become vectors of 1 × 49 scales. Furthermore, the reshape function is used to transform these vectors into matrixes of 7 × 7 × 1 scales. For the encoder, it includes four convolutional layers, one flattened layer, and three fully connected layers. Specifically, the first convolutional layer includes one filter, and the remaining three convolutional layers have 16 filters. The kernel size of the first two layers is 2, and that of the last two layers is 3. The activation function of all convolution layers is the ReLU function. Every convolutional layer is followed by a batch normalization layer. The flatten layer is used as the bridge between the pooling layer and the fully connected layer, which plays the role of format transformation for data. To calculate the KL divergence loss and sample latent variable, the encoder adds two fully connected layers with 32 neurons for variance and mean to its end. For the decoder, its input data are Gaussian noises of 1 × 32 scales. Two fully connected layers, three deconvolutional layers, and one convolutional layer constitute the decoder. Specifically, the numbers of neurons in the fully connected layers are 64 and 256, respectively. The numbers of filters in the deconvolutional layers are all 16 and the kernel size is all 3. The number of filters in the convolution layer is 1 and the kernel size is 2. The activation functions are all ReLU functions. In addition, the optimizer is the Rmsprop algorithm.

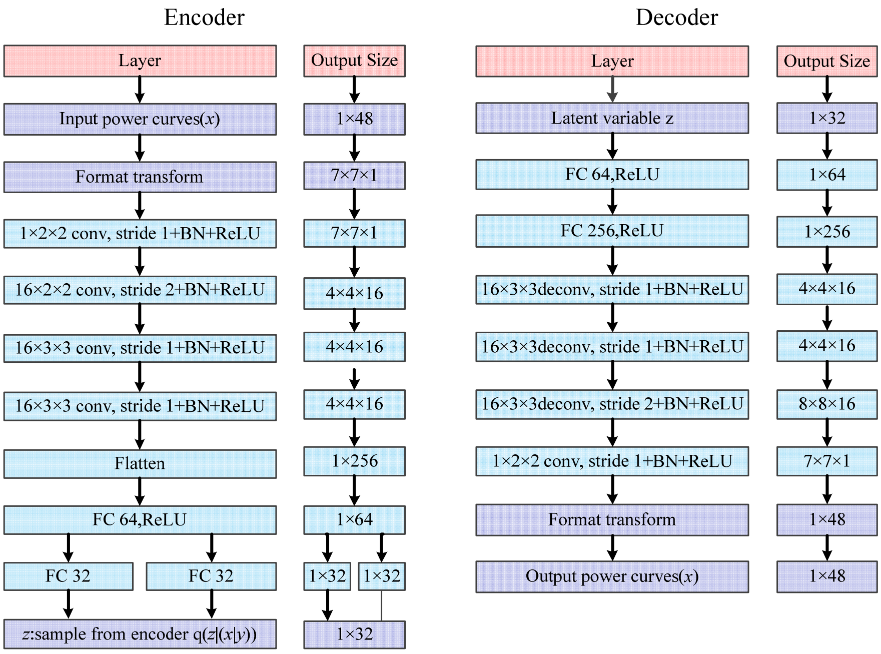

Figure 4. Structure and parameters of CVAE.

In order to observe the training stability of CVAE, Figure 5 visualizes the evolution process of CVAE. Obviously, the loss function decreases rapidly with the increase of iteration times. When the iteration times are more than 40, the value of the loss function tends to be stable, which indicates that CVAE has entered the convergence state. The training process of CVAE is very stable, not like the loss function of CGAN which fluctuates violently and is difficult to converge.

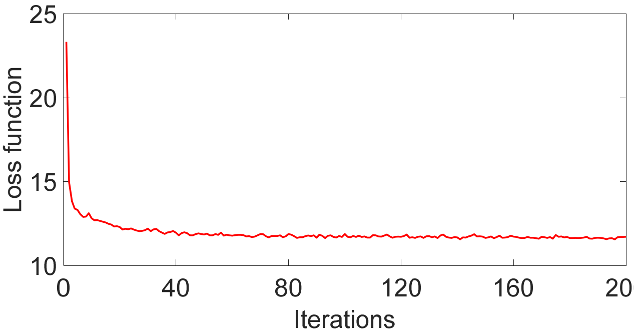

Figure 5. Evolution process of CVAE.

