Analysis of Smart Lighting Systems

Xuerui Yan
Boyang Cai

Supervised by Prof. Mani Srivastava
EE M202B @UCLA

Problem Statement

- Phenomenon of ever-lasting lighting at large-scale public buildings
- Goal: Compare energy consumptions in large-scale public buildings among naive lighting control strategy, sensor mode control strategy, and data precontrol model strategy

Background of Methodology

- Naive Lighting Control
  - Whatever Keep light ON
- Sensor-mode Control
  - Once “1” raised in sensor, keep light ON for a constant time
- Data Precontrol Model
  - Machine Learning based regression to predict light OFF waiting condition in different time intervals during a day

Data Precontrol Model (DPM) Frame

1. Mine the light ON/OFF (1/0) data of testing day(s) from smart sensors

2. Get non-linear regression of data by utilizing Machine learning algorithm

3. Obtain approximated function from regression, and do second derivative and definite integral to set different time intervals and their averaged light OFF waiting time (0->1), respectively

4. Output the predicted light power consumptions in the time intervals daily

Controllable Parameters in DPM Computation

- Changeable data collection area for DPM computing
- Regression degrees
- Regularization Strength (alpha)
- Normalization
- Light OFF waiting threshold time
- Customized light ON wait time for lighting comfort

Machine Learning Regression Algorithm in DPM

- Ridge Regression
  - From n data 1d points, it suffices to build the Vandermonde matrix, which is n data * n-th degree+1

  - Generated matrix can be interpreted as a matrix of pseudo features (the points raised to some power). The matrix is akin to (but different from) the matrix induced by a polynomial kernel

```
[[1, x_1, x_1 ** 2, x_1 ** 3, ...],
     [1, x_2, x_2 ** 2, x_2 ** 3, ...], ...]
```

Machine Learning Regression Algorithm in DPM

- Polynomial Features
  - Generate a new feature matrix consisting of all polynomial combinations of the features with degree less than or equal to the required regressive degree.

- Implement polynomial regression with a linear (Ridge) model, using a pipeline to add polynomial features.

Energy Computation Algorithm in DPM

- Compute the lighting energy consumption (Unit Watt * h) in different resulted time intervals obtained from regressive function
  - Base equation: Energy consumption = Energy_light_ON + Energy_light_setup

- 5-minute Rule[1] of light ON/OFF for setting Light OFF waiting threshold time
  - Rule 1: If you will be out of a room for 5 minutes or less, leave it on
  - Rule 2: If you will be out of a room for more than 5 minutes, turn it off.
  - Used for calculating setup energy cost of the light

Energy Computation Algorithm in DPM

- Introduce light-OFF Probability
  - Ratio of sum of light OFF waiting time (0->1) above the waiting threshold time vs. the sum of total light OFF waiting time in one time interval
  - If Probability <= 0.6, execute Rule 1 (over-threshold waiting time not dominated)
    - ***tdMean*** average waiting time of the time interval obtained by integration from regressive equation to compute Energy_light_ON
    - ***occurTime*** the times of over-threshold waiting time to compute Energy_light_setup
  - If Probability > 0.6, execute Rule 2 (over-threshold waiting time dominated)
    - ***totalLightOffTime*** under this rule, light no longer waits threshold time, use total light OFF time instead of *tdmean* to compute Energy_light_ON
    - ***totalTime*** the total light ON/OFF time to compute Energy_light_setup
      - Punish the *totalTime* for more general prediction

Energy Computation Algorithm in DPM

- Introduce light-OFF Probability

```
if lightOffProbability > 0.6:
    lightOnTime = deltaT - totalLightOffTime - totalTime * 1 + waitTime
    optimized_Energy = lightOnTime * maintainPower + (1 - lightOffProbability) * totalTime * setupEnergy
```

```
else:
    lightOnTime = deltaT - (tdMean * occurTime) - totalTime * 1 + waitTime
    optimized_Energy = lightOnTime * maintainPower + occurTime * setupEnergy
```

