Solar Panel Identification via Deep Semi-Supervised Learning and Deep One-Class Classification

Elizabeth Cook, *Member, IEEE,* Shuman Luo, *Student Member, IEEE,* Yang Weng, *Member, IEEE*

***Abstract*—As residential photovoltaic (PV) system installations continue to increase rapidly, utilities need to identify the locations of these new components to manage the unconventional two-way power flow and maintain sustainable management of distribution grids. But, historical records are unreliable and constant re-assessment of active residential PV locations is resource intensive. To resolve these issues, we propose to model the solar detection problem in a machine learning set up based on labeled data, e.g., supervised learning. However, the challenge for most utilities is limited labels or labels on only one type of users. Therefore, we design new semi-supervised learning and one-class classification methods based on autoencoders, which greatly improve the nonlinear data representation of human behavior and solar behavior. The proposed methods have been tested and validated not only on synthetic data based on a publicly available data set, but also on real-world data from utility partners. The numerical results show robust detection accuracy, laying down the foundation for managing distributed energy resources in distribution grids.**

***Index Terms*—solar panels, locations, detection, autoencoder, semi-supervised learning, one-class classification**

I. Introduction

With the increase in installations of residential photovoltaic (PV) systems, it is important for utilities to gain visibility of solar panels [1], [2]. Residential PV systems not only create sustainable electricity for their owners, but also build represent a new type of assets for utilities. To better evaluate the benefits and potential revenues associated with these new assets, utilities need to identify the locations of these new components to manage the unconventional two-way power flow and maintain sustainable management of distribution grids. For example, detecting and monitoring all active PV installations in a utility's territory allows the utility to perform accurate hosting capacity analysis (HCA). HCA allows utilities to determine the amount of additional distributed energy resources (DERs) that can be "hosted" on the distribution system at a given time and location, without threatening grid safety, reliability, or power quality [3].

Unfortunately, we cannot determine whether a customer has solar panels with certainty as new installations will go up and some may be retired as time passes by. Even worse, some solar panel installations took place without utility permission [4]. While a utility can manually update historical records on active solar locations, it is cost intensive and difficult to ensure the solar location data are accurate all the time. Without utility visibility of residential PV electricity generation, the system operation is prone to over-voltage and back-feeding through substations. These events can damage system equipment such as transformers, voltage regulators, and customers' appliances. Therefore, utilities are in urgent need of new methods for providing real-time renewable location data to better plan infrastructure and grid operation.

In the past, DER analyses required manual validation of locational information of PV [1], [5]–[7]. As manual checks are not scalable, automation of the localization process is an active area of research. For example, [8]–[10] propose to use an unmanned aerial vehicle (UAV) with different cameras, such as HD cameras, thermal cameras, and infrared cameras to localize different panels and their conditions for fault detection and maintenance. Although these methods are typically successful for detecting large PV arrays (i.e. solar farms), it is challenging to send UAV across different utility service areas, which can be geographically large. Therefore, instead of the UAV approach, [11] and [12] propose to use satellite images to detect solar panels. However, satellite images include many areas without PV systems and there are similar objects that can be incorrectly identified as solar panels. Even worse, such a satellite-based approach cannot distinguish active and non-operational PV installations. The use of smart meter data for solar detection may overcome the obstacles posed by UAV and satellite-based methods. For example, [13] aims to detect the solar panels behind the meter data. The paper proposes a change-point detection algorithm to screen out abnormal usage data. However, change-point detection can identify changes that are not due to solar behaviors.

One key drawback of change-point detection is due to its unsupervised nature and simplicity of using any change-point. While we demonstrate in this paper that supervised learning can achieve satisfactory performance, such learning requires adequate labels of the inputs and outputs [14]. This is insufficient because a utility may not be able to afford the cost and time for obtaining and maintaining a lot of the labels for solar and non-solar users [15], [16]. Therefore, we propose to use semi-supervised learning (SSL) by only requiring a small sample of the labeled data from both classes [17], [18]. When the utility only has labels on one class, e.g., non-solar users, we propose to use one-class classification (OCC) [19], [20].

During the implementation, the direct application of SSL and OCC have relatively low accuracy, as the power system has a high dimensionality in data. For example, each user represents one point in the classification problem, but the user data is the result of vectorizing a long time-series data that can last several days for a clear pattern [21]–[23]. Besides, as residential customers have diversified user behaviors, the data of each class lives on a highly non-linear surface [17].

For resolving the issue of dimensionality, there are mainly three types of methods available. The first type is linear mapping methods, the most typical method of this type is principal component analysis (PCA). The second type is nonlinear mapping methods, the mainstream of this type of method is based on manifold learning. The most basic method among them is multidimensional scaling (MDS), which tries to preserve the original relative distance between the data points

in the lower dimensions. Locally linear embedding (LLE) as another main dimensional reduction method under manifold learning, uses local linearity to reflect global nonlinearity and preserves the data topology structure in the original space. The last type is advanced methods based on neural networks. The most well-known one is the autoencoder (AE).

Considering that PCA only looks for the principal components and may lose the separability information in overlooked projection directions. Also, MDS assumes an equal contribution of all dimensions towards the dimension reduction result and may overlook the fact that some dimensions may be more important than others. Additionally, MDS suffers from high computation cost $O(n^3)$ [24], where n is the number of samples. The LLE method has a low computation cost but is sensitive to the selected neighborhood. An autoencoder does not have these limitations as it uses the data itself to supervise the mapping to lower dimensions. Therefore, we propose to solve the issue of dimensionality and nonlinear representation together by designing new SLL and OCC methods based on autoencoders. Constructed by the two deep neural networks of an encoder and a decoder, an autoencoder is capable of providing a universal approximation of nonlinear and low dimensional space while de-noising [25]–[28].

Finally, we use the known public and utility solar data arrays to validate the proposed methods. We use both accuracy and $F1$ score to measure the performance against baseline results. The baseline results were based on common SSL and OCC methods as well as including common supervised learning methods. Such an experiment shows enhanced solar usage detection when compared to the traditional methods. In summary, the contributions of the paper are:

1) The paper explains why solar detection is urgently needed and why the problem is challenging and cost intensive in reality based on our data mining of realistic utility data.
2) The paper models the solar detection problem in supervised learning, semi-supervised learning (SSL), and one class-classification (OCC) setups. Future researchers can develop relevant tools based on our problem modeling.
3) The paper proposes new SSL and OCC methods based on autoencoders, greatly boosting the power of data representation and model learning.
4) The paper not only validates the methods based on the publicly available synthetic data set, but also has great success on real utility data.

The rest of the paper is structured as follows. Section II shows the feasibility of solar detection via data mining. Section III formulates the solar panel detection problem with limited labels. Section IV and Section V show the enhanced SSL and OCC via autoencoder. Section VI provides numerical results, and Section VII concludes the paper.

II. Differences between Solar + Non-solar Users

The problem of determining whether there are solar panels that are generating power in a residence via utility data is not widely analyzed. The key concern is that solar users and non-solar users are difficult to differentiate. For example, it is difficult to determine whether solar exists behind a meter if the solar generation is small relative to the household usage.

A. Proof of Feasibility with Realistic Data

To validate this difficulty and illustrate the the feasibility of differentiation, we conduct data mining over realistic usage data from our partner utility with 600,000 meters from a major U.S. city. The one-hour interval usage data recorded between June 1st, 2019 to June 30th, 2019 by the billing meters was used for this exercise. We examined the label for the solar users by verifying that the net-metering data of the customers shows power injection and the database shows that the customer has passed the solar panel application. Similar procedures are adapted to examine the label for the customers without solar panels. From the data set with partial manually verified labels, we randomly sample 2,000 usage data with labels indicating functioning solar panels. We also sample another 2,000 usage data with labels indicating no solar panels. Combining them, we conduct supervised learning using classifiers such as logistic regression, support vector machine, k-nearest neighbor, and random forest. All the supervised learning classifiers report above 90% accuracy, which means that the data can be separable.

