Mining Smart Meter Data for Improving Distribution Grid Operation and Resilience

Zhaoyu Wang
Harpole-Pentair Assistant Professor
Iowa State University

IEEE PES Big Data Tutorial Series
Contents

• Smart Meter Data Description
  - Real Utility Smart Meter Data and Data Sharing
  - Smart Meter Data Quality Problem
  - Smart Meter Data Pre-processing
  - How Smart Meter Data Benefits Utility Operations

• Machine Learning Knowledge for Smart Meter Data Analytics
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning
  - Artificial Neural Network

• Case Study I – Load Modeling and Inference
  - Background
  - Technical Approach
  - Results

• Case Study II – Cold Load Pick-up Estimation
  - Background
  - Technical Approach
  - Results

• Conclusion
Contents

• **Smart Meter Data Description**
  - Real Utility Smart Meter Data and Data Sharing
  - Smart Meter Data Quality Problem
  - Smart Meter Data Pre-processing
  - How Smart Meter Data Benefits Utility Operations

• **Machine Learning Knowledge for Smart Meter Data Analytics**
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning
  - Artificial Neural Network

• **Case Study I – Load Modeling and Inference**
  - Background
  - Technical Approach
  - Results

• **Case Study II – Cold Load Pick-up Estimation**
  - Background
  - Technical Approach
  - Results

• **Conclusion**
Power Distribution Grid Data

- Where does the data come from?
  - SCADA (supervisory control and data acquisition); Smart Meters; Protection Devices; (micro)PMUs (phasor measurement units)
  - Measures voltage/current/frequency at different resolutions

- What are smart meters?
  - Different from conventional energy meters
  - Stay in your homes (not every home has it)
  - Measure energy and voltage
  - 15/30/60-minute resolution

- What are barriers to apply big data techniques in power industry?
  - Critical infrastructure
  - Conservative
  - Confidentiality

Fig. 1 Power distribution grid.
Our Real Data from Utilities

- We have NDAs with following utilities: MidAmerican Energy, Alliant Energy, Cedar Falls Utilities, Algona Municipal Utilities, Maquoketa Valley Electric Coop, Bloomfield, ...
- We have multi-year PMU/SCADA/Smart Meter data from utility partners.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Utilities</th>
<th>Measurement Locations</th>
<th>Data Length</th>
<th>Renewable Penetration</th>
<th>Historical Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMI &amp; SCADA</td>
<td>MVEC</td>
<td>14,000 customers</td>
<td>24 months with continuous updating</td>
<td>~45% relative to peak</td>
<td>Yes</td>
</tr>
<tr>
<td>AMI &amp; SCADA</td>
<td>Alliant</td>
<td>10 substations</td>
<td>24 months with continuous updating</td>
<td>~35% relative to peak</td>
<td>Yes</td>
</tr>
<tr>
<td>AMI</td>
<td>CFU</td>
<td>2,500 customers</td>
<td>18 months with continuous updating</td>
<td>~10% relative to peak</td>
<td>Yes</td>
</tr>
<tr>
<td>PMU &amp; SCADA</td>
<td>MidAmerican</td>
<td>3 Substations</td>
<td>24 months with continuous updating</td>
<td>~40% relative to peak</td>
<td>Yes</td>
</tr>
<tr>
<td>AMI &amp; SCADA</td>
<td>Algona</td>
<td>3,000 customers</td>
<td>30 months</td>
<td>Unknown</td>
<td>N/A</td>
</tr>
<tr>
<td>AMR &amp; SCADA</td>
<td>Bloomfield</td>
<td>1,329 customers</td>
<td>18 months</td>
<td>10%</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Smart Meter Data Collection

K. K. Kee, S. M. F. Shahab and C. J. Loh, “Design and development of an innovative smart metering system with GUI-based NTL detection platform”
Exemplary Smart Meter Data from Utilities

- More AMI data and circuit models:
  - Duration: 4 years (2014 - 2018)
  - Measurement Type: Smart Meters and SCADA
  - Detailed circuit models of all feeders in Milsoft and exact smart meter locations
  - Data Time Resolution: 15 Minutes – 1 Hour
  - Customer Type:

<table>
<thead>
<tr>
<th>Utilities</th>
<th>Substations</th>
<th>Feeders</th>
<th>Transformers</th>
<th>Total Customer</th>
<th>Customers with Meters</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td>27</td>
<td>1726</td>
<td>9118</td>
<td>6631</td>
</tr>
</tbody>
</table>

- Customer Type:
  - Residential: 84.67%
  - Commercial: 14.11%
  - Industrial: 0.67%
  - Other: 0.55%
Exemplary Smart Meter Data from Utilities

