Staring at the Sun: Black-box Solar Analytics and their Privacy Implications

David Irwin
Electrical and Computer Engineering
University of Massachusetts Amherst
Solar Energy is Rapidly Expanding

- Installed cost of solar continues to drop
  - Cost fell by 50% from 2008 to 2013
  - Led to 418% increase in solar capacity

- Many implications to this rising solar penetration
Implications of Solar to Grid

- Utilities must actively control **generation** to balance grid
  - **Individual** homes exhibit highly stochastic demand profiles
  - However, **aggregate** demand profiles are smooth and highly predictable

- Enables utilities to plan generator “dispatch” schedules in advance
Implications of Solar to Grid

• Large-scale solar penetration fundamentally alters this paradigm
  - **Increases stochasticity** of demand profiles, even when aggregated
  - Solar output **can change instantly**, while generators take time to “ramp up”
  - **Complicates controlling generation to balance supply and demand**
    ‣ May require more energy storage, spinning reserve, or demand response capacity

• **Accurate** solar monitoring and forecasting is **critical**
  - Track solar penetration rates over time
  - Monitor real-time fluctuations in grid solar production
  - Inform advanced planning of generator dispatch schedules
  - Identify faults and anomalies in solar output
Prior Work

- Possible to develop highly accurate models of solar performance
  - Leverages detailed information on site characteristics

Figure from PV Performance Modeling Collaborative

- However, detailed information not always available
Black-box Solar Analytics

• Assumes only access to **solar energy data time-series**
  - Without any detailed metadata

• **Motivating Scenarios**
  - Utilities managing grid with thousands of small-scale solar sites
    ‣ Might know location, but not deployment details
  - Third-party energy analytics companies
    ‣ Often do not know location, or deployment details
  - Researchers accessing public datasets
    ‣ Metadata is often scarce and unreliable
This Talk – Discuss Two Black-box Techniques

1. Solar Disaggregation
   - Turns out utilities often do not even have access to solar data
   - Residential “grid-tied” solar almost always “behind the meter”
     - Only directly monitor the net of consumption and generation
   - Prevents wide-range of learning-based data analytics
This Talk – Discuss Two Black-box Techniques

• 2. Solar Localization
  - Determine location from “anonymous” solar energy data
  - Both a privacy threat and/or a potentially useful tool
    ▪ Location is highly useful contextual information when analyzing energy data
SunDance - Solar Disaggregation

- Given meter location, separate “net” meter data into solar generation and consumption at each time $t$
  - $P_{\text{net}}(t) = P_s(t) + P_c(t)$, where $P_c(t) > 0$, $P_s(t) < 0$, $\forall \ t > 0$

- **Challenges**
  - 1. Do not have access to already-separated historical data
  - 2. Cannot individually model solar generation or energy consumption
SunDance Design Overview

1. Build a custom model of maximum solar generation
   - Find “best” fitting valid solar curve to the data using a small amount of data
   - *Can find accurately even on noisy net meter data*

2. Build a general model of weather’s effect on irradiance
   - Train model that maps weather metrics to fraction of clear sky irradiance
   - Use to infer fraction of clear sky irradiance at site based on weather
   - *Can train model using data from any solar sites where it is available*

3. Apply two models to disaggregate solar
   - Solar generation $P_s(t) = \text{Product of (1) and (2) at every time } t$
   - Energy consumption $P_c(t) = P_{\text{net}}(t) - P_s(t)$
SunDance Design Overview

• **1. Build a custom model of maximum solar generation**
  - Find “best” fitting valid solar curve to the data using a small amount of data
  - *Can find accurately even on noisy net meter data*

• **2. Build a general model of weather’s effect on irradiance**
  - Train model that maps weather metrics to fraction of clear sky irradiance
  - Use to infer fraction of clear sky irradiance at any site based on weather
  - *Can train model using data from any solar sites where it is available*

• **3. Apply two models to disaggregate solar**
  - Solar generation $P_s(t) = \text{Product of (1) and (2) at every time } t$
  - Energy consumption $P_c(t) = P_{\text{net}}(t) - P_s(t)$
Apply Physical Solar Performance Model

- Use *clear sky model* to compute maximum irradiance
- Search for *size, efficiency, tilt, and orientation* that yields the tightest upper bound on the data