To motivate if the data are separable, we use principal component analysis (PCA) tool to visualize the magnitude of eigenvalues of our data in Fig. 1. As the y-axis is a logarithmic scale, we can see that only the first few eigenvectors matter and most of the eigenvectors are noises. To illustrate further, we map the data into 2-D and 3-D space in Fig. 2. As we know who are the solar users and who are not in this example, we color the PCA results for visual inspection. The goal is to gain more knowledge about the data density and the possible shape of the boundary for separability. From the figure, we can see that the data can be separated in this case.

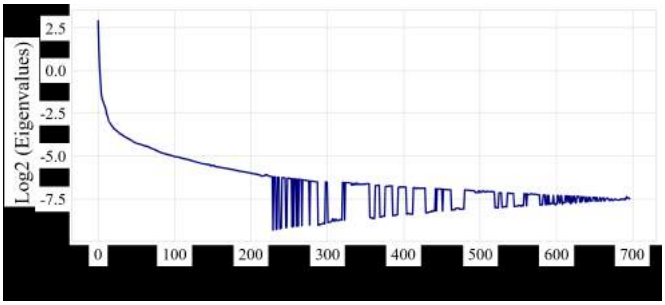


Fig. 1: Results from a popular principal component analysis tool to visualize the magnitude of our data's eigenvalues.

However, it is important to point out that this example is only for the motivation purpose. In reality, there can be serious overlap, calling for methods that can handle highly nonlinear boundaries. We will discuss this later in the paper.

B. Proof of Feasibility with Synthetic Data

As the data is sourced from one specific utility, we also conduct a constructive test to see how robust this differentiation capability is seen by controlling the noise levels. The motivation of controlling the noise is becasue real data shows high variability during the days and during different days with various weather conditions. For example, Fig. 3 comes

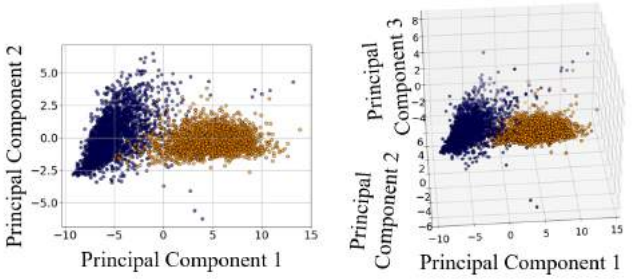


Fig. 2: Visualizations of the principal components showing a boundary between the two different behaviors allowing the data to be separable.

from a utility for solar generation of residential customers. From the data, we can see that solar panels always generate power from sunrise to sunset regardless of cloud cover, but the cloud coverage create intermittent patterns like noises, changing signal shapes. Therefore, we use different noise levels to mimic randomness in residential customers and the environment.

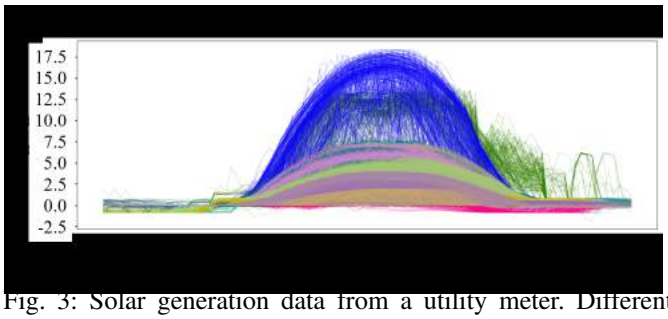


Fig. 3: Solar generation data from a utility meter. Different colors represent different customers, different lines of the same customers represent different days.

Specifically, we use square waves and sinusoidal waves to represent two signal patterns. The motivation comes from the appliances behavior in a home [29], [30] and also the curved shape of solar generation in the day time. With the two signals, we add noises. The data with different noise levels will be directly fed into typical classifiers such as support vector machine (SVM) and logistic regression to determine if accuracy can be preserved with different noise levels. For example, Fig. 4 presents an example of the data set with noise levels increasing from top to bottom. The x-axis shows the time indices, which are 29 days (to be consistent with the real data that we will demonstrate later on) with a one-hour time interval and the y-axis shows the normalized data. Although it becomes more difficult for us to determine the class of the data, Table I shows that the classification results are still high when the noise level is much higher than the signal level.

Classification	SVM (Linear Kernel)	Logistic Regression	Noise Level
Training/Test Acc	100%	100%	$\mathcal{N}(0,1.0)$
Training/Test Acc	100%	100%	$\mathcal{N}(0,4.0)$
Training/Test Acc	100%	100%	$\mathcal{N}(0,7.0)$
Training/Test Acc	100%, 99%	100%, 99.33%	$\mathcal{N}(0,12.0)$

TABLE I: Classification accuracy (acc) of different noise levels, which is normalized with signal level.

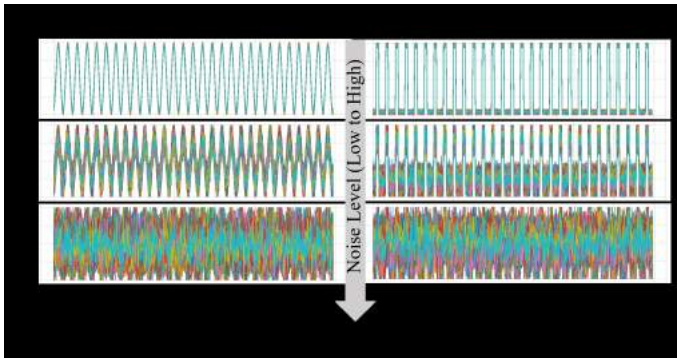


Fig. 4: Data set with different noise level increased from top to the bottom.

III. Problem Definition

The last section shows the feasibility based on rough visualization and supervised learning of abundant but synthetic data. However, the reality at utilities is that the knowledge of highly accurate labels, solar users and non-solar users can be quite limited. In some utilities, there may be only one class of labels with limited resources to manually label more. Therefore, we define the following two problems based on the scarcity of labels in a data set.

A. Semi-Supervised Learning (SSL) Problem

- Problem: Identify the customers who have functioning solar panels out of a large group of customers using smart meter data and a small amount of labels.
- Given:
  - Labeled electricity usage data: $(X_m, \mathbf{y}_m) = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$, where $\mathbf{x}_i$ is the time series smart meter data for a customer, y_i is the label showing whether the customer has solar panels or not, and m is the number of the meter data with labels.
  - Unlabeled electricity usage data: $X_n = \{\mathbf{x}_j\}_{j=m+1}^{m+n}$, where $\mathbf{x}_j$ is the time series smart meter data for a customer and n is the number of meter data without labels, usually $n \gg m$.
- Goal:
  - Find the optimal mapping rule of f_{SLL} so that we can obtain an accurate prediction of whether a customer has solar panels or not by using $\hat{\mathbf{y}}_{SLL} = f^*_{SLL}(\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}, \{\mathbf{x}_j\}_{j=m+1}^{m+n})$.

B. One-Class Classification (OCC) Problem

- Problem: Identify the customers who have functioning solar panels out of a large group of customers using smart meter data and one type of labels.
- Given:
  - Electricity usage data from a known class: $X_p = \{\mathbf{x}_i\}_{i=1}^{p}$, where $\mathbf{x}_i$ is the time series smart meter data for a customer. All the meter data belonging to the same class are assigned with indicators $\mathbf{y}_p = \{y_i\}_{i=1}^{p} = +\mathbf{1}$, where p is the number of meter data and $\mathbf{1}$ is a vector whose elements are all equal to 1.

  - Electricity usage data from other unknown classes: $X_q = \{\mathbf{x}_i\}_{i=p+1}^{p+q}$, where $\mathbf{x}_i$ is the time series smart meter data for a customer. All the meter data from other unknown classes are assigned with indicators $\mathbf{y}_q = \{y_i\}_{i=p+1}^{p+q} = -\mathbf{1}$, where q is the number of meter data.
- Goal:
  - Find the optimal mapping rule of f_{OCC} so that we can obtain an accurate prediction of whether a customer has solar panels or not by using $\hat{\mathbf{y}}_{OCC} = f^*_{OCC}(\{(\mathbf{x}_i, y_i)\}_{i=1}^{p}, \{\mathbf{x}_i\}_{i=p+1}^{p+q})$.