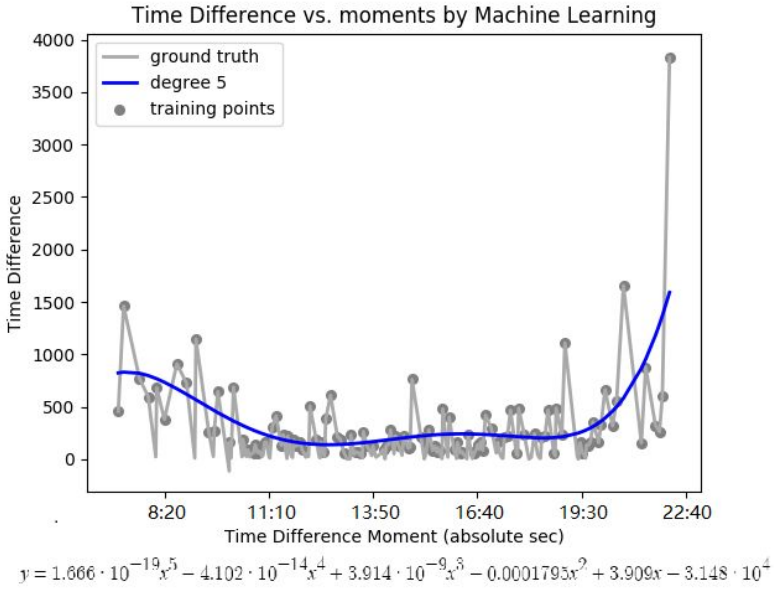
Energy Computation Algorithm in Naive and Sensor-mode Control

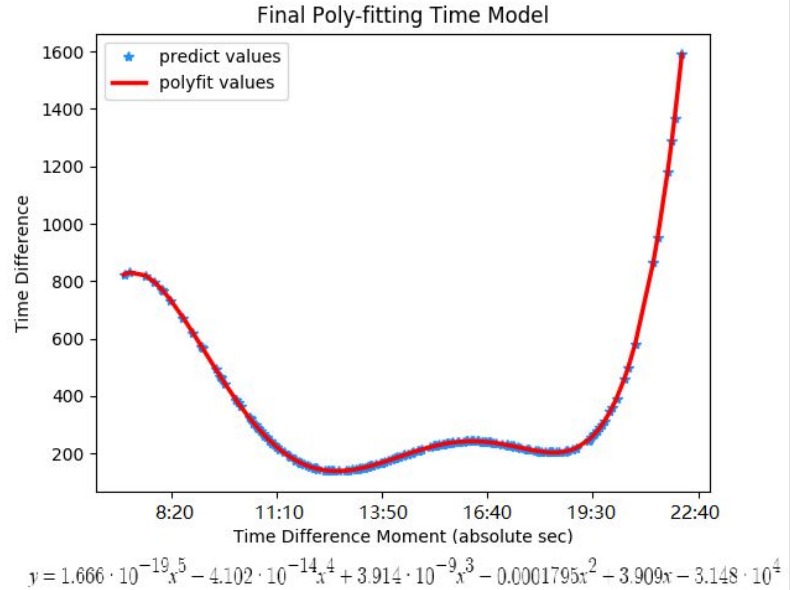
- Naive Energy consumption = Energy_light_ON
  - Ignore Setup energy

- Sensor-mode Energy consumption
  - ***LightOffTime*** the sum of light off time for over-threshold waiting time
  - Energy consumption = ***occurTime*** * setupEnergy + maintainPower * (deltaT - ***LightOffTime*** + ***occurTime*** * 5 * 60)

DPM Programming Output Validation Example

- Weekday Area 1





DPM Programming Output Validation Example

- Weekday Area 1

```
          5             4             3             2
1.666e-19 x - 4.102e-14 x + 3.914e-09 x - 0.0001795 x + 3.909 x - 3.148e+04
root_points [ 33153.34565545  51563.34181604  63042.51812588]
The testing time of this dataset is from 07:05:38 to 21:47:31 in real clock time
occurTime 9
7162 8129.09051961
0.994973471097
the average of time differences(0->1) in this time interval is  739.008229055 this time interval is from 25538s to 33153s in absolute seconds
the naive energy consumption in this time period is 57.1663852894WH, whereas our optimized energy consumption is 3.31451756156WH
the energy consumption for the sensor mode is 42.4547186227WH
occurTime 7
9219 17080.9183689
0.38670137759
the average of time differences(0->1) in this time interval is  240.576315054 this time interval is from 33153s to 51563s in absolute seconds
the naive energy consumption in this time period is 135.127750049WH, whereas our optimized energy consumption is 135.45250301WH
the energy consumption for the sensor mode is 137.034416715WH
occurTime 5
6837 11143.8058518
0.346204475647
the average of time differences(0->1) in this time interval is  227.42460922 this time interval is from 51563s to 63042s in absolute seconds
the naive energy consumption in this time period is 85.0718289044WH, whereas our optimized energy consumption is 85.1720513492WH
the energy consumption for the sensor mode is 87.1229400155WH
occurTime 14
11499 20046.3562938
0.756326637099
the average of time differences(0->1) in this time interval is  488.935519361 this time interval is from 63042s to 78451s in absolute seconds
the naive energy consumption in this time period is 113.450146869WH, whereas our optimized energy consumption is 49.5853528286WH
the energy consumption for the sensor mode is 108.842369091WH
[Finished in 6.8s]
```