Smart Meter Measurement
Data For Load Monitoring

Sample Customer Voltage

Network Topology/Model
Information

Sample Customer
Energy Consumption
Exemplary Smart Meter Data from Utilities

An exemplary distribution system and associated SM data from our utility partner:

System Information

- 2 substations
- 4 load tap changing substation transformers (69/13.8 kV)
- 14 feeders (83 miles)
- 1489 overhead line sections
- 2582 underground cable sections
- 5 capacitor banks
- 361 switching devices
- >1000 distribution transformers
- 5212 customers

SM Data

- Time period: 4 years (2015-2018)
- 4321 residential customers
- 696 small commercial customers
- 146 large commercial customers
- 17 industrial customers
- 32 other customers
- Time resolution:
  - Hourly – residential, small commercial
  - 15-min – large commercial, industrial
Data Sharing

With permission from our utility partner, we share a real distribution grid model with one-year smart meter measurements [1]. This dataset provides an opportunity for researchers and engineers to perform validation and demonstration using real utility grid models and field measurements.

- The system consists of 3 feeders and 240 nodes and is located in Midwest U.S.
- The system has 1120 customers and all of them are equipped with smart meters. These smart meters measure hourly energy consumption (kWh). We share the one-year real smart meter measurements for 2017.
- The system has standard electric components such as overhead lines, underground cables, substation transformers with LTC, line switches, capacitor banks, and secondary distribution transformers. The real system topology and component parameters are included.
- You may download the dataset at: http://wzy.ece.iastate.edu/Testsystem.html, including system description (in .doc and .xlsx), smart meter data (in .xlsx), OpenDSS model, and Matlab code for quasi-static time-series simulation.

Test system diagram

The dataset has been viewed/downloaded more than 10,000 times since June 12, 2019
Smart Meter Data Quality Problem

*SM data quality problems:*

- Outliers/Bad Data
- Missing Data
- Data Reset
- Repetitive Data

*Reasons that cause SM data quality problems:*

- **Intermittent errors** – large noise or temporary failures due to the communication failure or meter malfunction.

- **Systematic errors** – deterioration of measurements due to age, temperature, weather, and other environmental effects [2].
Smart Meter Data Quality Problem
Smart Meter Data Quality Problem

- Communication Failure (Missing Data)
- Anomalous Data Samples (More than 15 Times Load Capacity)
How to identify and correct the missing/bad data?

Solution for Missing/Bad Data identification:
• Exclude the outage cases, the missing data is identified based on invariant record.

• A statistical outlier detector (e.g., z-score) is used to identify the bad data.

\[ Z_i = \frac{x_i - \mu}{\sigma} \]

The samples with z-score outside a range of ±5 are identified as the bad data.

Solution for Missing/Bad Data Correction:
• The discontinuously missing/bad data is filled/corrected by taking an average of the two neighboring measurements.

• The continuously missing/bad data is filled/corrected by taking an average of the 1000 random samples generated from historical demand probability distributions. These probability distributions were developed for each hour using probability density estimators (e.g., kernel density estimation (24 PDFs in total)).
Contents

• Smart Meter Data Description
  - Real Utility Smart Meter Data and Data Sharing
  - Smart Meter Data Quality Problem
  - Smart Meter Data Pre-processing
  - How Smart Meter Data Benefits Utility Operations

• Machine Learning Knowledge for Smart Meter Data Analytics
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning
  - Artificial Neural Network

• Case Study I – Load Modeling and Inference
  - Background
  - Technical Approach
  - Results

• Case Study II – Cold Load Pick-up Estimation
  - Background
  - Technical Approach
  - Results

• Conclusion
How Smart Meter Data Benefits Utility Operations?

- The high-resolution data from SMs provides rich information to model consumption behaviors of consumers.
  - Load profiling
  - Load forecasting
  - Peak contribution estimation
  - Behind-the-metering generation modeling

- The large amount of customer metering data provides an unique opportunity for using data-driven techniques to improve distribution grid real-time monitoring and control.
  - Distribution system state estimation
  - VVC/CVR
  - Outage detection
  - Demand response implementation
Smart Meter Data Analytics

Fig. 2 Number of publication about SM data analytics [3].

Fig. 3 Number of publication in nine most popular journals in 2017 [3].

• In recent year, researchers have put considerable efforts in SM data analytics.

• The number of publications increases rapidly from 2012.
Fig. 4 Taxonomy of SM data analytics [3].
What is Machine Learning?

“Learning is any process by which a system improves performance from experience.”