IV. Deep Semi-Supervised Learning

One of the major issues of directly using SSL method from the computer science domain is due to the high dimensionality of power data and the need for nonlinear representation. Therefore, we propose to integrate the autoencoder (AE) into the proposed deep SSL method, where we show the expectation-maximization (EM) algorithm below so that we can properly illustrate the AE part afterwards.

A. Conventional Semi-Supervised Learning Method

EM algorithm relies on mixture models and is a popular way to solve SSL problems and the methods have lots of successful applications in different fields, such as image processing and data classification tasks [31]–[33]. As defined in Section III-A, $(X_m, \mathbf{y}_m) = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ denote the electricity usage data and their correlated labels, $X_n = \{\mathbf{x}_j\}_{j=m+1}^{m+n}$ denote electricity usage data without labels. In our model, labels are assigned binary values (0 or 1), labels with a value of 0 represent the customers who do not have solar panels and labels with a value of 1 represent the customers who have solar panels. Based on this premise, we assume we know the labels $\hat{\mathbf{y}}_{SLL} = \{y_j\}_{j=m+1}^{m+n}$ and we can compute the likelihood of all the data with respect to the underlying parameters Θ, to be shown in Equation (1).

$$P(X_m, \mathbf{y}_m, X_n, \hat{\mathbf{y}}_{SLL}|\Theta) = \prod_{i=1}^{m} P(\mathbf{x}_i, y_i|\Theta) \prod_{j=m+1}^{m+n} P(\mathbf{x}_j, y_j|\Theta) \quad (1)$$

The EM algorithm iteratively fixes the value of Θ and $\hat{\mathbf{y}}_{SLL}$ to find a suboptimal solution of the maximization of the log-likelihood function over all the data. Specifically, for the t^{th} iteration and in the expectation (E) step, Θ^t is fixed and the EM algorithm optimizes a lower bound given by the expected log-likelihood $Q(\Theta|\Theta^t)$ in Equation (2).

$$Q(\Theta|\Theta^t) = \mathbf{E}_{\hat{\mathbf{y}}_{SLL}|X_m, \mathbf{y}_m, X_n, \Theta^t}[\log P(X_m, \mathbf{y}_m, X_n, \hat{\mathbf{y}}_{SLL}|\Theta)] \quad (2)$$

In the maximization (M) step, the algorithm maximizes $Q(\Theta|\Theta^t)$ with respect to Θ given in Equation (3). Although the parameters Θ may be highly correlated, the above procedure faces high computational cost as Θ has high dimensionalities [21].

$$\Theta^{(t+1)} = arg\max_{\Theta} Q(\Theta|\Theta^t) \quad (3)$$

B. Autoencoder (AE) in a SSL Setup

The electricity usage data in the high dimensional space not only exhibit a high level of noise, but also have highly nonlinear user behaviors. In order to reduce the dimension of the data while preserving the nonlinear relationship of the features, we propose to use AE. An AE [25]–[28] constitutes an encoder that compresses the original data to a code and then a decoder which reconstructs the data from the code, as shown in Fig. 5. The encoder can be used to reduce the dimension of the data, help the similarity calculation, and extract the most representative information.

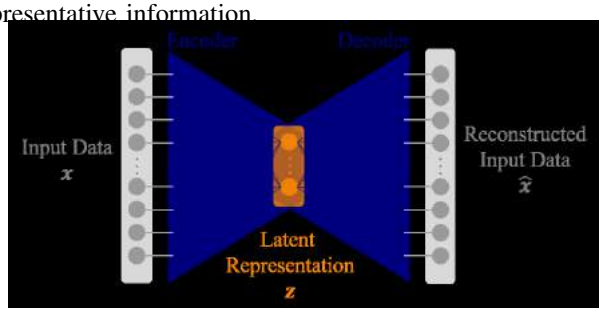


Fig. 5: Block diagram of an AE which constitutes an encoder that compresses the original data to a code and then a decoder which reconstructs the data from the code.

We take the labeled data X_m as an example to explain how the AE is used in our problem setup. An AE uses data themselves to supervise the learning. In our problem, we employ the AE shown in Fig. 6. The input data X_m is the time series smart meter data and will be nonlinearly mapped to a lower dimensional space. The transformed meter data Z_m is a nonlinear combination of the original meter data at each time index. The transformed meter data Z_m will be mapped back to the original space to reconstruct the input meter data. The AE attempts to minimize the error between the input data X_m and the reconstructed input $\hat{X}_m$, defined in Equation (4), to find the optimal representation Z_m of the input data in the low dimensional space. The same procedure will be used to obtain the hidden representation Z_n of the smart meter data without labels X_n.

$$\begin{aligned} Z_m &= f_e(W_e X_m + \mathbf{b}_e), \\ L(X_m, \hat{X}_m) &= \|X_m - \hat{X}_m\|^2 = \|X_m - f_d(W_d Z_m + \mathbf{b}_d)\|^2, \end{aligned} \quad (4)$$

where W_e is the weight matrix between the input data X_m and the latent representation Z_m, W_d is the weights matrix between the hidden representation Z_m and output $\hat{X}_m$. f_e and f_d are the activation functions, $\mathbf{b}_e$ is the bias vector of the encoder, and $\mathbf{b}_d$ is the bias vector of the decoder.

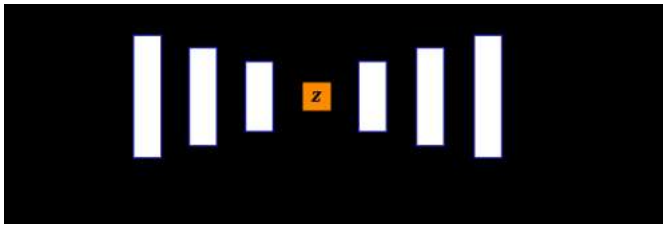


Fig. 6: An example of AE for power data.

Z_m with its associated labels $\mathbf{y}_m$ is fed into a Gaussian mixture model for EM. When EM iteratively finds the solution

of maximizing the log-likelihood function, the labels of the unlabeled data are produced. The complete structure is shown in Fig. 7.

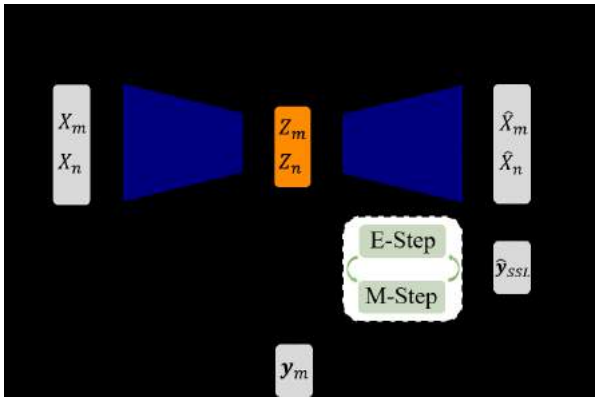


Fig. 7: Block diagram of the proposed deep semi-supervised EM approach.

