Prediction Models for Dynamic Decision Making in Smart Grids

Saima Aman

Committee
Prof. Viktor K. Prasanna
Prof. Cauligi Raghavendra
Prof. Cyrus Shahabi
Saima Aman

Ph.D. candidate in Computer Science (2010-present)
University of Southern California (USC)
Research Interests: Data Science, Health Informatics, Energy Informatics

Master’s in Computer Science
University of Ottawa, Ottawa, Canada

Bachelor’s in Computer Engineering
Aligarh Muslim University, Aligarh, India
Dynamic Decision Making in Smart Grid

*dynamic* means decisions are made a few minutes to a few hours before they are to be implemented.

The Power to Decide

What's the point of all that data, anyway? It's to make decisions.
**What?**
Electric grid equipped with advanced technologies for
– monitoring
– control
– communication

**Why**
reliability
efficiency
sustainability
USC campus as a ‘smart’ microgrid

Motivation for our work
• Eliminate the need for manual intervention for demand optimization
• Enable automated decision making

Diversity
• Demographics
• Buildings (academic, admin, residential)

Scale
• 45K+ population
• ~50K sensors and smart meters
• 170 Buildings

Smart Equipment
• Measure energy usage at 1 min intervals
• Central control for zone temperatures and HVAC, VFD equipment, etc.
Data is collected from sensors & other sources in real-time (every 15 minutes or less).

- Presents an opportunity to mine this data for actionable insights.

**Big Data Sources**

**Weather**
- 24 readings per day

**Electricity**
- 170 meters in USC
- 50K in LA
- 96 readings in a day

**Ambient Temperature**
- 1000s sensors in USC
- 50K in LA
- 96 readings in a day

**Sensors**
- Occupancy, light, thermal, etc.
- 50K sensors in USC
- O(1mil) in LA
- 288 readings in a day

**Physical features**
- 170 buildings in USC
- 500K buildings in LA

**Social media**
- 50K people in USC
- 4 million people in LA

**Events**
- O(100) events per day in USC
- O(10K) events per day in LA

**Big Data Sources**
Demand Response (DR)

**DR Event**
- Utilities ask consumers to **decrease consumption** during anticipated peak demand periods.
- **Utilities** avoid the need to add additional generation units.
- **Consumers**: get incentives in return.

**Solution**: Make the demand adaptive to supply conditions.

This works for ‘anticipated’ peak periods. Need to address “un-anticipated” peak periods.
Planning for DR
[day ahead] vs [hours/minutes ahead]

Planning for DR involves:
• Consumption prediction
• Decision making about when, by how much, and how to reduce consumption
• Sending notification to the customers

Day ahead planning
Traditionally, planning for DR is done **one day** ahead of the DR day. *(Ziekow et. al., 2013)*

Hours/Minutes ahead planning
Needed due to dynamically changing conditions of the grid *(Simmhan et. al., 2013)*:
• Intermittent renewable energy sources
• Distributed energy sources
• Electric Vehicles
• Customer participation
• Special events

Factors driving the grid toward more dynamic operations
Dynamic demand response (D2R) is the process of balancing supply and demand in real-time and adapting to dynamically changing conditions by automating and transforming the demand response planning process. (Aman et al., 2015)

D2R is a prime example of dynamic decision making in smart grid.
Prediction Models Help Enable D²R

**Consumer Data**
- Web
- Mobile
- Social media
  - Consumption data
  - Consumer features

**Building Data**
- Static Data (physical features)
- Dynamic kWh and sensor data
- Space & schedule data
- Event info
- Weather data

**Reduced Consumption Prediction Model**

**Dynamic Demand Response (D²R) Policy Engine**
- Dynamic reduction signal
- Dynamic Consumer selection
- Reduction strategy selection
- Direct reduction signal

**Prediction Modeling (our focus)**

**Dynamic Decision Making**
Prediction Models for $D^2R$ Must Address Big Data Challenges

**Feature Selection**
- Relevant ones from large variety of features
- Parsimonious models preferred

**Data Collection**
- Effort required to acquire, assemble, and clean

**Computational Complexity**
- Time required in training and predictions is critical for dynamic predictions

**Veracity**
- Deal with imperfect data:
  - Missing data, partial data, etc.

**Value**
- Need to balance cost-benefit tradeoffs

5Vs of Big Data pose challenges for prediction.
Prediction models utilizing big data can enhance dynamic decision making in smart grids.
Prediction models utilizing big data can enhance dynamic decision making in smart grids.