Experimental Setup

- Data collect place: UCLA Boelter Hall 4th Floor Hallway
  - Four testing areas along the hall, sensed by four motion sensors

- Data collect time (7:00 - 23:00)
  - Weekday: different periods of days in one-week cycle
  - Weekend: different periods of days in three weekend days
  - Data collect period constrained by SEASnet Lab hours

Experimental Setup

- Equipments
  - Samsung SmartThings Hub x 1
  - SmartThings Motion Sensor x 4
  - TP-LINK Wi-Fi Access Point
  - Fluorescent Lights in Boelter Hall
    - 26 Watts

Technical Details

Part Number	STSS-IRM-001
Item Weight	4.6 ounces
Product Dimensions	2.9 x 2.2 x 3.6 inches
Item model number	STSS-IRM-001
Batteries	2 AA batteries required. (included)
Size	Small
Color	White
Power Source	Battery
Item Package Quantity	1
Batteries Included?	Yes
Batteries Required?	Yes
Average Battery Life	2080 hours

Experimental Setup

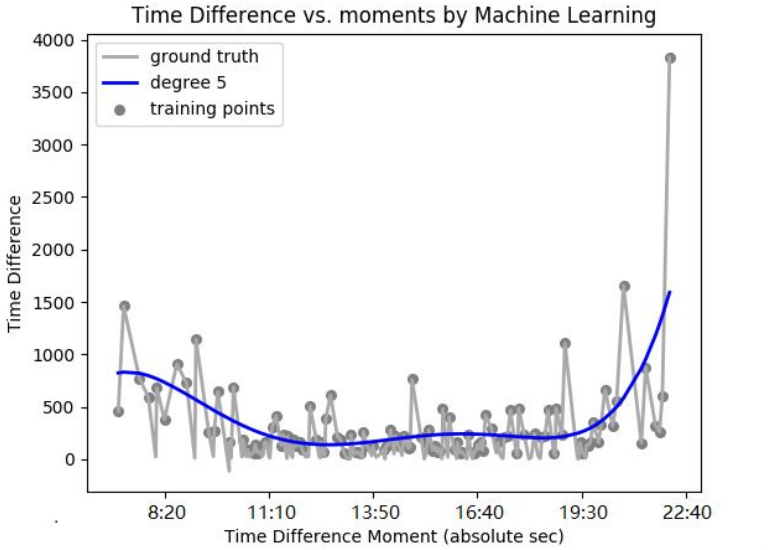
Brief Demonstration

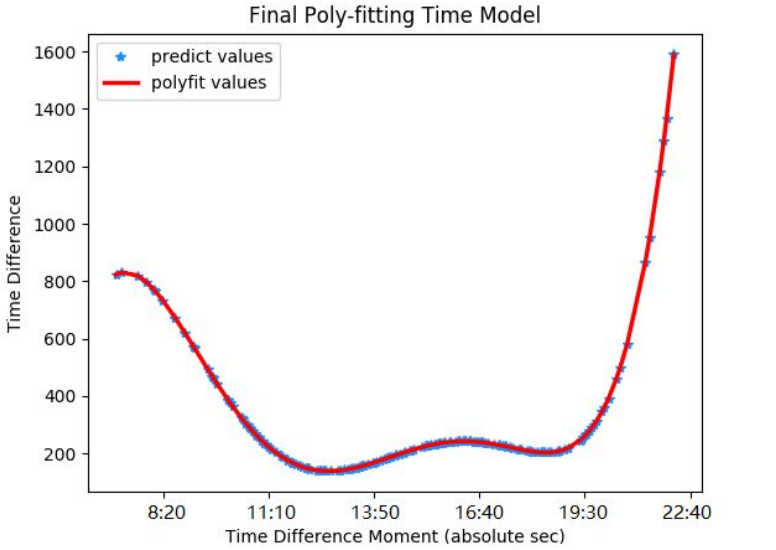
Experimental Results

- Better performance on Ridge Regression for Data Precontrol Model with our recorded data than other robust linear estimators[2]
  - RANSAC
  - TheilSen
  - OLS
- Regression degrees for two data set
  - Weekday: 5th order quadratic equation
  - Weekend: 4th order quadratic equation
    - 4th order Data more stable