C. Steps of the Proposed Algorithm

Let the representation $Z_m = \{\mathbf{z_i}\}_{i=1}^{m}$ coming from the AE be the hidden representations of the labeled data whose labels are $\mathbf{y}_m = \{y_i\}_{i=1}^{m}$. Let the representation $Z_n = \{\mathbf{z_j}\}_{j=m+1}^{m+n}$ coming from the AE be the hidden representations of the unlabeled data whose estimated labels are $\hat{\mathbf{y}}_{SSL} = \{y_j\}_{j=m+1}^{m+n}$. We will assume that labels can only take binary values (0 or 1). Based on this setting, suppose we know the labels $\hat{\mathbf{y}}_{SSL}$, we are able to compute the likelihood of the whole data set with respect to the underlying parameters Θ given in Equation (5).

$$P(Z_m, \mathbf{y_m}, Z_n, \hat{\mathbf{y}}_{SLL}|\Theta) = \prod_{i=1}^{m} P(\mathbf{z}_i, y_i|\Theta) \prod_{j=m+1}^{m+n} P(\mathbf{z}_j, y_j|\Theta) \quad (5)$$

For the t^{th} iteration and in the expectation (E) step, Θ^t is fixed and the EM algorithm optimizes a lower bound given by the expected log-likelihood given in Equation (6). In the maximization (M) step, the algorithm maximizes $Q(\Theta|\Theta^t)$ with respect to Θ.

$$\begin{aligned} Q(\Theta|\Theta^t) &= \mathbf{E}_{\hat{\mathbf{y}}_{SLL}|Z_m,\mathbf{y_m},Z_n,\Theta^t}[\log P(Z_m, \mathbf{y_m}, Z_n, \hat{\mathbf{y}}_{SLL}|\Theta)] \\ &= \sum_{\hat{\mathbf{y}}_{SLL}} P(\hat{\mathbf{y}}_{SLL}|Z_m, \mathbf{y_m}, Z_n, \Theta^t) \log P(Z_m, \mathbf{y_m}, Z_n, \\ &\quad \hat{\mathbf{y}}_{SLL}|\Theta) \\ &= \sum_{i=1}^{m} \log P(y_i, \mathbf{z}_i|\Theta) + \sum_{j=m+1}^{m+n} \sum_{y_j \in \{0,1\}} P(y_j|\mathbf{z}_j, \Theta^t) \\ &\quad \log P(y_j, \mathbf{z}_j|\Theta) \\ &= \sum_{i=1}^{m} \log P(y_i, \mathbf{z}_i|\Theta) + \sum_{j=m+1}^{m+n} \sum_{y_j \in \{0,1\}} r_{y_j}^{j} \log P(y_j, \\ &\quad \mathbf{z}_j|\Theta) \end{aligned} \quad (6)$$

In the last line of the equation, we define $r_0^j = P(y_j = 0|\mathbf{z}_j, \Theta^t)$, $r_1^j = P(y_j = 1|\mathbf{z}_j, \Theta^t)$, which are our current estimates for the probabilities of each of the labels in the unlabeled examples. Therefore, in the E step, we compute probabilities r_0^j and r_1^j for all the unlabeled data based on the current Θ^t. In the M step, we maximize the expected log-likelihood (the last term of Equation (6)) for all the data.

V. Deep One-Class Classification

When the labeled data are so limited at a utility that only one class of the labels can be obtained, e.g., only the labels of some non-solar users. In such a case, it is impossible to create a classification boundary between two classes like SSL.

A. Conventional One-Class Classification (OCC) Method

Therefore, one-class classification aims to regularize the descriptive loss, popular in supervised learning and SSL, with an additional loss on compactness. This method aims to evaluate the compactness of data with known labels and with nearby data to form a group, while looking for distinct boundaries that can separate the data into two or more groups. Support vector data description (SVDD) utilized in our paper is one of the OCC solvers. SVDD attempts to define the compactness of the targeted class by constructing a hypersphere with center $\mathbf{c}$ and radius $r > 0$, wrapped in a compactness matrix. The hypersphere gathers as many observations from one class as possible in the feature space with the help of the kernel function ϕ_k [34]. For example, if we have a group of smart meter data which customers using their solar panels have been verified by human effort, we can try to construct a hypersphere that gathers as many data from the group as possible. By minimizing the radius of the hypersphere, we obtain the optimal boundary to separate this group of people from others. The primal problem of SVDD is defined in Equation (7).

$$\begin{aligned} \min_{r,\mathbf{c},\zeta_i} \quad & r^2 + \frac{1}{\nu n}\sum_i \xi_i \\ \text{s.t.} \quad & \|\phi_k(\mathbf{x}_i) - \mathbf{c}\|^2 \le r^2 + \xi_i, \ \xi_i \ge 0, \quad \forall i, \end{aligned} \quad (7)$$

where $\mathbf{x}_i$ is the smart meter data from a known class, the slack variable ξ_i is introduced to allow a soft margin, and the regularization parameter ν controls the relative importance of the volume of the sphere and the penalties ξ_i.

The descriptiveness of the data is maintained in the constraints. Solving the minimization problem given in Equation (7) by using Lagrange multipliers, we can derive that the center $\mathbf{c}$ of the sphere should be a linear combination of some important input data. These input data have a significant influence on the construction of the sphere by describing the boundary of the sphere and are called support vectors.

B. Proposed Deep OCC Method

SVDD often has poor computational efficiency and scalability due to the structure and manipulation of the matrices and SVDD is prone to failure when the data set is extremely large and the dimension of the data is extremely high. Thus, substantial feature engineering is needed [35]. This makes it challenging to capture diversified nonlinear user behavior and remove noise from power data in high dimensions.

Therefore, we propose to use the hidden layers of autoencoder (AE) to extract the nonlinear features for one-class classification. For example, Fig. 8 provides a visual representation of AE's ability to represent highly nonlinear customer data in a low dimensional space for our utility data set. The top left figure is the non-solar data plotted in a 3-D plane, whereas the bottom left figure is the solar data. The two middle plots, top and bottom, are the representation of

the solar and non-solar data, respectively, reconstructed using principal component analysis (PCA). And the right top and bottom figures are the data set reconstructed using an AE and plotted in a 3-D plane. The figure is to provide clarity on the ability of the AE to retain the information more accurately than the PCA. As shown the PCA is not able to reconstruct the data as well as the autoencoder (AE) therefore providing evidence of the high accuracy and advantage of using an AE over a PCA to reconstruct the high-dimensional data for purposes of distinguishing solar and non-solar data. The AE can map the original data to a denser area which helps to construct the compactness description of the targeted class. This enhances the design of the OCC. Hence, the AE will be used in the design for the newly proposed method.

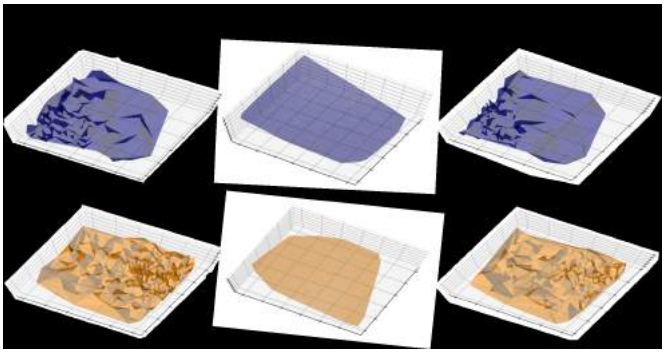


Fig. 8: Illustration comparing PCA reconstruction versus an autoencoder for non-solar (blue) and solar (orange) data set.

The architecture is shown in Fig. 9, where the extracted learned hidden features Z_p for labeled data and Z_q for unlabeled data are fed into the SVDD. Combining the extracted learned hidden features with their labels, the SVDD is able to determine the labels of the unlabeled data. The objective of the problem is to solve Equation (7) after replacing $\mathbf{x}_i$ with $\mathbf{z}_i$.

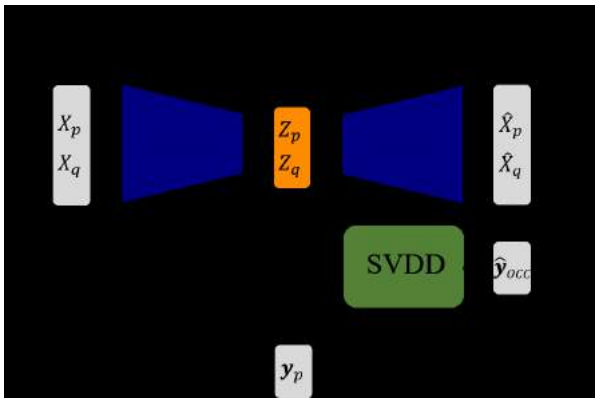


Fig. 9: Block diagram of the proposed deep SVDD approach.