- Herbert Simon

**Traditional Programming**

Data → Computer → Output

Program

**Machine Learning**

Data → Computer → Program

Output
Types of Machine Learning

**Supervised learning**
- Given training data with desired labels
- Classification (categorical labels)
- Regression (real-valued labels)

**Unsupervised learning**
- Given training data without desired labels
- Data clustering
- Independent component analysis
- Probability distribution estimator
- Dimension reduction

**Reinforcement learning**
- Given a sequence of states and actions with rewards
- Output is a policy
- Policy is a mapping from states to actions
- Decision making

Fig. 5 Artificial intelligence methods.
Supervised Learning: Regression

Regression Model:

• Given \((x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\)

• Learn a function \(f(x)\) to predict \(y\) given \(x\)
  - When \(y\) is real-valued, this learning process is called regression.
Supervised Learning: Classification

Classification Model:
- Given \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\)
- Learn a function \(f(x)\) to predict \(y\) given \(x\)
  - When \(y\) is categorical, this learning process is called classification.

Based on slide by Eric Eaton
Unsupervised Learning: Clustering

Clustering Model:

- Given $x_1, x_2, ..., x_n$
- Represent a form of data summarization
- Discover hidden structures and patterns behind the $x$’s
- Group data with similar patterns together

Based on slide by Eric Eaton
Unsupervised Learning: Clustering

- **Probabilistic techniques**
  - Identify the cluster with a *certain probabilistic model* whose unknown parameters have to be found.
  - Assume the data were generated from a mixture of $k$ probability distributions.
  - One advantage of probabilistic-based models is that they can be easily generalized to different types of data.
    - For numerical data, we may use a Gaussian mixture model.
    - For categorical data, we may use a Bernoulli model.
    - For sequence data, we may use a hidden Markov model.
Unsupervised Learning: Clustering

- **Distance-based techniques**
  - Optimize global criteria based on the distance between data.
  - Distance-based algorithms can be generally divided into two types:
    - Flat: e.g., k-Means, k-Medians.
    - Hierarchical: e.g., agglomerative, divisive.
  - Distance-based algorithms can be used with almost any data type when an appropriate distance function is created for that data type.
  - For *high-dimensional data*, the quality of the distance functions reduces because of many irrelevant dimensions [5].
Unsupervised Learning: Clustering

- **Density-based techniques**
  - Optimize local criteria based on density distribution of data.
  - Classical density-based clustering methods: Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Statistical Information Grid (STING)
  - One advantage is that these methods can handle outliers within the dataset.
  - Hard to use in a discrete or non-Euclidean space and the density computations becomes significantly difficult to define with greater dimensionality.
Reinforcement learning

Agent and environment interact at discrete time step: \( t = 0, 1, \ldots, K \)

- Agent observes state at step \( t \): \( s_t \in S \)
- Produces action at step \( t \): \( a_t \in A(s_t) \)
- Gets resulting reward: \( r_{t+1} \in R \)
- Gets resulting next state: \( s_{t+1} \)

Based on slide by Sutton & Barto
Reinforcement learning

Objective: Win the game!

State: Position of all pieces

Action: Where to put the next piece down

Reward: 1 if win at the end of the game, 0 otherwise

Fig. 7 The game of Go.
One Commonly-used Machine Learning technique: Artificial Neural Network

Artificial neural network (ANN) is one of the most important machine learning techniques that simulates the learning mechanism of biological neural network.

Computation units in ANN are defined as neurons which are connected to one another through weights.

An ANN computes a function of the inputs by propagating the computed values from the input neurons to the output neurons and using the weight as intermediate parameters [8].
One Commonly-used Machine Learning technique: Artificial Neural Network

![Multi-layer ANN structure](image1)

![Single layer ANN structure (perceptron)](image2)

Input to Hidden Layer \[\bar{h}_1 = \sigma(W_1^T \bar{x} + b_1)\]

Hidden to Hidden Layer \[\bar{h}_{n+1} = \sigma(W_{n+1}^T \bar{h}_n + b_{n+1}) \quad \forall n \in \{1, ..., m-1\}\]

Hidden to Output Layer \[\bar{o} = \sigma(W_{m+1}^T \bar{h}_m + b_{m+1})\]

where, \(\sigma\) is the nonlinear activation function (i.e., sigmoid, hyperbolic tangent, or rectified linear unit.)
One Commonly-used Machine Learning technique: Artificial Neural Network

How can we determine the hyperparameters of ANNs (i.e., number of layers, the number of neurons, and the types of activation.)?

- Manually or automatically select optimum hyperparameter combinations by using high computation complexity strategies, such as grid search and random search [9].

How can we ensure the generalization ability of ANNs?

- In order to ensure the generalization ability of ANNs, different strategies are developed to prevent overfitting problem which occurs when the model shows low bias but high variance, such as early stopping, regularization, cross-validation, dropout, etc.