- Apply *linear temperature adjustment* to data
  - Find linear constant $c$ (~0.4%/C) that yields the tightest upper bound
Modeling Net Meter Data

- Issues with modeling “noisy” net energy meter data
  - **Power Consumption Floor** – do not know zero point of solar

- SunDance estimates based on **minimum power consumption at night**, where solar power is known to be zero, to adjust the model
Issues with modeling “noisy” net energy meter data

- Consumption “Noise” – reduces solar generation like weather

- SunDance robust as long as at least one datapoint exists where solar generation is near its maximum potential and energy consumption is low
Issues with modeling “noisy” net energy meter data

- Consumption “Noise” – reduces solar generation like weather

- SunDance robust as long as at least one datapoint exists where solar generation is near its maximum potential and energy consumption is low
Can build **highly accurate** and **custom** maximum generation models with a minimal amount of net meter data

- In the limit, we only need the “right” **two datapoints**
  - Solar generation is near maximum, energy consumption is near minimum
  - There is a significant temperature difference
- Accuracy changes little when using pure solar or net meter data

<table>
<thead>
<tr>
<th></th>
<th>Days</th>
<th>Tilt</th>
<th>Orientation</th>
<th>$k$</th>
<th>$\text{Area (m}^2\text{)}$</th>
<th>$c$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ground Truth</strong></td>
<td>NA</td>
<td>35°</td>
<td>190°</td>
<td>12.3</td>
<td>48.88</td>
<td>NA</td>
</tr>
<tr>
<td><strong>Pure Solar</strong></td>
<td>365</td>
<td>36°</td>
<td>189°</td>
<td>10.6</td>
<td>48.18</td>
<td>0.57</td>
</tr>
<tr>
<td><strong>Net Meter (Temp)</strong></td>
<td>365</td>
<td>34°</td>
<td>186°</td>
<td>10.9</td>
<td>49.55</td>
<td>0.72</td>
</tr>
<tr>
<td><strong>Net Meter (Temp)</strong></td>
<td>2</td>
<td>36°</td>
<td>185°</td>
<td>11.6</td>
<td>52.73</td>
<td>0.69</td>
</tr>
</tbody>
</table>
Modeling Net Meter Data

- **Verify using data from 10 more solar sites**
  - Manually measure module tilt and orientation
  - Find values close to ground-truth using minimal data

- **Tilt slightly less accurate** – difficult to distinguish between different tilts and different module areas/efficiencies
SunDance Design Overview

• 1. Build a custom model of maximum solar generation
  - Find “best” fitting valid solar curve to the data
  - *Can find accurately even on noisy net meter data*

• 2. Build a general model of weather’s effect on irradiance
  - Train model that maps weather metrics to fraction of clear sky irradiance
  - Use to infer fraction of clear sky irradiance at any site based on weather
  - *Can train model using data from any solar sites where it is available*

• 3. Apply two models to disaggregate solar
  - Solar generation $P_s(t) = \text{Product of (1) and (2) at every time } t$
  - Energy consumption $P_c(t) = P_{\text{net}}(t) - P_s(t)$
Building a General Weather Model

• **Weather effects** – blocks solar irradiance from reaching module
  - Primarily clouds, but also humidity, dewpoint, wind, etc. increase particulates
  - *Weather’s impact on blocking solar irradiance is complex*

<table>
<thead>
<tr>
<th>Clear</th>
<th>Light Clouds</th>
<th>Overcast</th>
</tr>
</thead>
</table>

- Data above from three locations at different times for three different weather conditions **varies widely**
Building a General Weather Model

- **Key insight** – same weather reduces the maximum solar irradiance by the same fraction
  - Independent of location, time, magnitude, etc.

- Can infer weather’s effect at one location from others

![Graph showing percentage max over sample hour-long periods for clear, light clouds, and overcast conditions.]
Building a General Weather Model

• Build using supervised machine learning
  - **Input** – weather metrics, e.g., cloud cover, humidity, dew point, etc.
  - **Output** – the fraction of the clear sky solar generation
  - Construct **single** training set using data from many different solar sites where “pure” solar training data is available

- Use SVMs, but compatible with any machine learning model
Building a General Weather Model