Experimental Results

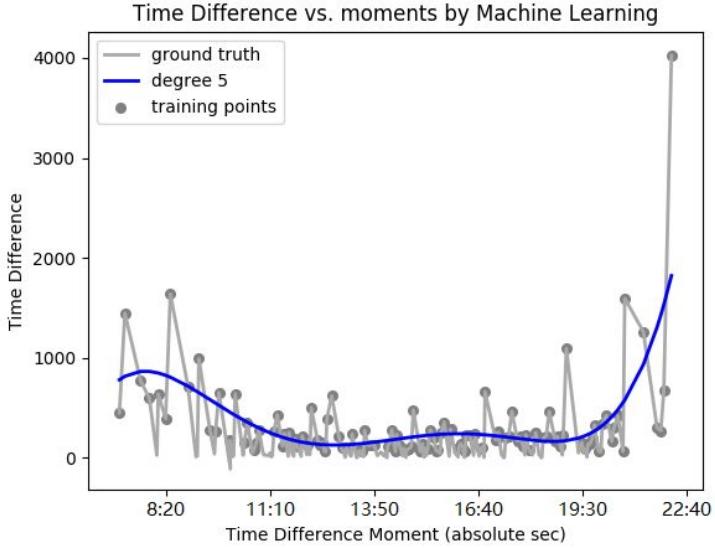
- Weekday Area 1



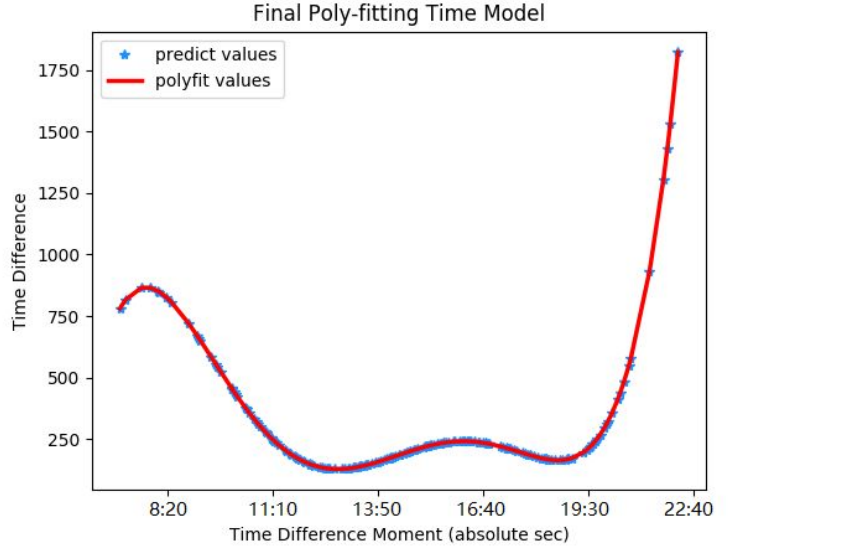


Experimental Results

- Weekday Area 2



$$y = 2.254 \cdot 10^{-19}x^5 - 5.62 \cdot 10^{-14}x^4 - 5.443 \cdot 10^{-9}x^3 - 0.0002544x^2 + 5.686x - 4.771 \cdot 10^4$$



$$y = 2.254 \cdot 10^{-19}x^5 - 5.62 \cdot 10^{-14}x^4 + 5.443 \cdot 10^{-9}x^3 - 0.0002544x^2 + 5.686x - 4.771 \cdot 10^4$$