ANNs can be utilized to perform the supervised learning (regression/classification), the unsupervised learning (clustering, generative model), and the reinforcement learning (deep reinforcement learning).

A deep neural network (DNN) is an ANN with multiple layers between the input and output layer.
Contents

• Smart Meter Data Description
  - Real Utility Smart Meter Data and Data Sharing
  - Smart Meter Data Quality Problem
  - Smart Meter Data Pre-processing
  - How Smart Meter Data Benefits Utility Operations

• Machine Learning Knowledge for Smart Meter Data Analytics
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning
  - Artificial Neural Network

• Case Study I – Load Modeling and Inference
  - Background
  - Technical Approach
  - Results

• Case Study II – Cold Load Pick-up Estimation
  - Background
  - Technical Approach
  - Results

• Conclusion
Case Study I – A Multi-Timescale Data-Driven Approach to Enhance Distribution System Observability
Case Study I: Background

Distribution system state estimation (DSSE) is the process of inferring the values of distribution system’s state variables using a limited number of measured data at certain locations in the system [10].

Fig. 11 DSSE function in smart grid environment [10].
Case Study I: Background

At presents, distribution systems can be divided into three groups according to the observability: fully observable systems, partially observable systems and fully unobservable systems.

![Distribution systems with different observability](Image)

Fig. 12 Distribution systems with different observability.
Case Study I: Overview Framework

Problem Statement:
Inferring hourly consumption data from customer monthly billing information as pseudo-measurements

Challenges:
• Loss of correlation between consumption time-series at different time-scales
• Unobserved customers’ unknown typical behaviors

Solution Strategy:
Extending observability from observed customers to unobserved customers

Proposed Solution:
• Using data clustering for capturing customer typical behaviors
• Multi-timescale load inference (stage by stage inference chain)
• Using state-estimation-based Bayesian learning for inferring unobserved customers’ typical behaviors

Fig. 13 Overall structure of the proposed method.
Evidence from Data: How to maintain high correlation

- Hourly load variations cannot be directly observed at the monthly level.

- An efficient inference model needs to keep the high correlation level between different time resolution data.

Fig. 14 Correlation between different customers’ smart meter and correlation between Consumption at Different Time-Scales.
Using Observed Customers’ Data for Training Multi-Timescale Load Inference Chain Models

Layer I
- Monthly Consumption
- Weakly Consumption
- Daily Consumption
- Hourly Consumption

Layer II
- Extends observability using data of customers with smart meters to obtain a stage-by-stage consumption transition process (Maintains High Correlation!)

Fig. 15 Multi-timescale learning structure.

$E_M$ – Monthly Consumption
$E_W$ – Weakly Consumption
$E_D$ – Daily Consumption
$E_H$ – Hourly Consumption

✓ Extends observability using data of customers with smart meters to obtain a stage-by-stage consumption transition process (Maintains High Correlation!)
Multi-Timescale Load Inference Chain Model Description

• Layer I maps total monthly consumption, $E_M$, to the set of weekly consumption values $E_W = \{E_{W1}, \ldots, E_{W4}\}$ using ANNs connected in series.

• To capture the temporal correlation between consumption behaviors, each week’s estimated consumption is also fed to the next ANN corresponding to the following week’s consumption.

• Above idea is generalized to all the layers of the proposed multi-timescale load inference method.
How to Update the ANN Weight and Bias

- The Levenberg-Marquardt (LM) backpropagation method is used to update the ANN weight and bias variables [24].

- The training objective function and the update equation of LM algorithm can be written as:

  \[
  \min_b F(b) = \sum_{i=1}^{Q} v_i^2(b) = v^T(b) v(b)
  \]

  \[
  \Delta b_l = -[J^T(b_l)J(b_l) + \mu_l I]^{-1} J^T(b_l) v(b_l)
  \]

  where, \( \mu_l \) is the combination parameter at iteration \( l \), \( b \) is the set of learning parameters, \( J \) is the training objective function’s Jacobian, \( I \) is the identify matrix, \( v \) is the error vector, \( T \) is the matrix transposition operation, and \( \Delta b_l \) defines the learning parameter updates at each iteration.

- In each iteration, the value of \( \mu_l \) is updated based on the change of approximated performance index \( F(b) \). If a smaller value is obtained, the \( \mu_l \) is divided by some factor \( \vartheta > 1 \). Otherwise, \( \mu_l \) is multiplied by \( \vartheta \) for the next iteration.
Hyperparameter Calibration Results

- For each ANN, the dataset is randomly divided into three separate subsets for training (70% of the total data), validation (15% of the total data), and testing (15% of the total data).

- *Early stopping mechanism* and *noise injection strategy* are utilized to reduce the overfitting in the training process.