VI. Numerical Validation

With the proposed methods in the last two sections, we will validate the performance in this section. The algorithms used are the deep semi-supervised expectation-maximization (Deep-EM) algorithm and deep support vector data description (Deep-SVDD). We use both public data sets and the utility data sets to conduct our experiments with traditional common semi-supervised learning and one-class classification algorithms. Principal component analysis is also used when necessary for consistency. As a baseline to our result, we also include the results of supervised learning in our experiments with accurate labeled data sets.

A. Data Preparation

The public *UMass Smart** data set [36] used in this study contains everyday electricity load profiles, extracted from the dataset named "Apartment dataset", from 114 single-family apartments from June 1st, 2015 to June 30th, 2015 with a 15-minute interval between each pair of readings. We take the average of the data to scale the original data to a one-hour interval. Therefore, the total number of time indices used in the study is 696, corresponding to 29 days. The solar generation data comes from another dataset named "Solar panel dataset" in the same public data repository, which documents the solar generation data for 50 rooftop solar panels with a one-minute interval between each pair of readings. We select 39 solar generation profiles as the other profiles contained bad data such as near-zero values. Then, we combine them with the aforementioned 114 load profiles to create the electricity usage data set corresponding to solar users. To mimic the unbalanced data set, we add a number of different noises to the 114 load profiles to create the profiles for non-solar customers for diversity, when compared to 39 solar customers. For example, as the results are similar, we show the case when we add four different noises to the 114 load profiles, leading to 456 non-solar profiles.

The utility data set used in this study corresponds to a set of everyday electricity usage readings from approximately $600,000$ meters from a U.S. city from June 1st, 2019 to June 30th, 2019 with a one-hour interval between each reading. The total number of time indices used in the study is 696, corresponding to 29 days. Around $1,973$ customers have installed solar panels. Their smart meter readings come from the net meters, which record the household electricity consumption and the PV generation as a whole. The rest of the approximately $598,000$ customers we assume never reported their installations of the solar panels, and therefore, we label them as non-solar. We then randomly select $20,000$ from this data set to conduct this study.

To eliminate the influence of different scales of the data, we use min-max normalization methods to scale the data between 0 and 1 throughout the paper.

B. Performance Metrics

To evaluate binary classification several statistical rates are available to measure performance (i.e., accuracy, $F1$, recall, or precision). For this work, we use the accuracy and $F1$ score as our performance measurements. Accuracy is used when the true positives (TP) and true negatives (TN) are important and the data set's class distribution is similar. $F1$ score is used when the False Negatives (FN) and False Positives (FP) are critical and the data set is unbalanced. These metrics are defined as follows:

$$\begin{aligned} Accuracy &= \frac{TP+TN}{TP+FN+TN+FP}, \\ Precision &= \frac{TP}{TP+FP}, \ Recall = \frac{TP}{TP+FN}, \\ F1 &= \frac{2\times Precision \times Recall}{Precision + Recall}. \end{aligned} \tag{8}$$

We use the $F1$ score since our data set will most likely have an imbalanced class. This will take the precision and recall rate into account which cares for both the majority class and the minority class [37]. We include the accuracy performance metric to observe considering that the synthetic data may not always be imbalanced and therefore should be available to observe any differences.

C. Performance Comparison of Autoencoder and Locally Linear Embedding

In the introduction, we claimed that the autoencoder (AE) has its advantages over other nonlinear decomposition methods such as multidimensional scaling (MDS) and locally linear embedding (LLE) methods. Considering the large computation time of MDS, and to confirm our claim, we provide the results of the performance comparison of the AE and LLE methods. We use the utility data set, which contains 1,973 customers with solar panels and 20,000 customers without solar panels, to conduct the analysis. The results are shown in Fig. 10. As can be seen from the figure, for semi-supervised learning, the LLE has comparable performance with AE in low dimensions, however, its performance suddenly crushes down after projecting to dimensions higher than 12. In terms of one class classification, the LLE has comparable performance with AE in some dimensions, however, it also experience tremendous performance deduction in some dimensions. We interpret from the fact that the distribution of the data is not uniform and LLE preserves the distance from a point to its neighbors, the results will be inaccurate in the sparse area. Therefore, when projecting to some lower dimensions, it is hard to preserve the original geometric features, this results in overlapping points. As AE is more stable and robust than the LLE, we choose to use AE for the experiments.

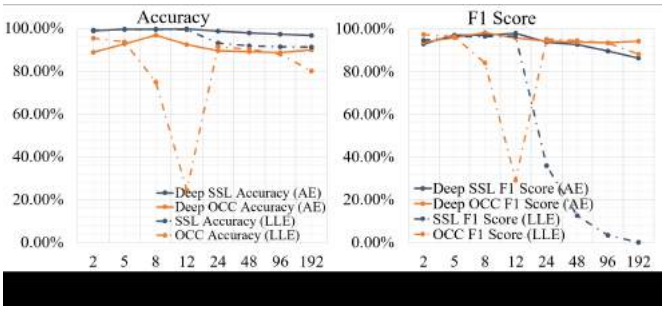


Fig. 10: The comparison of the AE and LLE.

D. Baseline of Supervised Learning for Deep SLL and OCC

As a reference for SLL and OCC, we conduct simulations for different supervised learning methods [38]–[41]. As the results are similar, we show the results of the support vector machine (SVM) and logistic regression (LR) in Fig. 11. The figure shows that when the provided information is little and the data set is unbalanced, the supervised learning method tends to classify all the data belonging to the minor class to be the majority class. This results in a fake high accuracy and the poor $F1$ score reveals the true performance. In Fig. 11, the x-coordinate is on hyper-parameter tuning. Therefore, we will only choose the dimension with the highest $F1$ score though we project data to different dimensions. Knowing this, we can conclude from the figure that middle to a relatively high projection dimension, which is between 8 to 48, helps to improve accuracy and $F1$ score. The results further indicates that more supervision, more information, but less noise ensures better results. Finally, Fig. 11 also shows that the results of the public data set and the utility data set are similar, which is also the case for SSL and OCC. So, we will focus on the utility dataset for the rest of the visualization work.

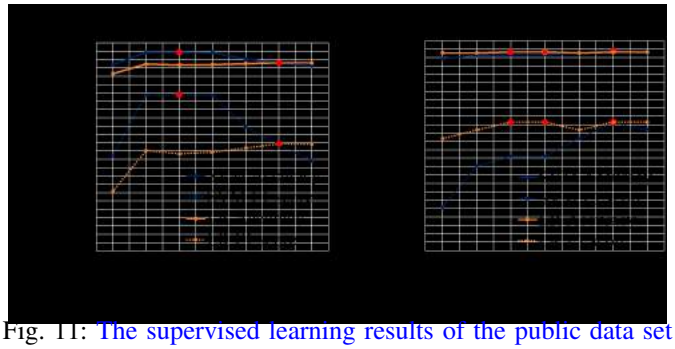
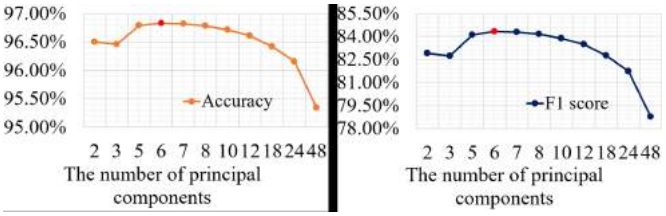


Fig. 11: The supervised learning results of the public data set and the utility data set.