Prediction models –
1) making predictions for the next few minutes to few hours horizon
2) evaluating prediction performance
Prediction models utilizing **big data** can enhance dynamic decision making in smart grids.

**big data** –
Using data from a variety of sources and addressing the challenges of 5 Vs.
Prediction models utilizing big data can enhance dynamic decision making in smart grids.

enhance –
our proposed prediction models help in some aspects of the decision making process, e.g., better accuracy with the available data, and faster decision making
Prediction models utilizing big data can enhance dynamic decision making in smart grids.

decision making –
when, by how much, and how to reduce electricity use by the demand side

dynamic –
decisions are made from a few minutes to a few hours ahead
Research Contributions

**Prediction with Partial Data**
- Unavailability of data from sensors in real time leads to partial data
- We propose a novel model to predict for all sensors using only partial real time data from some ‘influential’ sensors

**Prediction of Reduced Consumption**
- Identify challenges of consumption prediction under DR
- We propose a novel ensemble that models “mean behavior” and “context dependent behavior” to predict reduced consumption during DR

**Prediction Evaluation Measures**
- Identify limitations of existing measures
- Propose a suite of evaluation measures addressing the following:
  - Dimension, Prediction bias, Scale, Reliability, Cost, Application-relevance
Contribution 1

Prediction using Partial Data
Partial Data Problem

- Smart meters collect data in real-time (every 15 mins or less)
- Data is **not transmitted in real-time** to the utility, due to:
  - physical limitations of the transmission network (limited bandwidth)
  - security and privacy concerns of the consumers

- Only data from **some** meters (shown starred) is transmitted in real time.
- Complete data from **all** meters is available periodically when batch transmission takes place.

Only partial data is available in real-time.
Partial Data - Implications

Most prediction models are designed for ideal cases where all required data is readily available.

- Time-series models (e.g. ARIMA) and auto-regressive tree (ART) use recent real-time data. Without real-time data, the performance of these models deteriorates.

  For dynamic demand-response, real-time data is critical to predict peak demands.
# Partial Data Vs Missing Data

<table>
<thead>
<tr>
<th></th>
<th>Missing Data</th>
<th>Partial Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Timing</strong></td>
<td>Unavailability of data at <em>arbitrary</em> time periods</td>
<td>Systematic unavailability of data for <em>known</em> time periods</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>From <em>unknown</em> number of sensors</td>
<td>From a <em>known</em> subset of sensors</td>
</tr>
<tr>
<td><strong>Cause</strong></td>
<td>Due to <em>diverse factors</em>, such as faults</td>
<td>Due to <em>non-transmission of data</em> in that period</td>
</tr>
<tr>
<td><strong>Recovery</strong></td>
<td>Missing data is <em>lost</em></td>
<td>Partial data becomes <em>available</em> when batch transmission occurs, and can be used to re-train our models</td>
</tr>
<tr>
<td><strong>Related work</strong></td>
<td>Missing data is estimated by interpolation methods (<a href="#">Kreindler et. al., 2006</a>), (<a href="#">Cuevas-Tello et al., 2010</a>)</td>
<td>None for partial data</td>
</tr>
<tr>
<td></td>
<td>The volume of transmitted data is reduced by data compression (<a href="#">Marascu, 2013</a>) or data aggregation (<a href="#">Karimi et. al., 2013</a>)</td>
<td></td>
</tr>
</tbody>
</table>
Our approach – discovering ‘influential’ sensors