- Several hyper-parameters are calibrated using the grid search methods.
  - The number of hidden layer.
  - The number of neurons.
  - The value of learning rate.

Fig. 16 Calibration result for ANN.
Sensitivity Analysis of Observability

Fig. 17 Performance of quantifying learning reliability.

20% of Customers are Observed
Problem: Performance of Multi-timescale Chain Models Highly Depend on Typical Daily Consumption Patterns of Different Customers

Solution: Assign a Multi-Timescale Model to Each Typical Load Behavior Pattern Discovered From Observed Loads (Method: Spectral Clustering)

Train Load Inference Chain Models Using the Data of Observed Customers Belonging to Each Cluster ($C_i$)
Spectral Clustering

Advantages of the spectral clustering:

• Spectral clustering can work with *arbitrary complex* data, because it does not make assumptions on the shapes of clusters.

• Spectral clustering is able to handle the *high-dimensional time-series load data*, because it leverages the inherent redundancy of the *entire* distance matrix and enhances distance representations of newly embedded data in a lower dimensional space.

• Unlike the standard spectral clustering that relies on a *scaling parameter* $\alpha$ to measure the similarity between two data samples, a local parameter strategy is applied in our work. This strategy not only provides a way for automatic pick the $\alpha$ but also effectively exploits the local statistics for handling multi-scale data.
Spectral Clustering

Using the spectral clustering to develop the consumption pattern bank can be summarized as a *three-step* algorithm:

1) The first step is to transform SM dataset into an *undirected similarity graph* $G$.

2) The data points are embedded in a space, in which the clusters are more *observable*, with the use of the *eigenvectors of the graph Laplacian*.

3) A classical clustering algorithm (e.g., k-mean) is applied to *partition the embedding*.
Spectral Clustering

Fig. 18 Example illustrating three steps of spectral clustering [21].

Fig. 19 Spectral clustering on toy datasets [22].
The spectral clustering is utilized to classify different load shapes and to create the consumption pattern banks.

Fig. 17(a) shows the typical load patterns for different types of customers for weekends.

Fig. 17(b) shows the typical load patterns for different types of customers for weekdays.

Red represents Industrial, blue represents commercial, and black represents residential customers.

Fig. 20 Typical pattern banks for weekday and weekend.
Unobserved Customers’ Pattern Identification and Hourly Consumption Inference

- **Basic Idea**: Pick the Cluster that has the Best State Estimation Performance for Each Customer

- **Methodology**: Assign and Update Probability Values to Different Clusters Based on State Estimation Residuals (Recursive Bayesian Learning)

- **Outcome**: Pick the Most Probable Cluster for Each Unobserved Customer and Use its Corresponding Chain Model for Hourly Load Inference
Branch Current State Estimation

BCSE algorithm:

- Three-phase feeder model
- Use *branch current magnitudes* and *phase angles* as state variables
- Insensitive for branch impedance
- Better performance in computation speed and memory usage
- Weighted Least Square (WLS) estimator

$$\min_{x} J = (z - h(x))^T \Sigma (z - h(x))$$

$$G(x) = H^T(x) \Sigma H(x)$$

$$[G(x^m)] \Delta x^m = H^T(x^m) \Sigma (z - h(x^m))$$

$$x^{m+1} = x^m + \Delta x^m$$

Note: $z$ is the real measurement, $x$ is the state variable, $h(.)$ is the nonlinear measurement function, $\Sigma$ is the weight matrix that represents the users confidence in the measured data.

Fig. 21 A 18-node real utility feeder case.
The probability of each typical load pattern is computed using the residuals of the BCSE algorithm and the recursive Bayesian learning model.

The residuals are calculated by comparing the real measurements of several feeder-level devices and the estimated values from the BCSE.

Applying the Bayes theorem and assuming a Gaussian distribution for measurement error, a recursive expression for updating the probability of each load pattern over time can be written as [25]:

$$ p_{i,j}^o = \frac{\exp \left( -\frac{1}{2} r_{i,j}^o T * \Phi * r_{i,j}^o \right) * p_{i,j}^{o-1}}{\sum_{t=1}^{N} \exp \left( -\frac{1}{2} r_{i,j}^o T * \Phi * r_{i,j}^o \right) * p_{i,j}^{o-1}} $$

Where, $o$ is the iteration count, $r_{i,j}^o$ is the residual vector of the i’th class with respect to j’th customer and is computed by the corresponding state and real measurement vectors, $\Phi$ is a diagonal matrix that represents the variances corresponding to the branch current real/imaginary part residuals.
Numerical Results: Unobserved Individual Customer Hourly Load and Pattern Inference

Fig. 22 Inferring the hourly demand of an unobserved residential load in one month (average estimation error $\approx 8.5\%$ of total energy)

Fig. 23 Impact of accurate consumption pattern identification on the accuracy of the inference (industrial load patterns are close and stable)
Numerical Results: DSSE Performance Using the Proposed Method

- Apply the MAPE criterion to evaluate the accuracy of BCSE based on the proposed pseudo load estimation.