- **Key insight** – we can use our maximum generation model to infer fraction of solar irradiance from solar power data
  - More solar power data available than pyranometer data
  - Physical effects “cancel out” in the equation below

\[
\begin{align*}
P_{\text{actual}}(t) &= I_{\text{actual}}(t) \times k \times (\cos(90-\Theta) \times \sin(\beta) \times \cos(\Phi-\alpha) + \sin(90-\Theta) \times \cos(\beta)) \\
P_{\text{smax}}(t) &= I_{\text{max}}(t) \times k \times (\cos(90-\Theta) \times \sin(\beta) \times \cos(\Phi-\alpha) + \sin(90-\Theta) \times \cos(\beta))
\end{align*}
\]

- Ratio above is called the **clear sky index**
  - Widely used in solar forecasting
  - Enables us to estimate a clear sky index from solar power data
    - Potentially useful for estimating ground-level irradiance
SunDance Design Overview

1. Build a custom model of maximum solar generation
   - Find “best” fitting valid solar curve to the data
   - *Can find accurately even on noisy net meter data*

2. Build a general model of weather’s effect on irradiance
   - Train model that maps weather metrics to fraction of clear sky irradiance
   - Use to infer fraction of clear sky irradiance at any site based on weather
   - *Can train model using data from any solar sites where it is available*

3. Apply two models to disaggregate solar
   - Solar generation \( P_s(t) = \text{Product of (1) and (2) at every time } t \)
   - Energy consumption \( P_c(t) = P_{\text{net}(t)} - P_s(t) \)
Implementation and Evaluation

• Implement SunDance in python
  - Use simple Bird clear sky generation model, weather data from Weather Underground, and Scikit-learn machine learning library

• Compare SunDance with a supervised approach for 100 sites
  - Supervised approach uses exact same method as SunDance
  - **Supervised approach** - builds and trains each site’s maximum generation model and weather model on historical solar generation data from that site
  - **SunDance** – assumes no access to already disaggregated training data
  - Use MAPE from sunrise to sunset as the evaluation metric

\[
MAPE = \frac{100}{n} \sum_{t=0}^{n} \left| \frac{S_t - P_t}{S_t} \right|
\]
Disaggregating Solar

- Illustration on a net-zero home
  - Top – net meter data
  - Bottom – disaggregated solar

MAPE: 26.178426732

SunDance
Ground truth
Evaluation

- Accuracy close to that of the supervised approach
  - Improves as consumption-to-solar ratio decreases

- MAPE highly sensitive to small errors near sunrise/sunset
  - SunDance very close to supervised approach over middle of the day
Summary

• SunDance - “Behind the Meter” Solar Disaggregation
  - Leverages multiple insights into fundamental relationships between location, weather, physical characteristics, and solar generation
  - Achieves similar accuracy without access to solar training data as a fully supervised approach with complete access to solar training data
  - Requires little historical net meter data to build model

• Enables utilities to accurately monitor solar data
  - Provides training data to support other solar energy analytics
This Talk – Discuss Two Black-box Techniques

• 2. Solar Localization
  - Determine location from “anonymous” solar energy data
  - Both a privacy threat and/or a potentially useful tool
    ▶ Location is highly useful contextual information when analyzing energy data
Privacy Implications

• Energy data routinely **monitored by third-parties**, including…
  - …utilities, solar installers, researchers, governments, etc.
  - Not treated as sensitive if “anonymized”
  - Found ~28k “anonymous” homes making data available over public Internet
Exploiting Energy Data using Analytics

• Many companies actively working to develop energy data analytics
  - Identify energy waste to improve energy-efficiency
  - May also provide deep insights into user behavior
    ▷ What are a home’s occupancy patterns?
    ▷ How often do occupants go out for vacations?
    ▷ How often do occupants eat-in versus go out to eat?

• Privacy implications are less concerning for anonymized data
  - Cannot associate behaviors with specific people
This is Real

- **Real public job advertisement for an energy analytics startup**

- **Identify User Behavior.** Parameterize each appliance use based on user lifestyle and consumption and help them identify where to target energy reduction.

- **Identify Appliance Brands.** Use energy data to predict whether the user has GE or Maytag refrigerator. Very cool! Imagine the value of that information for Whirlpool to target this house for selling their appliance.