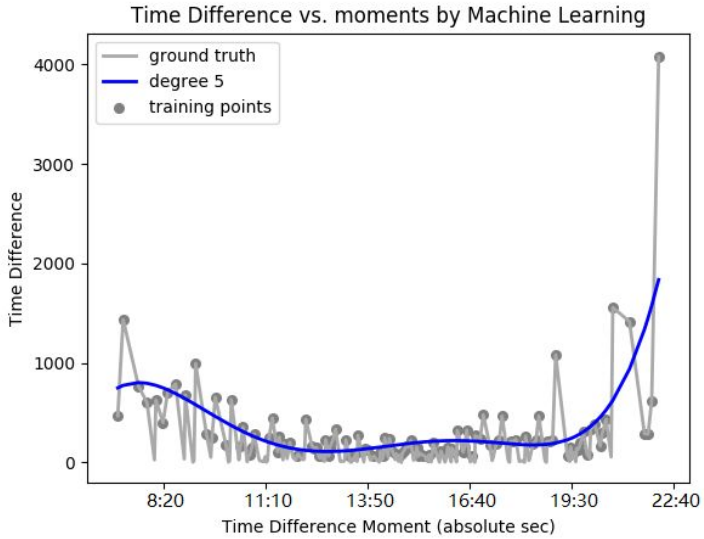
Experimental Results

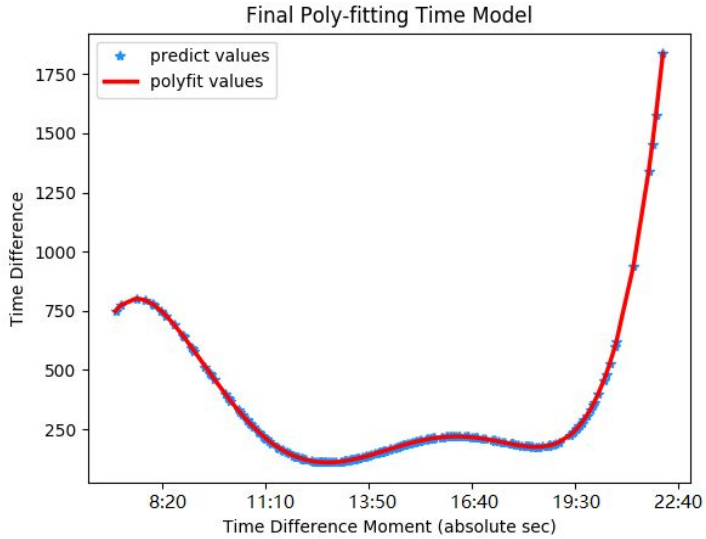
- Comparison among three strategies

weekday								
	Area 1				Area 2			
Sensor Mode Type	Period	Sensor Mode	Naive	Data Precontrol Model	Period	Sensor Mode	Naive	Data Precontrol Model
	7:05:38	42.45	57.17	3.31	**7:05:38**	47.35	65.81	7.11
	9:12:33	137.03	135.13	135.45	**9:32:30**	129.69	127.98	128.51
	14:19:23	87.12	85.07	85.17	**14:22:51**	87.19	86.60	86.30
	17:30:42	108.84	113.45	49.59	**17:37:42**	103.89	110.75	60.96
Total Energy(WH)		**375.45**	**390.82**	**273.52**		**368.12**	**391.14**	**282.88**