After training CVAE, the Gaussian noises of 1 × 32 scales are used as input variables of the decoder, and a large number of new stealing power curves are obtained. Then, some new stealing power curves are selected to verify the effectiveness of the power curves generated by CVAE. Next, the Euclidean distance of each power curve in the test set and the new power curve generated by CVAE is calculated, and the power curve in the test set with the minimum Euclidean distance is selected. Finally, Figure 6 visualizes the shapes of the generated power curves and the real power curves.

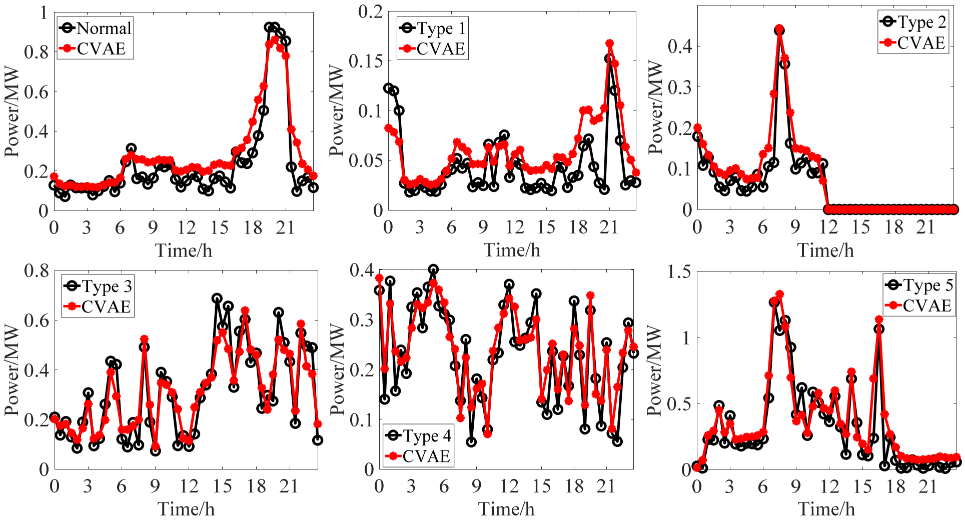

Figure 6. Comparison of shapes between real samples and generated samples.

It can be seen from Figure 6 that the stealing power curves generated by CVAE are very close to those from the test set. The stealing power curves in the test set do not participate in the training process of the CVAE, which indicates that CVAE has a strong generalization ability, and the stealing power curves generated by CVAE are very in line with the actual scene.

In addition to comparing the shape similarity of stealing electricity curves, the validity of CVAE can be further verified by the probability density function (PDF). It can be seen from the Figure 7 that the probability distribution functions of the stealing power curves generated by CVAE are very close to those from the test set, which indicates that CVAE can not only learn the shape of the stealing power curves, but also capture the distribution characteristics of historical data to generate high-quality samples.

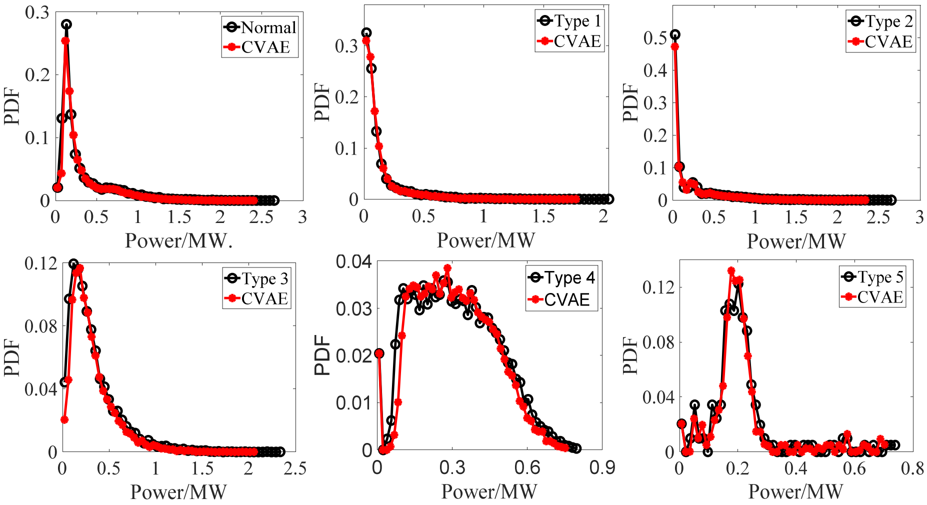

Figure 7. Comparison of probability distribution between real samples and generated samples.