Instead of estimating unavailable real time data, we first discover influential sensors and use real time data only from them to do predictions for all sensors

We leverage the following:
• Fine grained data logged locally at sensors – available periodically at the utilities
• Real-time data – always available from some sensors

Hypothesis
Time series data of electricity consumption (and other schedule-driven data) shows dependencies
Our approach - Influence Model

Identify dependencies/influence between time series from recent historical data

Identify sensors that show a stronger ‘influence’ on other sensors using the Lasso Granger method

Train regression tree models using real-time data from influential sensors as features

We use Lasso-Granger as a novel way of feature selection for regression tree. (Arnold et. al., 2007)
We use Lasso Granger for Influence Discovery

- Given \( n \) sensor outputs in form of time series \( \mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n \)
- Each series has observations at timestamps \( t = 1, \ldots, T \)
- For each series \( \mathbf{x}^i \), a sparse solution for coefficients \( \mathbf{w} \) is obtained by minimizing the sum of squared error and a constant times the L1-norm of the coefficients:

\[
\mathbf{w} = \text{arg min} \sum_{t=l+1}^{T} \left\| \mathbf{x}^i_t - \sum_{j=1}^{n} \mathbf{w}_{i,j}^T \mathbf{P}^j_t \right\|^2 + \lambda \| \mathbf{w} \|_1
\]

where \( \mathbf{P}^j_t = [x^j_{t-l}, \ldots, x^j_{t-1}] \) is the sequence of past \( l \) readings

\( \mathbf{w}_{i,j} \) is the coefficient representing the dependency of series \( i \) on series \( j \)

\( \lambda \) is the parameter that determines the sparseness of the coefficient vector \( \mathbf{w}_i \)

*Lasso allows an efficient method for variable selection in high dimension* (Tibshirani '96), (Arnold et. al., 2007)
Influence Model (IM)

• We propose Influence Model to solve the partial data problem \((Aman \ et. \ al., \ 2015)\)
• Recent real-time values of other sensors are useful as predictors even in absence of the sensor’s own real-time values
• A sensor’s own relatively older values are less useful as predictors
**Baseline Models**

**Autoregressive Tree (ART)**
- ART($p$, $h$) – uses recent $p$ values of a variable as features in a regression tree for $h$ interval ahead prediction (*Meek et. al., 2002*)
- ART is a natural choice for baseline as it is also based on regression trees (like IM model)
- ART has been shown to offer high accuracy on large number of datasets (*Meek et. al., 2002*)

![Time series data from a sensor]

...  |
---   |
recent $p$ values  |
---   |
h-interval prediction horizon  |
Local Influence & Global Influence Models

LIM (Local Influence Model)

• Without restricting the number of influential sensors, the selected influential sensors may include the total number of sensors.
• In LIM, we ensure that only a fraction of sensors is selected (top $\mathcal{T}_l$ sensors selected locally).

GIM (Global Influence Model)

• Because local influencers are selected for each sensor, overall it may still result in a large number of sensors being selected
• In GIM, we select the top $\mathcal{T}_g$ sensors globally.
Results for IM

- Baseline (ART) performs well up to 6 intervals benefiting from real-time data.
- IM achieves comparable accuracy despite the lack of real-time data.
- IM's errors increase at a lower rate compared to ART.
- With time, a sensor's own data becomes stale, and more recent real-time values of other sensors become more useful predictors.