Experimental Results

- Weekday Area 3

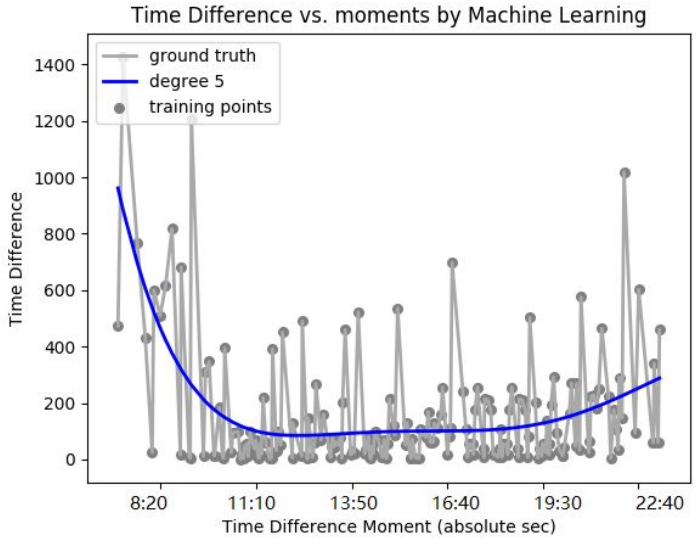




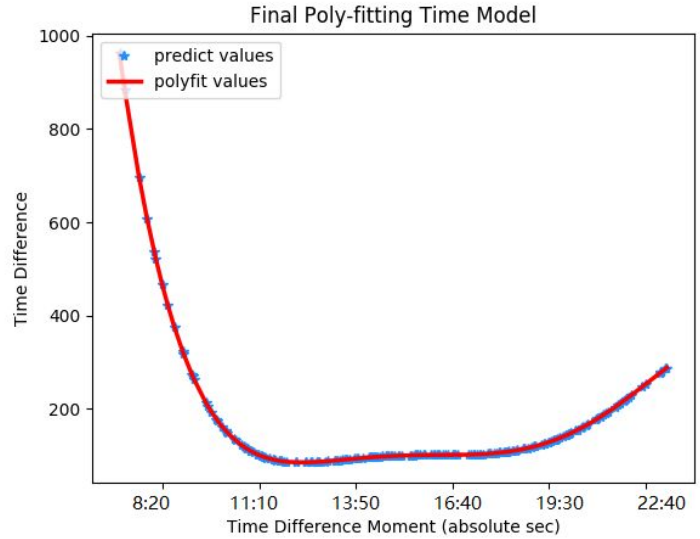
$y = 2.015 \cdot 10^{-19}x^5 - 4.996 \cdot 10^{-14}x^4 + 4.811 \cdot 10^{-9}x^3 - 0.0002235x^2 + 4.956x - 4.115 \cdot 10^4$ $y = 2.015 \cdot 10^{-19}x^5 - 4.996 \cdot 10^{-14}x^4 + 4.811 \cdot 10^{-9}x^3 - 0.0002235x^2 + 4.956x - 4.115 \cdot 10^4$

Experimental Results

- Weekday Area 4



$$y = -2.268 \cdot 10^{-20}x^5 + 7.105 \cdot 10^{-15}x^4 - 8.706 \cdot 10^{-10}x^3 + 5.23 \cdot 10^{-5}x^2 - 1.542x + 1.796 \cdot 10^4$$



$$y = -2.268 \cdot 10^{-20}x^5 + 7.105 \cdot 10^{-15}x^4 - 8.706 \cdot 10^{-10}x^3 + 5.23 \cdot 10^{-5}x^2 - 1.542x + 1.796 \cdot 10^4$$