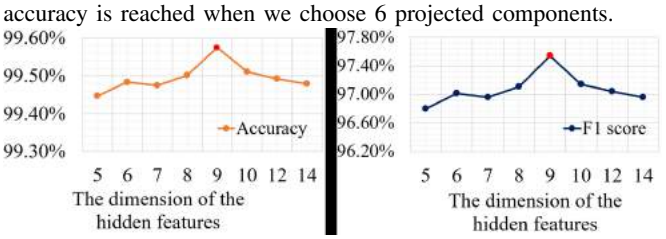
E. Feature Numbers for Linear and Nonlinear Representation

To understand how many features are needed in nonlinear representation learning of autoencoder, we plot the results in terms of the two performance metrics in Fig. 12, where we also show results of linear representation of PCA for comparison. In the sub-figures, we try to ensure the consistency in the setups for all the learning processes. For the deep semi-supervised learning (SSL) method, we choose to use the first 50 solar data and the first 50 non-solar data as the labeled data, all the other 1,923 solar data and 19,950 non-solar data as the unlabeled data. The proposed deep SSL method takes all the labeled data and the unlabeled data and infers the labels for the unlabeled data. For the deep one-class classification (OCC) method, we keep the same structure by using the first 50 non-solar data as the given class with remaining data representing unknown classes. The proposed deep OCC method interprets the labels for the rest of the 21,923 data based on the 50 non-solar data.

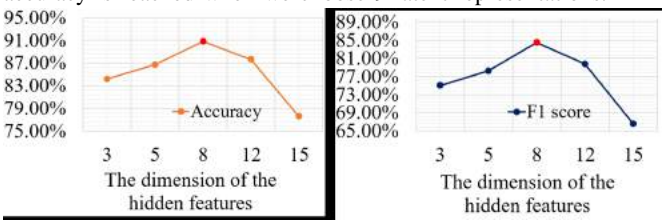
For the deep SSL method, as can be seen from Fig. 12a, when we increase the dimension of the projected principal components, the $F1$ score and the accuracy increase with little fluctuation until reaching the optimal, after which they decrease. The optimal value is reached when we choose 6 projected components. Also shown in Fig. 12b, when we increase the dimension of the hidden representations we extracted, the $F1$ score and the accuracy reach the optimal with little fluctuation, after which they finally decrease. The optimal value is reached when a 9-dimensional hidden representation is used. Although the accuracy of using PCA and AE is always above 95%, the true performance of the classification for the minor class may not be overly optimistic. For example, suppose we have 100 data points, out of which 95 are from



(a) The optimal number of features for semi-supervised learning using data after PCA. The red dot shows the optimal $F1$ score and accuracy is reached when we choose 6 projected components.



(b) The optimal number of features for proposed deep semi-supervised learning. The red dot shows the optimal $F1$ score and accuracy is reached when we choose 9 latent representations.



(c) The optimal number of features for proposed deep one-class classification. The red dot shows the optimal $F1$ score and accuracy is reached when we choose 8 latent representations.

Fig. 12: The optimal dimension for each method.

nonsolar users and 5 are from solar users. If the first algorithm reports that all the data points are from nonsolar users, the accuracy is 95% and the $F1$ score is 0. If the second algorithm successfully determines one data point from a solar user and predicts all the others as data from nonsolar users, the accuracy is 96% and the $F1$ score is 33%. We see that the $F1$ score successfully distinguishes the better performance of the second algorithm. The same conclusion applies to our results, the results of the autoencoder (AE) has a $F1$ score increase of more than 10%, representing significant improvement.

For the deep OCC method, as shown in Fig. 12c, the accuracy and the $F1$ score first increase to the peak and then decrease. The optimal value is reached when an 8-dimensional hidden representation is used. The deep OCC has a reasonable performance reduction in both accuracy and $F1$ score, it's acceptable because less information is provided. All aforementioned results indicate that a relatively low dimension is sufficient for learning. Higher-dimensional components may contain information that is harmful to the results, i.e., noises and bad data, so the results guide us to experiment on a dimension between 5 to 12 as the representations of the original data. The results also indicate that as PCA is a linear transformation of the input space aiming to find the directions that have higher variances, the projected data have low or close to zero correlation with each other. However, the electricity usage data used in our simulation are highly nonlinear and the features which are different timestamps are correlated with each other.

F. Performance Improvements for Deep SSL and Deep OOC

To better visualize the performance benefits of the proposed methods, we plot all the results together in Fig. 13. These results include supervised learning, SSL, OCC, with and without autoencoder components. The left graph illustrates the comparison of accuracy where the right graph is the $F1$ score. The dashed green line shows the performance of the supervised learning method based on support vector machine with radial basis functional (RBF) kernel. The dashed orange and navy line are the results of the classic SSL and classic OCC methods when using the projected data based on principal component analysis (PCA), respectively. The solid orange and navy lines are the performance of the proposed deep SSL and deep OCC methods when using the hidden representations extracted from the autoencoder (AE), respectively.

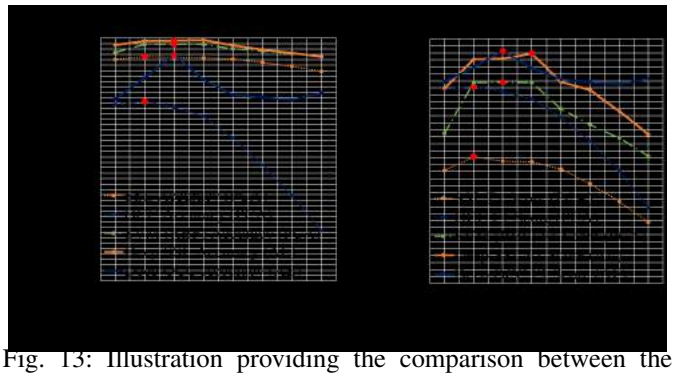


Fig. 13: Illustration providing the comparison between the accuracy and $F1$ score of the study results between the baseline supervised learning, the proposed deep SSL and deep OCC methods utilizing the projected data of the PCA and the hidden representation extracted from the AE.

For the three dashed lines, we can observe that the accuracy of supervised learning is always higher than the accuracy of the SSL and the accuracy of SSL is always higher than that of OCC, if we use the projected data after PCA. We also obtain a similar conclusion for the $F1$ score by ignoring the projection to 2 principal components. The results confirm that more information guarantees better performance.

Next, we focus on the performance of the proposed deep SSL method, which is shown by the orange dash line and the orange solid line in the figure. The performance curves first increase and then decrease as we increase the dimensionality of the projected data, either from PCA or AE. We conclude that a relatively low dimension, from 5 to 12 is enough to summarize the characteristics of electricity usage. The figure also shows that the accuracy has a clear improvement and the $F1$ score increases by more than 10% with the help of the AE, representing a significant improvement. The result also indicates that supervised learning tends to overfit the data when given limited information. The unlabeled data helps to improve the performance by providing more complete information on the distribution of the data.

Finally, we look at the performance of the regular OCC and the proposed deep OCC, which are shown by the navy dash

line and the navy solid line in the figure. We can observe from the figure that the AE can stabilize the accuracy and improve the $F1$ score, which is also an enhancement. The performance curves first increase and then decrease as we increase the dimensionality of the projected data, either from PCA or AE. While the performance of using the projected principal components has a sharp decline when the dimensionality of the projected data increases, the performance of using the hidden representations from the AE remains stable. This indicates that the nonlinear transformation of the AE guarantees the OCC method to find a good hypersphere regardless of the dimensionality. The performance of the proposed OCC is slightly worse than the supervised learning in terms of the accuracy, which is acceptable as the provided information is much less.

Overall, the proposed methods with the assistance of the AE provide greater accuracy and $F1$ scores than the supervised learning and merely using the principal components from PCA.

G. When to Choose SSL or OCC

Previously, we showed the outstanding performance of the autoencoder (AE) in assisting semi-supervised learning (SSL) and one-class classification (OCC). In this subsection, we discuss the limitation of SSL and when OCC is required. Specifically, we gradually reduce the number of the labeled data and observe the minimum labeled data to hold the performance of the AE. Learned from the previous results, the best performance of AE is reached when we reduce the dimensions of data to 8 or 12. Therefore, we present the simulation results of using AE to reduce the dimension to 12 in Fig. 14. The x-axis shows the size of the labeled data, for example, "5+5" represents the use of 5 labeled solar data and 5 labeled nonsolar data as the training data to conduct SSL. If we consider accuracy and F1 score below 90% as unacceptable performance, we should choose no less than 40 labeled solar data and 40 labeled nonsolar data as the training data to conduct the SSL. Otherwise, OCC is required.