4.3. Performance Comparison of Different Methods for Data Augmentation

In order to illustrate the effectiveness of generating stealing power curves by CVAE, the ROS, SMOTE, and CGAN are used as the baseline. These methods are used to expand the samples of the training set to train a classifier (e.g., CNN), and the results of the classifier for test set are shown in Table 4.

Table 4. Results of the test set under different data augmentation methods.

Methods	Accuracy	Macro F1	G-Mean
No data augmentation	83.25%	83.90%	83.55%
ROS	85.25%	85.54%	85.24%
SMOTE	86.75%	86.80%	86.88%
GAN	88.00%	88.36%	88.23%
CVAE	90.25%	90.55%	90.56%

As can be seen from Table 4, the detection performance of CNN has been significantly improved after data augmentation by various methods. Specifically, after data augmentation by ROS, the accuracy, Macro F1, and G-mean of CNN are improved by 2.00%, 1.64%, and 1.69%, respectively, compared with the original training set. After data augmentation by SOMTE, the accuracy, Macro F1, and G-mean of CNN are improved by 3.50%, 2.90%, and 3.33%, respectively, compared with the original training set. After data augmentation by CGAN, the accuracy, Macro F1, and G-mean of CNN are improved by 4.46%, 4.46%, and 4.68%, respectively compared with the original training set. After data augmentation by CVAE, the accuracy, Macro F1, and G-mean, of CNN are improved by 7.00%, 6.65%, and 6.01%, respectively, compared with the original training set. Therefore, compared with the existing methods for data augmentation, the proposed CVAE can expand the training set according to the actual shape and distribution characteristics of stealing power curves, and has the strongest improvement on CNN performance.

4.4. Adaptability Analysis of CVAE

In order to verify the adaptability of CVAE to different classifiers, CVAE is used to expand the samples from the training set, and then the performance of different classifiers (e.g., CNN, MLP, SVM, and XGBoost) after data augmentation is tested. After many experiments, their optimal parameters are found as follows:

For MLP, the number of neurons in the input layer is 48, and the number of neurons in the middle layer is 24 and 12, respectively. The number of neurons in the output layer is equal to the number of categories. The optimizer is the root mean square prop (RMSprop) and the loss function is cross-entropy. Besides, a dropout layer with a rate of 0.25 is inserted between each fully connected layer to alleviate over-fitting. For SVM, the fitcecoc function from MATLAB2018a is used to classify stealing power curves. For XGBoost, the min child weight is 2 and the subsample is 0.8. The max depth is 6 and eta is 0.1. The gamma is 0.2. The results of the test set using different classifiers as shown in Table 5.

Table 5. Results of the test set using different classifiers.

Classifiers	Before Data Augmentation			After Data Augmentation		
	Accuracy	Macro F1	G-Mean	Accuracy	Macro F1	G-Mean
CNN	83.25%	83.90%	83.55%	90.25%	90.55%	90.56%
MLP	78.75%	78.78%	78.06%	82.50%	83.13%	82.71%
SVM	75.25%	75.64%	74.44%	79.75%	80.04%	78.70%
XGBoost	80.25%	80.75%	80.65%	83.25%	83.48%	82.62%

As can be seen from Table 5, the performance of each classifier has been greatly improved after data augmentation by CVAE. Specifically, the accuracy, Macro F1, and G-mean of CNN are improved by 7.00%, 6.65%, and 7.01%, respectively after data augmentation. The accuracy, Macro F1, and G-mean of MLP are improved by 3.75%, 4.35%, and 4.65%, respectively after data augmentation. The accuracy, Macro F1, and G-mean of SVM are improved by 4.50%, 4.40%, and 4.26%, respectively after data augmentation. The accuracy, Macro F1, and G-mean of XGBoost are improved by 3.00%, 2.73%, and 1.96%, respectively, after data augmentation. In general, CVAE can effectively improve the accuracy of electricity theft detection through the unsupervised generation of new samples, which is suitable for different classifiers.