- 115 USC buildings
- 3 years’ data
- @15-min intervals
- 8 hour prediction horizon
Results for LIM and GIM

**LIM (vs IM)**
1.97% average increase for Top 8 and less than 1% increase for Top 12, 16, and 20 models.

**GIM (vs ART)**
Uses significantly lower number of sensors
Only ~0.5% increase in average error while using data from ~7% of sensors
Advantage of Influence Models

- ART requires real-time data from all sensors
- IM requires real-time data from only the influential sensors
- LIM requires real-time data from influential sensors selected locally for each sensor
- GIM uses real-time data from all influential sensors selected globally

Influential Models solve partial data problem in an efficient way without sacrificing accuracy.
Contribution 2
Prediction of Reduced Consumption
Reduced consumption prediction is useful in following decision-making tasks:

- estimating potential reduction during DR (Chelmis et. al., 2015)
- intelligently targeting customers for participation in DR (Ziekow et. al., 2015)
- performing dynamic DR at a few hours’ notice (Aman et. al., 2015)
- estimating the amount of incentives to be given to the customers (Wijaya et. al., 2014)
### Characteristics and Challenges

<table>
<thead>
<tr>
<th></th>
<th>Normal Consumption</th>
<th>DR Baseline</th>
<th>Reduced Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal</strong></td>
<td>Planning, DR</td>
<td>Curtailment calculation</td>
<td>Planning, DR, dynamic DR</td>
</tr>
<tr>
<td><strong>Timing</strong></td>
<td>Outside the DR event</td>
<td>Outside the DR event</td>
<td>During the DR event</td>
</tr>
<tr>
<td><strong>Historical data</strong></td>
<td>Readily available</td>
<td>Readily available</td>
<td>Sparse or non-existent</td>
</tr>
<tr>
<td><strong>Compute requirements</strong></td>
<td>Offline or real-time</td>
<td>Offline</td>
<td>Real-time for dynamic DR</td>
</tr>
<tr>
<td><strong>Profile changes</strong></td>
<td>Gradual</td>
<td>Gradual</td>
<td>Abrupt at DR event boundaries</td>
</tr>
<tr>
<td><strong>Prior Work</strong></td>
<td>Several</td>
<td>Several</td>
<td>None</td>
</tr>
</tbody>
</table>

We are the first to address this problem using data from DR experiments done on USC campus. *(Aman et. al., 2016), (Chelmis et. al., 2015)*
Key Challenges

- Unavailability of reduced consumption data
- Cancellation of DR event when found violating thermal comfort limits of occupants.
- Reduced consumption is affected by several factors:
  - time of day/ day of week
  - reduction strategy
  - human behavior
  - external/environmental factors, e.g., temperature

- Time series models that work well for normal consumption prediction are ineffective for reduced consumption prediction, due to
  - abrupt changes in consumption profile at the beginning and end of the DR event
  - insufficient recent observations within the DR window for a time series model to be trained reliably

Hypothesis
Historical data from the past DR events can be used as predictors for reduced consumption.
Consumption Sequences

**Daily sequence** \( \mathcal{E}_i = \{e_{i,1}, e_{i,2}, \ldots, e_{i,J}\} \)

- Electricity consumed on day \( i \) in interval \( j \)
- Subsequence of daily sequence \( \mathcal{E}_i \) starting at \( s \) of length \( l \)
- Length of the DR interval
- The interval when DR begins
- Number of intervals in a day

\[ \begin{align*}
e_{i,j} & \quad \text{Electricity consumed on day } i \text{ in interval } j \\
\mathcal{E}_{i,s,l} & \quad \text{Subsequence of daily sequence } \mathcal{E}_i \text{ starting at } s \text{ of length } l \\
L & \quad \text{Length of the DR interval} \\
d & \quad \text{The interval when DR begins} \\
J & \quad \text{Number of intervals in a day}
\end{align*} \]

\[d > 1\]
\[d + L - 1 \leq J\]
Contextual Attributes

- **Time Series attributes**: vary over intervals  
  - temperature, dynamic pricing, occupancy, etc.

- **Static attributes**: same for all intervals  
  - day of week, holiday, etc.