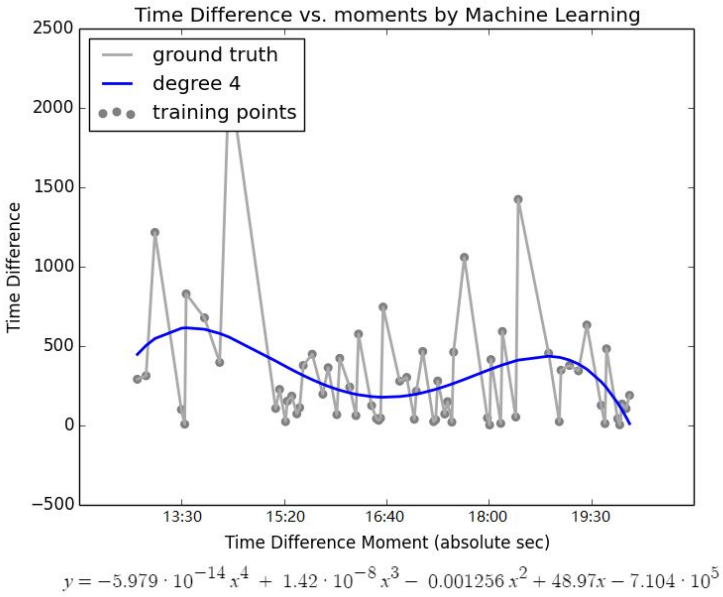
Experimental Results

- Comparison among three strategies

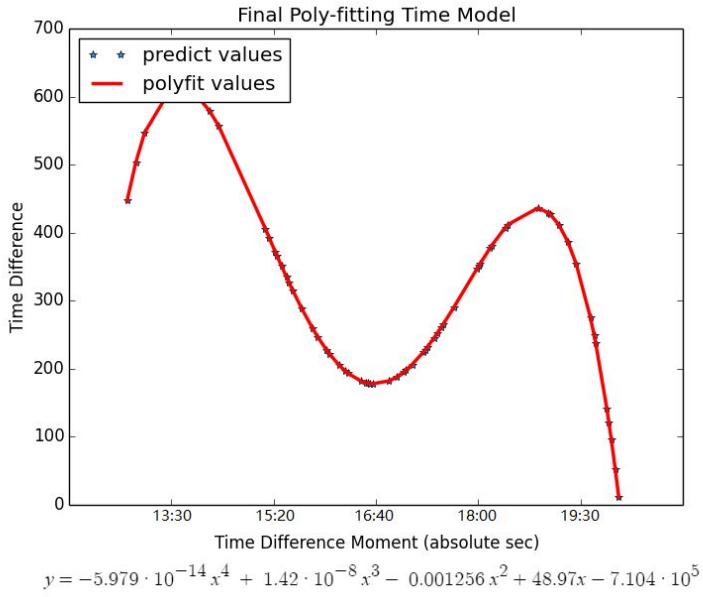
weekday								
	Area 3				Area 4			
Sensor Mode Type	Period	Sensor Mode	Naive	Data Precontrol Model	Period	Sensor Mode	Naive	Data Precontrol Model
	7:05:38	50.54	63.43	6.95	**7:05:38**	165.74	170.69	108.04
	9:26:59	132.75	130.15	131.24	**13:34:32**	72.50	72.94	74.86
	14:22:20	85.81	83.49	83.04	**16:17:52**	154.64	159.28	163.26
	17:30:00	103.88	114.13	68.13	**22:20:26**	14.82	15.14	12.18
Total Energy(WH)		**372.98**	**391.19**	**289.35**		**407.70**	**418.06**	**358.33**

Experimental Results

- Weekend Area 1



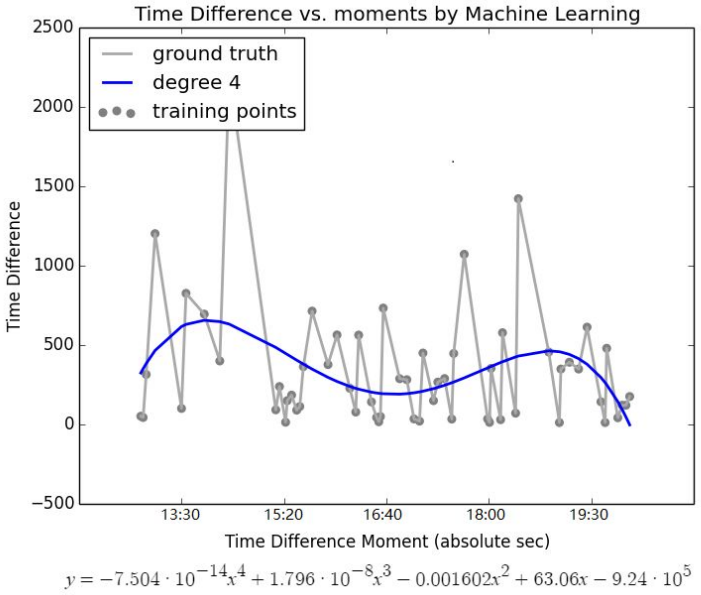
$$y = -5.979 \cdot 10^{-14}\,x^4 + 1.42 \cdot 10^{-8}\,x^3 - 0.001256\,x^2 + 48.97x - 7.104 \cdot 10^5$$



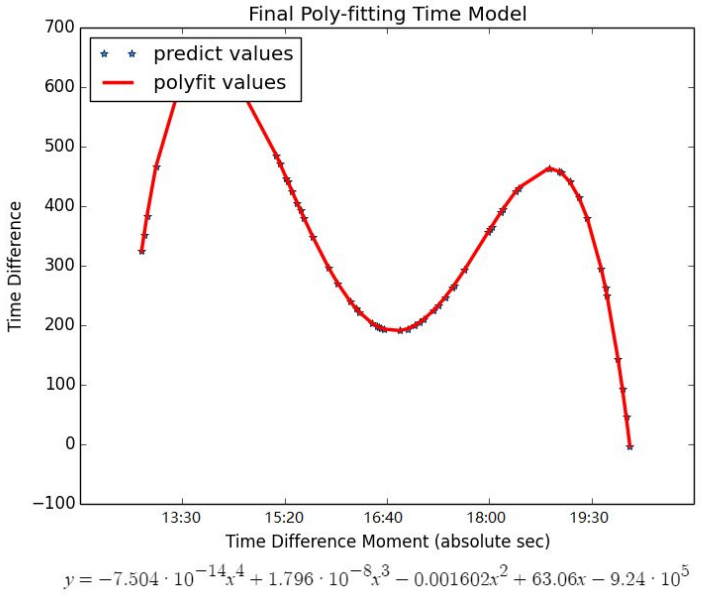
$$y = -5.979 \cdot 10^{-14}\,x^4 + 1.42 \cdot 10^{-8}\,x^3 - 0.001256\,x^2 + 48.97x - 7.104 \cdot 10^5$$