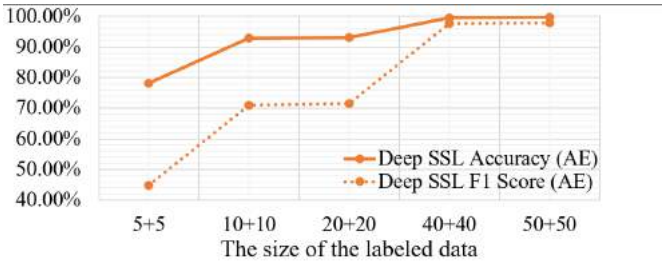


Fig. 14: The performance of SSL when we progressively reduce the number of the labeled data.

H. Computational Time

Method	Supervised learning	SSL (PCA)	SSL (AE)	OCC (PCA)	OCC (AE)
Average computation time	0.3 s	354.0 s	215.6 s	716.6 s	878.3 s

TABLE II: The average computation time for all the methods.

Table II shows the average computation time for each method based on a CPU *Intel(R) Xeon(R) CPU E5-2687W v4 @ 3.00GHz* and 64 *GB* memory. As illustrated in the table, supervised learning is the fastest, however, the method is infeasible when accurately labeled data cannot be accessed or data are highly unbalanced. SSL method and OCC method can relieve the above problems with a sacrifice of computation time. Among the results of the SSL methods, the AE used in the proposed method can accelerate the speed of the SSL method because the AE maps the data to $[-1, 1]$ and saves the computation cost. Among the results of the OCC methods, the AE used in the proposed method slows down the speed of the OCC method, this may be because the OCC method must compute the relative distance of the data, so the AE cannot reduce the computation time. Nevertheless, note that the AE improves the accuracy and $F1$ score for both the SSL method and OCC method. As the analysis of this work is offline, the required computation time is feasible.

I. Generalization Ability of the Proposed Methods

We have shown that the proposed methods can improve the accuracy of detecting active solar panel with a cost of time complexity. Finally, we will discuss the generalization ability of the proposed methods to different sizes of the data sets, different durations of data, different months (especially non-sunny months), and different grids.

1) Generalization ability to different sizes of the data sets: Specifically, we further test different sizes of the data set to determine method robustness, the results are shown in Fig. 15. As shown in the figure, for semi-supervised learning, the performance of PCA declines as we increase the size data set. Conversely, the AE maintains adequate performance. For one-class classification, when we vary the size of the data set, the AE experiences a slight performance decline, overall performance remains superior to PCA.

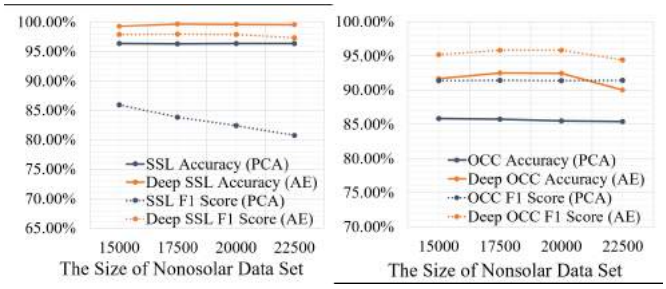
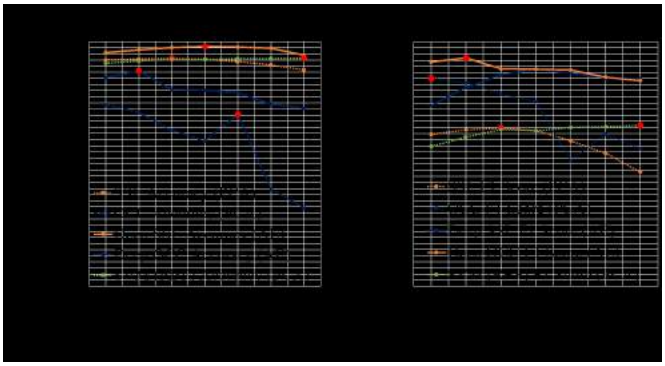
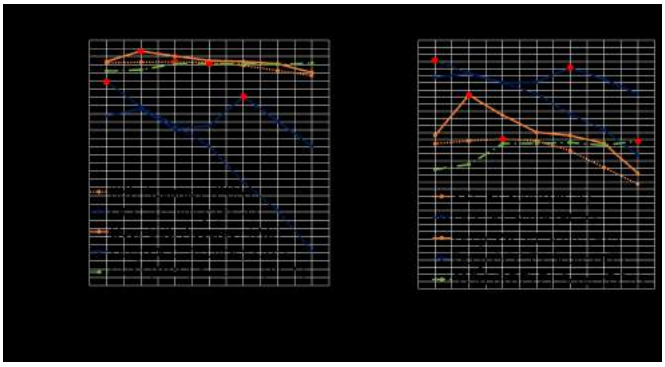


Fig. 15: The performance tendency when varying the size of the data set.

2) Generalization ability to different duration of data: As solar user detection is more for planning rather than for operation, the proposed methods are for offline analysis. While it is possible that a solar panel is installed in the middle of the period leading to wrong identification, the identification will be correct when the moving window covers more of the days after installation. Using less days may help reduce the wait, but will also reduce learning accuracy overall due to information lose for other data points. To evaluate the results with different data length, we test proposed methods on two



(a) The performance of the proposed methods on two weeks' worth of data.



(b) The performance of the proposed methods on one week's worth of data.

Fig. 16: The performance of the proposed methods on different duration of the data.

weeks' worth of data and on one week's worth of data. The results are shown in Figure 16a and 16b.

As shown in the figures, the proposed method maintains satisfactory performance when using a shorter duration of two weeks' worth of data. However, the proposed method becomes ineffective when we reduce the duration of data to one week. The results imply that when the data volume is less, the accuracy or the $F1$ score also deteriorates. Therefore, we need a reasonable length of data for the autoencoder to capture the nonlinear features that distinguish customers with active solar panels from customers without solar panels.

3) Generalization ability to different months: We have shown the simulation results of sunny months, next, we will discuss the results of the proposed methods of non-sunny months. Fig. 17 shows the separability difference between sunny month (June) and non-sunny month (November). Specifically, we project solar customers and non-solar customers onto two dimensions using principal component analysis. For summer, we plot the results on the left of Fig. 17, and there is little overlapping between the usage data from customers with solar panels and the usage data from customers without solar panels. This makes the detection easier. For winter, we plot the results on the right of Fig. 17, and the usage data from customers with solar panels are buried under the usage data from customers without solar panels in November. This makes the work more challenging for learning algorithms. With such observation, we conduct same simulation in November. The results show that the proposed methods obtain similar accuracy with a some deterioration, about 10%, in $F1$ score.

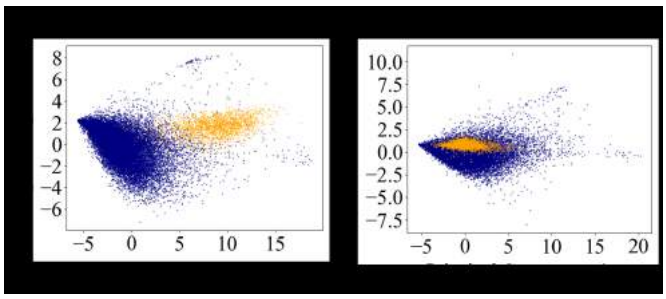


Fig. 17: The separability difference between sunny months and non-sunny months by projecting the data to two dimensions using principal component analysis.

4) Generalization ability to different grids: In addition, we test the methods with data from another grid provided by a partner utility to see if the proposed methods adapt well to other grids. This utility is in the southwest of the US, while the utility for the original simulation is in the northeast of the US. The data from the southwest utility contains around 350 users' billing meter and solar meter readings from October 2018 to October 2019. We select the billing meter data in June 2019, which is the same time range selected for the northeast utility, to conduct the simulation. For deep semi-supervised learning, the accuracy is 90.00% and the $F1$ score is 90.91%. For deep one-class classification, the accuracy is 80.77% and the $F1$ score is 80.00%. Although the accuracy and the $F1$ score decrease marginally, the results maintain acceptable performance. We will focus on improving the generalization ability in future work.