- The MAPE values for the voltage magnitude and phase angle are around 0.704% and 0.24%, respectively.

- In the previous work [11], the MAPE values are around 0.73% and 0.36%, respectively with 20% maximum Gaussian error for pseudo measurements.

Fig. 24 BCSE-based state estimation performance using the proposed load inference model.
Contents

• Smart Meter Data Description
  - Real Utility Smart Meter Data and Data Sharing
  - Smart Meter Data Quality Problem
  - Smart Meter Data Pre-processing
  - How Smart Meter Data Benefits Utility Operations

• Machine Learning Knowledge for Smart Meter Data Analytics
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning
  - Artificial Neural Network

• Case Study I – Load Modeling and Inference
  - Background
  - Technical Approach
  - Results

• Case Study II – Cold Load Pick-up Estimation
  - Background
  - Technical Approach
  - Results

• Conclusion
Case Study II - A Data-Driven Framework for Assessing Cold Load Pick-up Demand in Service Restoration
What is CLPU?

- In distribution systems, thermostatically controlled loads (TCLs) are **diversified** during normal operation, and **undiversified** in restoration.

- The phenomenon of losing load diversity is called Cold load pick-up (CLPU) and can cause considerable demand *increase*.

- Serious *consequences*: restoration failure, transformer aging and overloading, and unacceptable voltage drops.

- It is necessary to *assess* historical CLPU demand, and *extract* useful information to predict CLPU demand in future restorations.
**Background**

- **Typical exponential CLPU model**

  The CLPU ratio is used to predict CLPU demand in future restorations.

  \[
  \text{CLPU ratio} = \frac{P_u}{\hat{P}_d}
  \]

  - \(P_d\) -- Normal demand
  - \(\hat{P}_d\) -- Estimated normal demand
  - \(P_u\) -- Undiversified demand at the time of restoration

  Real CLPU at the feeder level
Literature Review

• Previous works have mainly focused on *model-driven* methods to obtain CLPU ratios [1-3]
  ➢ Use thermostatically controlled load models and thermal parameters to model houses

• *Drawbacks*:
  ➢ Need to collect detailed house-level thermal *parameters*
  ➢ Need to model *individual* thermostatically controlled load

Obtain CLPU ratio at feeder-level, using **LS-SVM** prediction approach

Determine customer demand increase, using **GMM** and probabilistic reasoning

Obtain useful statistics at feeder- and customer-level to fully quantify CLPU demand

---

$$P_d$$ -- Feeder-level normal demand  
$$T$$ -- Temperature  
$$b^*, \alpha^*$$ -- Trained model Parameters  
$$\hat{p}^t_{d}$$ -- Estimated feeder-level normal demand at $$t_r$$  
$$p^t_{u,i}$$ -- Undiversified feeder-level demand at $$t_r$$

$$p_{d,i}$$ -- Historical customer-level normal demand  
$$C_i$$ -- Historical customer contribution factor  
$$(\hat{p}^t_{d}, C_i)$$ -- Joint distribution of $$\hat{p}^t_{d}$$ and $$C_i$$  
$$p^t_{u,i}$$ -- Customer demand at the time of restoration  
$$\hat{f}^t_{i}$$ -- Estimated customer demand increase due to CLPU
Feeder-Level CLPU Demand Assessment

- **Objective**: Obtain CLPU Ratios from historical outage events, and develop a CLPU ratio regression model

- **Methodology**: Estimate the feeder-level normal demand assuming *the outage did not happen*

- **Algorithm**: Least-squares support-vector machines (LS-SVM)
LS-SVM

• Why do we choose LS-SVM?
  ➢ Advantages: good generalization capability, and low susceptibility to local minima, etc.

Let’s start with SVM.

• What is Support Vector Machine (SVM)?
  ➢ It is a **supervised** machine learning algorithm
  ➢ It can be used for **classification** or **regression**

• How does SVM work?
  Let’s start with classification.
  In this scenario,
  ➢ hyper-plane “B” **can** successfully segregate the two classes.
  ➢ hyper-plane “A” and “C” **cannot**.
LS-SVM

• **How does SVM work?**

In this scenario, hyper-planes A, B and C all can segregate the two classes.

Which hyper-plane is **best**?