Daily Context

\[ C_i = \langle A_i[1], ..., A_i[N_t], B_i[1], ..., B_i[N_s] \rangle \]

\[ A_i[k] = \{ a_{i,1}, a_{i,2}, ..., a_{i,J} \} \]

Pre-DR Context

\[ C_{i,1,d-1} = \langle A_i[1], ..., A_i[N_t], B_i[1], ..., B_i[N_s] \rangle \]

\[ A_i[k] = \{ a_{i,1}, a_{i,2}, ..., a_{i,d-1} \} \]

Correspond to the Daily Sequence and Pre-DR Sequence defined previously.
REDUCE – Reduced Consumption Ensemble

IDS
In-DR Sequence Model

\[ \hat{E}_{\epsilon,d,L} IDS \]

PDS
Pre-DR Sequence Similarity Model

\[ \hat{E}_{\epsilon,d,L} PDS \]

DSS
Daily Sequence Similarity Model

\[ \hat{E}_{\epsilon,d,L} DSS \]

Random Forest Model

Final Output

- \[ \hat{E}_{\epsilon,d,L} m \] – In-DR sequence predicted by model \( m \) on day \( \epsilon \)
- Ensemble Models combine base models that model different behaviors, for e.g., mean behavior, context dependent behavior, etc.
- Random Forest Models are found to perform better than a single regression tree (Breiman, 2001)
IDS – In-DR Sequence Model

• Models “mean behavior”
• Similar to the averaging approach used by the utilities/ISOs to calculate the DR baseline.
• While utilities average over similar non-DR days, IDS averages over all DR days.

• Advantages:
  • Low computation cost – suitable for real-time predictions
  • Uni-variate model – low data collection cost

• Predicted sequence is given by:

\[
[\hat{E}_{i,d,L}]_{IDS} = \frac{1}{|\mathcal{E}|} \sum_{e=1}^{|\mathcal{E}|} \mathcal{E}_{e,d,L}
\]

\( \mathcal{E} \) is the set of historical DR days
PDS – Pre-DR Sequence Similarity Models

If two DR days have similar pre-DR sequences, their in-DR sequences would be similar.

- Pre-DR sequence
- Pre-DR context

Used to select similar DR days

Similarity is calculated by:

$$SimScore(\epsilon, i) = \text{sim}(\langle \epsilon_{\epsilon,1,d-1}, C_{\epsilon,1,d-1} \rangle, \langle \epsilon_{i,1,d-1}, C_{i,1,d-1} \rangle)$$

- Selected days are sorted based on decreasing similarity and weighed accordingly.
- Predicted sequence is given by:

$$[\hat{E}_{i,d,L}]_{PDS} = \frac{1}{|\mathcal{E}|} \sum_{\epsilon=1}^{||\mathcal{E}||} \omega_\epsilon \times \mathcal{E}_{\epsilon,d,L}$$

$\mathcal{E}$ is the set of historical DR days

$\omega_\epsilon$ is the weight on day $\epsilon$

PDS models context dependent behavior
DSS – Daily Sequence Similarity Models

• Daily sequence
• Daily context

} Used to discover clusters of daily profiles

• Form daily profiles for each day
  \( \mathcal{P}_e = \langle \mathcal{E}_e, \mathcal{C}_e \rangle \)

• Cluster daily profiles and let \( c_m \) be the centroid of each cluster

• Probability of a given DR day belonging to a cluster is given by:

\[
P(i \in C_m) = \frac{1}{\alpha \left\| \mathcal{P}_{i,1,d-1} - \mathcal{P}_{c_m,1,d-1} \right\|_2}
\]

\( \alpha \) is constant used to normalize the probability values between 0 and 1

• Predicted sequence is given by:

\[
[\hat{\mathcal{E}}_{i,d,L}]_{DSS} = \frac{1}{N_k} \sum_{m=1}^{N_k} P(i \in C_m) \times \mathcal{E}_{c_m,d,L}
\]
Results

- REDUCE outperforms the baseline IDS for about 70% of the buildings
- It also limits prediction error to <10% for over half the buildings
  – considered highly reliable by domain experts (Aman et. al., 2015)
- Overall average error is 13.5%, an improvement of 8.8% over the baseline