- A sample of projects that we are working on and problems that we are trying to solve:
  - **Appliance Energy Signatures.** Its the pattern recognition algorithms applied to the energy consumption signatures of various appliances to extract them from the whole house energy profile from Smart Meters.
  - **Identify User Behavior.** Parameterize each appliance use based on user lifestyle and consumption and help them identify where to target energy reduction.
  - **Identify Appliance Brands.** Use energy data to predict whether the user has GE or Maytag refrigerator. Very cool! Imagine the value of that information for Whirlpool to target this house for selling their appliance.
  - **Handling Big Data.** With live data from millions of homes, imagine how interesting it would be to be able to predict the clustering of appliances based on model, year, geography, efficiency and user behavior.
Exploiting Energy Data using Analytics

- Policies for handling energy data are still evolving
  - DOE’s Data Privacy and the Smart Grid: A Voluntary Code of Conduct
  - Finalized on January 8th, 2015
  - **Does not require** user consent to release “anonymized” energy data
    - Defined as user account information: name, address, SSN, etc.

(4) Aggregated or Anonymized Data Service Providers can share Aggregated or Anonymized data with Third Parties without first obtaining customer consent if the methodology used to aggregate or anonymize Customer Data strongly limits the likelihood of reidentification of individual customers or their Customer Data from the aggregated or Anonymized data set.
Key Insight

- **Solar energy data is not anonymous**
  - Every location on Earth has a unique *solar signature*.
  - Sun’s position in the sky is unique at each location at every moment.
    - E.g., unique sunrise, sunset, and solar noon time each day.

- Solar data embeds detailed location information.
Problem

• **How to localize the source of anonymous solar data?**
  - Explore severity and threat of solar localization
    ‣ Depending on perspective, could also be a useful tool
  - Significant prior work on estimating solar output based on location
  - No work on estimating location based on solar output

• **SunSpot** – system for localizing anonymous solar-powered homes based on their solar energy data
  - Inform evolving policies on handling energy data that includes solar
  - Reconsider current notions of anonymity in energy data
Basic Approach

• Location uniquely identified by a **latitude** and **longitude**
  - **Latitude** – uniquely identified by the **daylength [sunrise->sunset]**
    › Duration from first to last times of positive solar generation
  - **Longitude** – uniquely identified by **time of solar noon**
    › Maximum solar generation
Basic Approach

- Location uniquely identified by a **latitude** and **longitude**
  - **Latitude** – uniquely identified by the **daylength [sunrise->sunset]**
    - Duration from first to last times of positive solar generation
  - **Longitude** – uniquely identified by **time of solar noon**
    - Maximum solar generation
Basic Approach

- Location uniquely identified by a **latitude** and **longitude**
  - **Latitude** – uniquely identified by the **daylength [sunrise->sunset]**
    - Duration from first to last times of positive solar generation
  - **Longitude** – uniquely identified by **time of solar noon**
    - Maximum solar generation
Deriving Location from the Sun

• Algorithms for deriving location from the sun are **obscure**
  - Typically used for celestial navigation of primitive ships
  - No widely-used open-source libraries or online APIs

• Algorithms for deriving sunrise/sunset for location are **common**
  - Highly accurate but not easily reversible
  - Many open-source libraries and online APIs available

• Leverage existing APIs as a sub-routine to conduct a **binary search for location** given sunrise/sunset times
  - (sunrise, sunset) == (daylength, solar noon)
Deriving Latitude given Daylength

• Note that….
  - …in winter, daylength decreases moving south to north
Deriving Latitude given Daylength

- Note that:
  - ...in winter, daylength **decreases** moving south to north
  - ...in summer, daylength **increases** moving south to north
Deriving Latitude given Daylength

• Note that….
  - …in winter, daylength decreases moving south to north
  - …in summer, daylength increases moving south to north

• Binary Search using API
Deriving Latitude given Daylength

• Note that….
  - …in winter, daylength decreases moving south to north
  - …in summer, daylength increases moving south to north

• Binary Search using API
Deriving Latitude given Daylength

• Note that….
  - …in winter, daylength **decreases** moving south to north
  - …in summer, daylength **increases** moving south to north

• Binary Search using API
Deriving Latitude given Daylength

• Note that….
  - …in winter, daylength decreases moving south to north
  - …in summer, daylength increases moving south to north

• Binary Search using API

![Binary Search Diagram]
Deriving Latitude given Daylength

• Note that....
  - …in winter, daylength decreases moving south to north
  - …in summer, daylength increases moving south to north