- By drawing perpendicular lines, we can know that maximizing the *distances* between nearest point and hyper-plane can determine the optimal hyper-plane.
- This distance is called *margin*, the nearest points are called *support vectors*.
- A large margin means less chance of *misclassification*, and this is what we want.
LS-SVM

• How does SVM work?

In this scenario, the previous linear hyper-plane cannot work. What can we do?

**Solution:** Introduce higher-order features. For instance, instead of using the original $p$ features:

$$X_1, X_2, \ldots, X_p,$$

we enlarge the feature space to $2p$:

$$X_1, X_1^2, X_2, X_2^2, \ldots, X_p, X_p^2.$$

In general, a **kernel**, which has a form of $K(x_i, x_i')$, is used to enlarge feature space.
LS-SVM

• *How does SVM work?*

Now we extend the classification to regression.
• Similarly, for regression, we use support vectors to determine an optimal margin.
LS-SVM

• **How does SVM work?**

  • In our problem, we have

    
    \[
    P_t = w^T \phi(x_t) + b
    \]

    \[
    P_t \text{ denotes the normal demand at time } t
    \]

    \[
    x_t = [P_{t-1}^t, ..., P_{t-n_{lag}}^t, T_t]^T
    \]

    \[
    \text{where, } P_{t-1}^t \text{ denotes the normal demand at time } t - 1, n_{lag} \text{ is the time lag, } T_t \text{ is the temperature at time } t.
    \]

    \[
    \phi(x) \text{ denotes the nonlinear transformation.}
    \]

    \[
    w \text{ and } b \text{ are regression parameters, where } w = [w_1, ..., w_n]^T
    \]

• To train the **regression model**, first, we define:
LS-SVM

• Then, maximizing the **margin** is formulated as

\[
\min_{w,b} \frac{1}{2} w^T w \\
\text{s.t.} \quad P_d^t = w^T \phi(x_t) + b, \quad t = 1, ..., N
\]

where, \( x_t = [P_d^{t-1}, ..., P_d^{t-n_{\text{lag}}}, T_t]^T \).

• By introducing a “**Least Square**” term, we obtain LS-SVM

\[
\min_{w,b,e} \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{t=1}^{N} e_t^2 \\
\text{s.t.} \quad P_d^t = w^T \phi(x_t) + b + e_t, \quad t = 1, ..., N
\]

where, \( e_t \) denotes the estimation residual, \( \gamma \) is a regularization constant.
To solve this optimization problem, the Lagrangian, $\mathcal{L}$, is constructed as a function of regression parameters:

$$
\mathcal{L}(w, b, e; \alpha) = J(w, e) - \sum_{t=1}^{N} \alpha_t \{w^T \phi(x_t) + b + e_t - P_d^t\}
$$

where, $\alpha_t$ are Lagrange multipliers.

The solution is given by

$$
\begin{bmatrix}
    b^* \\
    \alpha^*
\end{bmatrix} = \begin{bmatrix}
    0 & 1^T \\
    1 & \Omega + \frac{1}{\gamma} I
\end{bmatrix}^{-1} \begin{bmatrix}
    0 \\
    P_d
\end{bmatrix}
$$

where, $P_d = [P_d^1, ..., P_d^N]^T$, $1 = [1, ..., 1]^T$, $\alpha = [\alpha_1, ..., \alpha_N]^T$, $\Omega_{t't} = K(x_{t'}, x_t) = \exp\left(-\frac{||x_{t'} - x_t||^2}{\sigma^2}\right)$, $t', t = 1, ..., N$, $\sigma$ is the width of Gaussian Kernel.
LS-SVM

• Then, the *normal demand at the time of restoration* is estimated as

\[
\hat{P}_{d_{tr}} = \sum_{t=1}^{N} \alpha_t^* K(x_t, x_{t_r}) + b^*
\]

where, \(x_t = [P_d^{t-1}, ..., P_d^{t-n_{lag}}, T_t]^T\), \(x_{t_r}^* = [P_d^{t_{tr}-1}, ..., P_d^{t_{tr}-n_{lag}^*}, T_{t_r}]^T\), \(t_r\) is the time of restoration, \(n_{lag}^*\) is the optimal time lag which is obtained by grid search.

• Finally, we calculate the **CLPU ratio** via

\[
\hat{R}_{CLPU} = \frac{P_{u_{tr}}}{E\{\hat{P}_{d_{tr}}\}}
\]

when, \(P_{u_{tr}}\) is the recorded *undiversified* feeder demand at \(t_r\). \(E\{\cdot\}\) denotes the *empirical averaging operator*, this is based on the consideration that \(\hat{P}_{d_{tr}}\) follows a distribution due to estimation errors.
Feeder-Level CLPU Ratio Result

- Developed CLPU ratio regression model
- Expected CLPU ratio with respect to temperature and outage duration
- CLPU ratio increases and saturates with outage duration
- CLPU ratio is sensitive to ambient temperature
At the feeder level, we obtained historical CLPU ratios. Can we apply the same approach to the customer level?