- 952 DR events (2012 – 2014)
- 32 USC buildings
- Contextual attributes:
  - temperature (NOAA)
  - day of week
Scheduled Vs. Non-scheduled

- Scheduled – activities governed by schedules, for e.g., classrooms
- For non-scheduled:
  - REDUCE gives superior performance
  - IDS does not perform well due to the absence of repetitive human activity coupled to class schedules

MAPE errors for Non-scheduled building (large computer labs, faculty and student offices)

- Corollary: REDUCE would perform better for residential buildings (with non-scheduled activities).
Effect of Training Data

- REDUCE is not sensitive to the training data size -> reduces computational and storage requirements
- **Corollary:** REDUCE would allow accurate predictions to be made for new buildings with fewer historical data
• For REDUCE, error decreases with increasing average consumption
  - more stable and predictable behavior for larger buildings
• Insight: The performance of REDUCE slightly improves for larger buildings
Contribution 3
Prediction Evaluation Measures
Common Evaluation Measures

\( o_i \) – observed value at interval \( i \)
\( p_i \) – predicted value at interval \( i \)
\( n \) – number of intervals for which prediction is made

**Mean Absolute Percentage Error (MAPE)**
Mean absolute Error (MAE) normalized by the observed value.

\[
MAPE = \frac{1}{n} \sum_{i=1}^{n} \frac{|p_i - o_i|}{o_i}
\]

**Coefficient of Variation of Root Mean Square Error (CV-RMSE)**
Root Mean Square Error (RMSE) normalized by the mean of observed values

\[
CVRMSE = \frac{1}{\bar{o}} \sqrt{\frac{1}{n} \sum_{i=1}^{n} (p_i - o_i)^2}
\]
## Limitations of Current Evaluation Approaches

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Uni-dimensional focus on error measures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prediction Bias</strong></td>
<td>Insensitive to prediction bias</td>
</tr>
<tr>
<td></td>
<td>Under-prediction is deleterious for estimating peak demand</td>
</tr>
<tr>
<td><strong>Scale</strong></td>
<td>Scale-dependent measures are unsuitable for comparing customers of different sizes</td>
</tr>
<tr>
<td><strong>Reliability</strong></td>
<td>Don’t consider the frequency with which a model does good predictions:</td>
</tr>
<tr>
<td></td>
<td>– # times a model outperforms the baseline</td>
</tr>
<tr>
<td></td>
<td>– # times a model’s error is within a tolerance level</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>Don’t consider the costs</td>
</tr>
<tr>
<td></td>
<td>– data collection</td>
</tr>
<tr>
<td></td>
<td>– building models and running them</td>
</tr>
<tr>
<td><strong>Application-relevance</strong></td>
<td>Based on “abstract metrics”; not relevant to the end application</td>
</tr>
</tbody>
</table>

We propose holistic evaluation measures to address these limitations.
'Application-specific' Bias based Measure

\( o_i \) – observed value at interval \( i \)
\( p_i \) – predicted value at interval \( i \)
\( n \) – number of intervals for which prediction is made
\( \alpha, \beta \) – penalty parameters associated with over- and under- predictions

Domain Bias Percentage Error (DBPE)

\[
DBPE = \frac{1}{n} \sum_{i=1}^{n} \frac{\mathcal{L}(p_i, o_i)}{o_i}
\]

\[
\mathcal{L}(p_i, o_i) = \begin{cases} 
\alpha \cdot |p_i - o_i|, & \text{if } p_i > o_i \\
0, & \text{if } p_i = o_i \\
\beta \cdot |p_i - o_i|, & \text{if } p_i < o_i 
\end{cases}
\]

Penalty parameters are configured for specific applications in consultation with the domain experts.
Reliability Measure

\( o_i \) – observed value at interval \( i \)

\( p_i \) – candidate model predicted value at interval \( i \)

\( b_i \) – baseline model predicted value at interval \( i \)

\( n \) – number of intervals for which prediction is made

Relative Improvement (RIM)
Fraction of predictions made by a candidate model better than the baseline model.