• Binary Search using API
Deriving Latitude given Daylength

• Note that….
  - …in winter, daylength decreases moving south to north
  - …in summer, daylength increases moving south to north

• Binary Search using API
  - Accuracy below on June 21st (summer solstice)
Deriving Latitude given Daylength

- Note that....
  - ...in winter, daylength **decreases** moving south to north
  - ...in summer, daylength **increases** moving south to north

- Binary Search using API
Deriving Longitude given Time of Solar Noon

- Binary Search using API
  - Use API to compute solar noon for 0° and ±180°
    - Pick any latitude value
  - Select region with desired solar noon time
    - Either [0°,180°] or [0°,-180°]
  - Divide selected interval by two ([0°,90°], [0°,-90°]) and repeat…
    - …until longitude does not change
Deriving Longitude given Time of Solar Noon

- Binary Search using API
  - Use API to compute solar noon for 0° and ±180°
    - Pick any latitude value
  - Select region with desired solar noon time
    - Either [0°,180°] or [0°,-180°]
  - Divide selected interval by two ([0°,90°], [0°,-90°]) and repeat…
    - …until longitude does not change
SunSpot Challenge

- Ideally, take solar generation from one day
  - Extract precise sunrise, sunset, and solar noon time (to the second)
  - Directly compute latitude and longitude accurately
  - But, solar cells are highly imprecise sensors of the sun
    - Error translates to hundreds-to-thousands of miles
Solar Imprecision and Inefficiency

• Many dimensions of imprecision
  - **Solar cell inefficiency** – sunrise/sunset detection lag
  - **Variable weather** – may be cloudy at sunrise/sunset/solar noon
  - **Shading from obstructions** – nearby buildings, trees
  - **Non-optimal physical properties** – tilt/orientation
  - **Non-optimal electrical characteristics** – variations in grid voltage
  - **Meter inaccuracy** – typically 0.5% to 2% off

• Accurate localization challenging using one day’s data
  - Impossible if day is near the equinox
  - SunSpot **leverages data across multiple days**
Inferring Longitude from Noisy Solar Data

- Day-to-day changes in solar noon over the year are the same at **every location on Earth**
  - 31 minutes of movement captured by the **Equation of Time** (EoT)
  - Solar noon **should** precisely track the EoT
Inferring Longitude from Noisy Solar Data

- Day-to-day changes in solar noon over the year are the same at every location on Earth
  - 31 minutes of movement captured by the Equation of Time (EoT)
  - Solar noon should precisely track the EoT
Day-to-day changes in solar noon over the year are the same at every location on Earth.

- To “fit” EoT, we move it up and down the y-axis.
- Stop where it overlaps the most absolute data points (within ±1m).
Inferring Longitude from Noisy Solar Data

- Day-to-day changes in solar noon over the year are the same at every location on Earth
  - To “fit” EoT, we move it up and down the y-axis
  - Stop where it **overlaps** the most absolute data points (within ±1m)
Inferring Latitude from Noisy Solar Data

- **Problem:** sunrise/sunset always lags solar data detection
  - Again, recall that daylength varies with latitude
    - …in fall/winter, daylength shorter moving south to north
    - …in spring/summer, daylength longer moving south to north
  - In fall/winter, always infer location **north** of actual location
  - In spring/summer, always infer location **south** of actual location
Inferring Latitude from Noisy Solar Data

- **Problem:** sunrise/sunset always lags solar data detection
  - Again, recall that daylength varies with latitude
    - …in fall/winter, daylength shorter moving south to north
    - …in spring/summer, daylength longer moving south to north
  - **In fall/winter, always infer location north of actual location**
  - In spring/summer, always infer location south of actual location
Inferring Latitude from Noisy Solar Data

- **Problem:** sunrise/sunset always lags solar data detection
  - Again, recall that daylength varies with latitude
    - …in fall/winter, daylength shorter moving south to north
    - …in spring/summer, daylength longer moving south to north
  - In fall/winter, always infer location north of actual location
  - In spring/summer, always infer location south of actual location

![Graph showing daylength variations over the year](image-url)
Inferring Latitude from Noisy Solar Data

- **Problem:** sunrise/sunset always lags solar data detection
  - Again, recall that daylength varies with latitude
    - ...in fall/winter, daylength shorter moving south to north
    - ...in spring/summer, daylength longer moving south to north
  - In fall/winter, always infer location **north** of actual location
  - In spring/summer, always infer location **south** of actual location