- Individual customer consumption can be quite *stochastic*.
- Considering this uncertainty, the feeder-level approach cannot be directly applied to customer-level demand estimation.
- We seek to use **GMM** and *probabilistic reasoning* to estimate the distribution of normal customer demand at the time of restoration.
Customer-Level CLPU Demand Assessment

- **Objective**: Quantify the CLPU demand increase caused by outage.
- **Methodology**: Estimate the distribution of customer-level normal demand assuming the outage did not happen.
- **Algorithm**: Gaussian mixture model (GMM) and probabilistic reasoning.
Why do we need GMM?

• The estimated feeder-level normal demand, \( \hat{P}_{d}^{tr} \), follows a distribution due to regression residuals.
• The historical customer contribution factor, \( C_i \), also follows a distribution due to the uncertainty of customer demand. Note that historical \( C_i \) is calculated by \( C_i = p_{d,i} / P_d \).
• The bivariate pair, \( \{ \hat{P}_{d}^{tr}, C_i \} \), forms a 2-dimensional empirical histogram.

• This 2-dimensional histogram does not strictly fit a single distribution model. Therefore, a mixture model should be used to represent the empirical histogram.
• In our problem, we used Gaussian mixture models (GMM).
GMM

How does GMM work?
• It approximates *arbitrary* probability density functions (PDFs) using a weighted summation of Gaussian density components

![Graph showing 1-dimension empirical histogram approximated by two Gaussian density components.](image)

The 1-dimension empirical histogram (bars) is approximated by two Gaussian density components (yellow and green lines)
For our problem, we approximate the joint 2-dimensional PDF of $\hat{P}_{d}^{tr}$ and $C_i$, using multiple Gaussian functions

$$f (\hat{P}_{d}^{tr}, C_i) = \sum_{j=1}^{S_i} w_j g_j (\hat{P}_{d}^{tr}, C_i)$$

where, $g_j(\cdot)$ denotes a bi-variate Gaussian function, $w_j$ is the weight corresponding to each $g_j(\cdot)$, $S_i$ is the total number of Gaussian functions. Note that $w_j$ and the parameters in $g_j(\cdot)$ are determined by the maximum likelihood (ML) estimation, using the empirical histogram.
**Customer-level Demand Increase Estimation**

**Estimate customer-level normal demand and demand increase?**

- Note that the estimated customer-level normal demand, $\hat{p}^t_d$, is determined by
  
  $$\hat{p}^t_d = \hat{p}^t_d \times C_i$$

- Until now, we know the joint distribution of $\hat{p}^t_d$ and $C_i$, that is, $f(\hat{p}^t_d, C_i)$, therefore, we can compute the distribution of the estimated customer-level normal demand using
  
  $$h(\hat{p}^t_{d,i}) = \int_{0+}^1 f^* \left( \frac{\hat{p}^t_{d,i}}{C_i}, C_i \right) \frac{1}{C_i} dC_i$$

- Finally, the distribution of demand increase of the $i$th customer is calculated as
  
  $$q(\hat{i}^t_i) = h(p^t_{u,i} - \hat{i}^t_i)$$

  where, $p^t_{u,i}$ is the actual customer demand at the time of restoration, which is recorded by smart meter.
Customer-level Result

(a) Distribution of estimated customer normal demand. (b) Distribution of customer demand increase.

Distributions of aggregate demand increase of a group with different number of customers.
Customer-level Result

Expected customer contribution to CPLU demand as a function of outage duration and ambient temperature in summer
Contents

• Smart Meter Data Description
  - Real Utility Smart Meter Data and Data Sharing
  - Smart Meter Data Quality Problem
  - Smart Meter Data Pre-processing
  - How Smart Meter Data Benefits Utility Operations

• Machine Learning Knowledge for Smart Meter Data Analytics
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning
  - Artificial Neural Network

• Case Study I – Load Modeling and Inference
  - Background
  - Technical Approach
  - Results

• Case Study II – Cold Load Pick-up Estimation
  - Background
  - Technical Approach
  - Results

• Conclusion
Conclusion

• We have archived a large amount of real smart meter data from our utility partners, which provides an unique opportunity to improve distribution grid modeling.

• The development of machine learning techniques provides a useful tool to process and analyze smart meter data.

• A data-driven load inference method is developed for enhancing distribution system observability.

• We have used smart meter data to model the cold load pick up, which can help utilities design restoration plans.
Reference:


Reference:

Reference:

Reference:

Thanks!

Q & A