VII. Conclusion

In summary, solar detection is urgently needed as it is challenging and cost intensive to maintain accurate utility databases with current methods. Electric Distribution Companies need to have visibility of these assets to avoid potential risks of two-way power flow, e.g., outages and equipment damages. In this paper, we proposed a deep semi-supervised learning and a deep one-class classification approach to detect residential PV systems under different scenarios. The proposed methods use the extracted features from the autoencoder and combine them with the original label information to predict the labels for the rest of the data. The proposed methods have been validated on a utility data set and a publicly available data set and have shown their effectiveness and robustness for solving the solar panel detection problem.

References

[1] D. L. Donaldson and D. Jayaweera, "Effective solar prosumer identification using net smart meter data," *International Journal of Electrical Power & Energy Systems*, vol. 118, p. 105823, 2020.

[2] M. C. Di Piazza, A. Ragusa, G. Vitale *et al.*, "Identification of photovoltaic array model parameters by robust linear regression methods," in *International Conference on Renewable Energies and Power Quality*, 2009, pp. 143–149.

[3] V. Schwarzer and R. Ghorbani, "Transient over-voltage mitigation and its prevention in secondary distribution networks with high PV-to-load ratio," Hawai'i Natural Energy Institute, Tech. Rep. HNEI-02-15, February 2015, accessed August 2020.

[4] KHON2, "HECO customers asked to disconnect unauthorized PV systems," 2014, accessed August 2020.

[5] Z. Chi, L. Pandian, Z. Xueying, Z. Jie, Z. Wei, H. Jiajian, X. Yu, and X. Qi, "Research on the impacts of grid-connected distributed photovoltaic on load characteristics of regional power system," in *International Conference on Green Energy and Applications*, 2017, pp. 95–99.

[6] D. Cheng, B. A. Mather, R. Seguin, J. Hambrick, and R. P. Broadwater, "Photovoltaic (PV) impact assessment for very high penetration levels," *IEEE Journal of photovoltaics*, vol. 6, no. 1, pp. 295–300, 2015.

[7] A. Argüello, J. D. Lara, J. D. Rojas, and G. Valverde, "Impact of rooftop PV integration in distribution systems considering socioeconomic factors," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3531–3542, 2017.

[8] H. Tribak and Y. Zaz, "Remote solar panels identification based on patterns localization," in *International Renewable and Sustainable Energy Conference*, 2018, pp. 1–5.

[9] S. Lee, K. E. An, B. D. Jeon, K. Y. Cho, S. J. Lee, and D. Seo, "Detecting faulty solar panels based on thermal image processing," in *IEEE International Conference on Consumer Electronics*, 2018, pp. 1–2.

[10] K. Liao and J. Lu, "Using matlab real-time image analysis for solar panel fault detection with UAV," in *Journal of Physics: Conference Series*, vol. 1509, 2020, p. 012010.

[11] V. Golovko, S. Bezobrazov, A. Kroshchanka, A. Sachenko, M. Komar, and A. Karachka, "Convolutional neural network based solar photovoltaic panel detection in satellite photos," in *IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, vol. 1, 2017, pp. 14–19.

[12] Y. Bazi and F. Melgani, "Convolutional SVM networks for object detection in UAV imagery," *IEEE transactions on geoscience and remote sensing*, vol. 56, no. 6, pp. 3107–3118, 2018.

[13] X. Zhang and S. Grijalva, "A data-driven approach for detection and estimation of residential PV installations," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2477–2485, 2016.

[14] A. Salazar, G. Safont, and L. Vergara, "Semi-supervised learning for imbalanced classification of credit card transaction," in *International Joint Conference on Neural Networks*, 2018, pp. 1–7.

[15] M. Hajighorbani, S. R. Hashemi, B. Minaei-Bidgoli, and S. Safari, "A review of some semi-supervised learning methods," in *IEEE international conference on new research achievements in electrical and computer engineering*, 2016, pp. 250–259.

[16] Y. C. P. Reddy, P. Viswanath, and B. E. Reddy, "Semi-supervised learning: A brief review," *International Journal of Engineering & Technology*, vol. 7, no. 1.8, pp. 81–85, 2018.

[17] T. M. Mitchell, "Learning from labeled and unlabeled data part 2: coupled training," *Machine learning*, vol. 10, p. 601, 2009.

[18] M. Zhao, R. H. M. Chan, T. W. S. Chow, and P. Tang, "Compact graph based semi-supervised learning for medical diagnosis in alzheimer's disease," *IEEE Signal Processing Letters*, vol. 21, no. 10, pp. 1192–1196, 2014.

[19] P. Perera and V. M. Patel, "Learning deep features for one-class classification," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5450–5463, 2019.

[20] M. Ribeiro, M. Gutoski, A. E. Lazzaretti, and H. S. Lopes, "One-class classification in images and videos using a convolutional autoencoder with compact embedding," *IEEE Access*, vol. 8, pp. 86 520–86 535, 2020.

[21] K. Greff, S. Van Steenkiste, and J. Schmidhuber, "Neural expectation maximization," in *Advances in Neural Information Processing Systems*, 2017, pp. 6691–6701.

[22] H. Song, Z. Jiang, A. Men, and B. Yang, "A hybrid semi-supervised anomaly detection model for high-dimensional data," *Computational intelligence and neuroscience*, vol. 2017, no. 8501683, p. 9, 2017.

[23] X. Mai and R. Couillet, "A random matrix analysis and improvement of semi-supervised learning for large dimensional data," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 3074–3100, 2018.

[24] Y. Shang and W. Ruml, "Improved mds-based localization," in *IEEE INFOCOM 2004*, vol. 4, 2004, pp. 2640–2651.

[25] A. Makhzani and B. Frey, "K-sparse autoencoders," *arXiv preprint arXiv:1312.5663*, 2013.

[26] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research*, vol. 11, no. 12, 2010.

[27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[28] B. Hou, J. Yang, P. Wang, and R. Yan, "LSTM-based auto-encoder model for ECG arrhythmias classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 1232–1240, 2019.

[29] J. Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," in *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, vol. 25, no. Citeseer, 2011, pp. 59–62.

[30] M. Aiad and P. H. Lee, "Unsupervised approach for load disaggregation with devices interactions," *Energy and Buildings*, vol. 116, pp. 96–103, 2016.

[31] N. Kumar and S. P. Awate, "Semi-supervised robust mixture models in RKHS for abnormality detection in medical images," *IEEE Transactions on Image Processing*, vol. 29, pp. 4772–4787, 2020.

[32] Z. Li, L. Yang, and Z. Li, "Mixture-model-based graph for privacy-preserving semi-supervised learning," *IEEE Access*, vol. 8, pp. 789–801, 2019.

[33] M. Stritt, L. Schmidt-Thieme, and G. Poeppel, "Combining multi-distributed mixture models and bayesian networks for semi-supervised learning," in *International Conference on Machine Learning and Applications*, 2007, pp. 354–362.

[34] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[35] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*, 2018, pp. 4393–4402.

[36] Smart*, "Umass smart* dataset," Accessed July 31, 2020.

[37] Z. C. Lipton, C. Elkan, and B. Narayanaswamy, "Thresholding classifiers to maximize f1 score," *Machine Learning and Knowledge Discovery in Databases*, vol. 8725, pp. 225–239, 2014.

[38] O. Chapelle, P. Haffner, and V. N. Vapnik, "Support vector machines for histogram-based image classification," *IEEE transactions on Neural Networks*, vol. 10, no. 5, pp. 1055–1064, 1999.

[39] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural networks*, vol. 10, no. 5, pp. 1048–1054, 1999.

[40] B. Krishnapuram, L. Carin, M. A. Figueiredo, and A. J. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 6, pp. 957–968, 2005.

[41] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 11, pp. 4085–4098, 2010.