\[
RIM = \frac{1}{n} \sum_{i=1}^{n} C(p_i, o_i, b_i)
\]

\( C(p_i, o_i, b_i) = \begin{cases} 
1, & \text{if } |p_i - o_i| < |b_i - o_i| \\
0, & \text{if } |p_i - o_i| = |b_i - o_i| \\
-1, & \text{if } |p_i - o_i| > |b_i - o_i| 
\end{cases} \)

Candidate model performs better than the baseline
‘Application-specific’ Reliability Measure

$O_i$ – observed value at interval $i$

$P_i$ – candidate model predicted value at interval $i$

$e_t$ – error threshold

$n$ – number of intervals for which prediction is made

**Reliability Threshold Estimate (REL)**

Measures how frequently the errors fall within a set threshold

$$REL = \frac{1}{n} \sum_{i=1}^{n} C(p_i, o_i)$$

$$C(p_i, o_i) = \begin{cases} 
1, & \text{if } \frac{|p_i - o_i|}{o_i} < e_t \\
0, & \text{if } \frac{|p_i - o_i|}{o_i} = e_t \\
-1, & \text{if } \frac{|p_i - o_i|}{o_i} > e_t 
\end{cases}$$

The error threshold is set for specific applications in consultation with the domain experts.
Cost Measures

Computation Cost (CC)

\[ CC = CC_t + CC_p \]

- \( CC_t \) – time taken to train a model
- \( CC_p \) – time taken to predict using the trained model

Total Computation Cost (TCC)

\[ TCC = CC_t \cdot \tau + CC_p \cdot \pi \]

- \( \tau \) – number of times a model is trained in a duration of interest
- \( \pi \) – number of times a model makes prediction in that duration
For prediction models using “big data”, it is critical to consider the cost of building and using a model relative to the gain it provides.

**Cost-Benefit Measure (CBM)**

\[ CBM = \frac{(1 - DBPE)}{TCC} \]

A model with high accuracy but with prohibitive cost may be unsuitable.
Results – Bias based DBPE measure

To avoid missing peaks, we favor over-predictions to under-predictions. We set $\alpha = 0.5$ and $\beta = 1.5$ for DBPE.

DBPE is uniformly smaller than MAPE. CBM is lower for Regression Tree model due to high data collection cost for different features.
Conclusion
Conclusion

• We proposed **Dynamic Demand Response (D²R)**
  – as a novel extension of the state-of-the-art DR practice in smart grids
  – as a prime example of **dynamic decision making** in smart grids

• We made following contributions:
  • Proposed novel model for prediction with partial data
  • Proposed novel ensemble model for prediction of reduced consumption
  • Proposed holistic measures to evaluate prediction performance

• Our proposed models are being used (or in process of deployment) at the USC Facilities Management and will eventually be used at the LA Department of Water and Power (LADWP).
Publications

• Aman et. al., Holistic Measures for Evaluating Prediction Models in Smart Grids, Transactions in Knowledge and Data Engineering, 27(2), 2015.

• Aman et. al., Influence-driven Model for Time Series Prediction from Partial Observations, AAAI 2015.

• Aman et. al., Learning to REDUCE: A Reduced Electricity Consumption Prediction Ensemble, AAAI Workshop, 2016.


• Aman, Analytics for Demand Response Optimization in a MicroGrid, SDM Doctoral Forum, 2012

• Aman et. al., Improving Energy Use Forecast for Campus Micro-grids using Indirect Indicators, ICDM International Workshop on Domain Driven Data Mining, 2011.


• Chelmis et. al., Estimating Reduced Consumption for Dynamic Demand Response, AAAI Workshop on Computational Sustainability, 2015.


References


• Meek et. al., Autoregressive tree models for time-series analysis, SIAM Conference on Data Mining (SDM) 2002.


• Arnold et. al., Temporal causal modeling with graphical granger methods. In International conference on Knowledge discovery and data mining (KDD ’07)


Thank you!