5. Discussion

The objective of this paper is to propose a new method based on CVAE to improve the accuracy for electricity theft detection. In this paper, the effectiveness of the proposed CVAE has been tested on the smart meter dataset from the low carbon London project. The simulation results show that the accuracy of electricity theft detection can be significantly enhanced after data augmentation by CVAE. For the CVAE model, its training process requires some labeled power curves. However, the labels of the power curves of stealing electricity are difficult to obtain in some cases, which make it impossible to train the CVAE model. At this time, we can try to use the traditional VAE to model different types of stealing power curves. If the data set contains n kinds of different electricity stealing power curves, we have to train n VAE model. Relatively, if each stealing power curve has a label, we only need to train one CVAE model.

6. Conclusions

Due to the strong concealment of electricity theft and the limitation of inspection resources, the number of power theft samples mastered by the power department is insufficient, which limits the accuracy of power theft detection. Therefore, a data augmentation method for electricity theft detection based on a conditional variational auto-encoder is proposed. The following conclusions are drawn through simulation:

(1). The training process of CVAE is very stable, and the convergence speed is fast. The generated stealing power curves have a similar shape and distribution characteristics with the original stealing power curves.

(2). After data augmentation by CVAE, the accuracy, Macro F1, and G-mean of CNN are improved by 7.00%, 6.65%, and 6.01%, respectively compared with the original training set. Compared with existing data augmentation methods (e.g., ROS, SMOTE and GAN), the accuracy, Macro F1, and G-mean values of CNN are the largest, which indicates that the new samples generated by CVAE have the strongest improvement on detection performance.

(3). Compared with the original training set, the training set augmented by CVAE improves the comprehensive detection performance of classifiers such as CNN, MLP, SVM, and XGBoost, which indicates that CVAE is suitable for different classifiers.

For future work, we can try other generative networks (e.g., a flow-based network) to model the stealing power curve. In addition, the capsule network can be used to distinguish the stealing curves from the normal curves.

Author Contributions: Conceptualization, B.T.; Formal analysis, X.G.; Investigation, X.G.; Software, B.T.; Supervision, W.L.; Visualization, R.Z.; Writing—original draft, L.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by The National Natural Science Foundation of China (Grant No. 51667017) and the Key Laboratory of Tibet Department of Education: support project of Electrical Engineering Laboratory of Tibet Agriculture and Animal Husbandry University (Grant No. 2019D-ZN-02).

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

Nomenclature

X	historical stealing power curves
$p_\theta(X)$	data distribution of stealing power curves
z	latent variable
$p_{\theta^*}(z)$	the prior probability
$p_{\theta^*}(x \mid z)$	the posterior probability
$D_{KL}\big(q_\phi(z \mid x) \Vert p_\theta(z \mid x)\big)$	the Kullback–Leibler divergence
q_ϕ	the probability distribution to be learned
p_ϕ	the prior distribution of latent variables
α	a random noise in the range of 0.1 to 0.8
γ_t	a random noise in the range of 0.1 to 0.8 for the t-th point of power curves
x'_t	the t-th point of stealing power curves generated by attack models
x_t	the t-th point of normal power curves
w_i	the weights of the i-th convolutional layer
b_i	the offset vector of the i-th convolutional layer
y'	the output data of i-th max-pooling layer
w''_i	the weights of the i-th fully connected layer
b''_i	the offset vector of the i-th fully connected layer
y''_i	the output data of i-th fully connected layer

References

1. Nabil, M.; Ismail, M.; Mahmoud, M.M.E.A.; Alasmary, W.; Serpedin, E. PPETD: Privacy-Preserving Electricity Theft Detection Scheme with Load Monitoring and Billing for AMI Networks. *IEEE Access* **2019**, *7*, 96334–96348. [CrossRef]
2. Zanetti, M.; Jamhour, E.; Pellenz, M.; Penna, M.; Zambenedetti, V.; Chueiri, I. A Tunable Fraud Detection System for Advanced Metering Infrastructure Using Short-Lived Patterns. *IEEE Trans. Smart Grid* **2019**, *10*, 830–840. [CrossRef]
3. Yip, S.; Tan, C.; Tan, W.; Gan, M.; Bakar, A.A. Energy theft and defective meters detection in AMI using linear regression. In Proceedings of the 2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I & CPS Europe), Milan, Italy, 6–9 June 2017.