![Graph showing daylength over the course of the year](image.png)
Inferring Latitude from Noisy Solar Data

- **Problem**: sunrise/sunset always lags solar data detection
  - Again, recall that daylength varies with latitude
    - …in fall/winter, daylength shorter moving south to north
    - …in spring/summer, daylength longer moving south to north
  - In fall/winter, always infer location **north** of actual location
  - In spring/summer, always infer location **south** of actual location
Localizing a Specific Home

• Previous steps identify only a **region of interest**
  - Limited by data resolution, and other inaccuracies
  - Search **satellite imagery** for solar-powered homes within region
    ▶ Filter out land area without man-made structures (>97%)
    ▶ Apply image recognition (either manually or algorithmically)
    ▶ Filter identified solar sites by size of deployment, physical properties, etc.
Implementation

- SunSpot implemented in python
  - Uses available online APIs for computing sunrise/sunset for locations
    - For latitude, use to derive daylength curves
    - For longitude, use to derive solar noon time
  - Uses public satellite imagery from Google Earth
    - Leverage Google Maps API to extract images with man-made structures
    - Apply OpenCV to remove images without >5% black pixels
  - Automatically identify solar sites from images
    - Feed images to Mechanical Turk to identify solar sites
    - Could also include adjustments for non-south facing sites
Evaluation

- Amazon **Mechanical Turk**
  - A crowdsourcing Internet marketplace
  - Leverages humans to perform routine tasks

Task: Is there a solar panel in the image?

Yes.                                                        No.
Evaluation

- Three homes with per-second data resolution
  - Maximum localization precision ~500m
  - Inaccuracy ranges from 10-20km
Evaluation

- Microbenchmarks of image processing using **Mechanical Turk**
  - Took random urban area with 2km radius (or 12.6km²)
    - 82% covered with man-made structures
  - Extract and filtered satellite images from Google Earth
    - **Ground truth** - manually checked these images for visible solar sites
  - Programmatically submitted images to Mechanical Turk
    - 99% categorized within 30m, with average time ~42 seconds
    - 93% accurate - identified all but 2 solar sites we identified manually
  - **Total cost**: $170.82 or $13.6/km²
    - Costs lower, the more rural the area
    - More privacy in urban areas – offers k-anonymity
Prior Work

- **Estimating and predicting solar generation from location**
  - Commonly done by solar installers
  - Variety of models have been proposed
  - SunSpot does the opposite – estimates location from generation

- **Energy analytics on smart meter data**
  - Analytics represent a potential privacy threat
    ◦ Not significant, as long as energy data is anonymous
  - SunSpot exposes a new and different vulnerability
    ◦ Data most believe is anonymous may not be
Summary

- SunSpot first work to expose this localization threat

**Some issues**
- Only uses 3 datapoints per day to infer location
  - Every datapoint provides identifying information
- Only uses solar signature
  - Weather signature also provides identifying information
- Requires data over many days
  - Can reduce using more datapoints and weather
- Requires fine-grained data for precision (second or minute)
  - 5-minute to 1-hour resolution more typical
Combining SunDance and SunSpot?

• Can we accurately localize coarse net meter data?
  - Much more significant privacy threat
  - SunSpot requires second- or minute-level data to localize a region
  - SunDance uses 1-hour resolution, since weather archives are 1-hour

• Current Work
  - Looking at adding weather signatures to solar signatures
    ‣ Look to be *much more* accurate with *coarser* data
  - Leveraging irradiance estimates from visible satellite imagery
    ‣ Highly accurate and updated every ~15 minutes
    ‣ Spatial resolution of 1km²
Preserving Privacy?

• Many possible options with different tradeoffs
  
  1. Remove timestamps from data
     - Pro – cannot identify longitude from solar signature
     - Con – can still identify latitude from solar signature
       - Can probably identify longitude from weather signature
       - Timestamps are useful for well-intentioned analytics
  
  2. Obscure the time of sunrise, sunset, and solar noon
     - Use battery, or actively control solar output at inverter/optimizer
     - Pro – could likely mitigate SunSpot attack using little energy
     - Con – more sophisticated attacks that use whole signature still possible
       - Does not address weather-based attacks, potentially accurate at day-level
       - Less efficient, reduces our generation
Questions?