4. Wu, R.; Wang, L.; Hu, T. AdaBoost-SVM for Electrical Theft Detection and GRNN for Stealing Time Periods Identification. In Proceedings of the IECON 2018—44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018.
5. Gu, G.; He, Q.; Wang, B.; Dai, B. Comparison of Machine Learning Techniques for the Detection of the Electricity Theft. In Proceedings of the 2018 IEEE 3rd International Conference on Cloud Computing and Internet of Things (CCIOT), Dalian, China, 20–21 October 2018.
6. Jindal, A.; Dua, A.; Kaur, K.; Singh, M.; Kumar, N.; Mishra, S. Decision Tree and SVM-Based Data Analytics for Theft Detection in Smart Grid. *IEEE Trans. Ind. Inform.* **2016**, *12*, 1005–1016. [CrossRef]
7. Punmiya, R.; Choe, S. Energy Theft Detection Using Gradient Boosting Theft Detector with Feature Engineering-Based Preprocessing. *IEEE Trans. Smart Grid* **2019**, *10*, 2326–2329. [CrossRef]
8. Ahmad, T.; Chen, H.; Wang, J.; Guo, Y. Review of various modeling techniques for the detection of electricity theft in smart grid environment. *Renew. Sustain. Energy Rev.* **2018**, *82*, 2916–2933. [CrossRef]
9. Messinis, G.M.; Hatziargyriou, N.D. Review of non-technical loss detection methods. *Electr. Power Syst. Res.* **2018**, *158*, 250–266. [CrossRef]
10. Toma, R.N.; Hasan, M.N.; Nahid, A.; Li, B. Electricity Theft Detection to Reduce Non-Technical Loss using Support Vector Machine in Smart Grid. In Proceedings of the 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 3–5 May 2019.
11. Zheng, Z.; Yang, Y.; Niu, X.; Dai, H.; Zhou, Y. Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids. *IEEE Trans. Ind. Inform.* **2018**, *14*, 1606–1615. [CrossRef]
12. Bhat, R.R.; Trevizan, R.D.; Sengupta, R.; Li, X.; Bretas, A. Identifying Nontechnical Power Loss via Spatial and Temporal Deep Learning. In Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 18–20 December 2016.
13. Wei, L.; Gao, D.; Luo, C. False Data Injection Attacks Detection with Deep Belief Networks in Smart Grid. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an China, 30 November–2 December 2018.
14. Wu, X.; Zuo, W.; Lin, L.; Jia, W.; Zhang, D. F-SVM: Combination of Feature Transformation and SVM Learning via Convex Relaxation. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 5185–5199. [CrossRef]
15. Yu, S.; Li, X.; Zhang, X.; Wang, H. The OCS-SVM: An Objective-Cost-Sensitive SVM with Sample-Based Misclassification Cost Invariance. *IEEE Access* **2019**, *7*, 118931–118942. [CrossRef]
16. Liu, J.; Li, D.; Xu, Y. Collaborative Online Edge Caching with Bayesian Clustering in Wireless Networks. *IEEE Internet Things J.* **2020**, *7*, 1548–1560. [CrossRef]
17. Han, S.; Kong, G.; Choi, S. A Detection Scheme with TMR Estimation Based on Multi-Layer Perceptrons for Bit Patterned Media Recording. *IEEE Trans. Magn.* **2019**, *55*, 1–4. [CrossRef]
18. Jiang, Y.; Tong, G.; Yin, H.; Xiong, N. A Pedestrian Detection Method Based on Genetic Algorithm for Optimize XGBoost Training Parameters. *IEEE Access* **2019**, *7*, 118310–118321. [CrossRef]
19. Gu, X.; Han, Y.; Yu, J. A Novel Lane-Changing Decision Model for Autonomous Vehicles Based on Deep Autoencoder Network and XGBoost. *IEEE Access* **2020**, *8*, 9846–9863. [CrossRef]
20. He, Y.; Mendis, G.J.; Wei, J. Real-Time Detection of False Data Injection Attacks in Smart Grid: A Deep Learning-Based Intelligent Mechanism. *IEEE Trans. Smart Grid* **2017**, *8*, 2505–2516. [CrossRef]
21. Niu, X.; Li, J.; Sun, J.; Tomsovic, K. Dynamic Detection of False Data Injection Attack in Smart Grid using Deep Learning. In Proceedings of the 2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 18–21 February 2019.
22. Pang, Y.; Chen, Z.; Peng, L.; Ma, K.; Zhao, C.; Ji, K. A Signature-Based Assistant Random Oversampling Method for Malware Detection. In Proceedings of the 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), Rotorua, New Zealand, 5–8 August 2019.
23. Pan, T.; Zhao, J.; Wu, W.; Yang, J. Learning imbalanced datasets based on SMOTE and Gaussian distribution. *Inf. Sci.* **2020**, *512*, 1214–1233. [CrossRef]
24. Elreedy, D.; Atiya, A.F. A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance. *Inf. Sci.* **2019**, *505*, 32–64. [CrossRef]
25. Feng, W.; Huang, W.; Bao, W. Imbalanced Hyperspectral Image Classification with an Adaptive Ensemble Method Based on SMOTE and Rotation Forest with Differentiated Sampling Rates. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1879–1883. [CrossRef]

26. Wang, D.; Yang, K. A Data Generation Method for Electricity Theft Detection Using Generative Adversarial Network. *Power Syst. Technol.* **2020**, *44*, 775–782.
27. Du, C.; Chen, B.; Xu, B.; Guo, D.; Liu, H. Factorized discriminative conditional variational auto-encoder for radar HRRP target recognition. *Signal Process.* **2019**, *158*, 176–189. [CrossRef]
28. Pesteie, M.; Abolmaesumi, P.; Rohling, R.N. Adaptive Augmentation of Medical Data Using Independently Conditional Variational Auto-Encoders. *IEEE Trans. Med. Imaging* **2019**, *38*, 2807–2820. [CrossRef]
29. Sadeghi, M.; Leglaive, S.; Alameda-Pineda, X.; Girin, L.; Horaud, R. Audio-Visual Speech Enhancement Using Conditional Variational Auto-Encoders. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 1788–1800. [CrossRef]
30. Pan, Z.; Wang, J.; Liao, W.; Chen, H.; Yuan, D.; Zhu, W.; Fang, X.; Zhu, Z. Data-Driven EV Load Profiles Generation Using a Variational Auto-Encoder. *Energies* **2019**, *12*, 849. [CrossRef]
31. Du, Y.; Xu, J.; Zhen, X.; Cheng, M.; Shao, L. Conditional Variational Image Deraining. *IEEE Trans. Image Process.* **2020**, *29*, 6288–6301. [CrossRef] [PubMed]
32. Liang, G.; Weller, S.R.; Zhao, J.; Luo, F.; Dong, Z.Y. The 2015 Ukraine Blackout: Implications for False Data Injection Attacks. *IEEE Trans. Power Syst.* **2017**, *32*, 3317–3318. [CrossRef]
33. Lin, J.; Yu, W.; Yang, X. On false data injection attack against Multistep Electricity Price in electricity market in smart grid. In Proceedings of the 2013 IEEE Global Communications Conference (GLOBECOM), Atlanta, GA, USA, 9–13 December 2013.
34. Yu, L.; Sun, X.; Sui, T. False-Data Injection Attack in Electricity Generation System Subject to Actuator Saturation: Analysis and Design. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1712–1719. [CrossRef]
35. Xin, R.; Zhang, J.; Shao, Y. Complex network classification with convolutional neural network. *Tsinghua Sci. Technol.* **2020**, *25*, 447–457. [CrossRef]
36. Zhou, D.-X. Theory of deep convolutional neural networks: Downsampling. *Neural Netw.* **2020**, *124*, 319–327. [CrossRef]
37. Kerschke, P.; Hoos, H.H.; Neumann, F.; Trautmann, H. Automated Algorithm Selection: Survey and Perspectives. *Evol. Comput.* **2019**, *27*, 3–45. [CrossRef] [PubMed]
38. Qin, Z.; Hu, L.; Zhang, N.; Chen, D.; Zhang, K.; Qin, Z.; Choo, K.R. Learning-Aided User Identification Using Smartphone Sensors for Smart Homes. *IEEE Internet Things J.* **2019**, *6*, 7760–7772. [CrossRef]
39. UK Power Networks. Low Carbon London Project. Available online: https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households (accessed on 10 August 2020).