Experimental Results

- Weekend Area 2



$$y = -7.504 \cdot 10^{-14}x^4 + 1.796 \cdot 10^{-8}x^3 - 0.001602x^2 + 63.06x - 9.24 \cdot 10^5$$



$$y = -7.504 \cdot 10^{-14}x^4 + 1.796 \cdot 10^{-8}x^3 - 0.001602x^2 + 63.06x - 9.24 \cdot 10^5$$

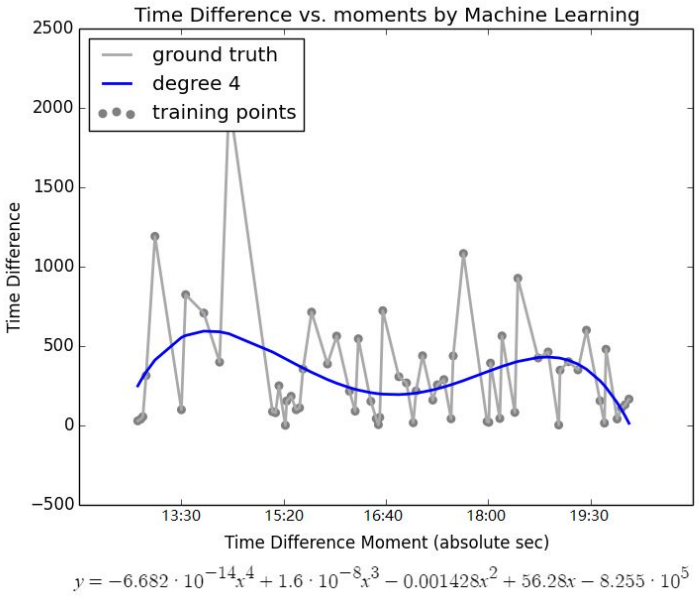
Experimental Results

- Comparison among three strategies

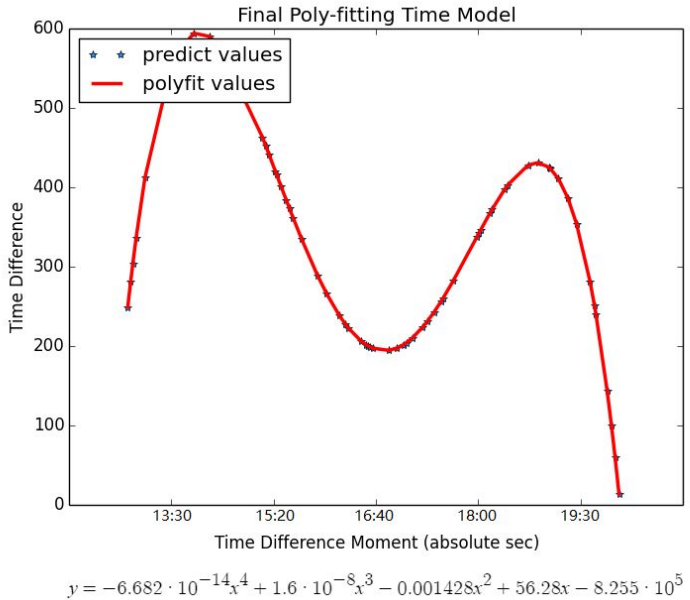
weekend								
	Area 1				Area 2			
Sensor Model Type	Period	Sensor Model	Naive	Data Precontrol Model	Period	Sensor Model	Naive	Data Precontrol Model
	13:17:22	32.18	49.16	4.59	**13:20:11**	35.76	52.80	7.99
	15:05:48	77.97	74.79	40.10	**15:17:02**	70.28	71.81	32.76
	17:53:24	56.14	56.08	17.40	**17:57:45**	54.89	54.24	15.97
Total Energy(WH)		**166.28**	**180.03**	**62.08**		**160.93**	**178.86**	**56.71**

Experimental Results

- Weekend Area 3



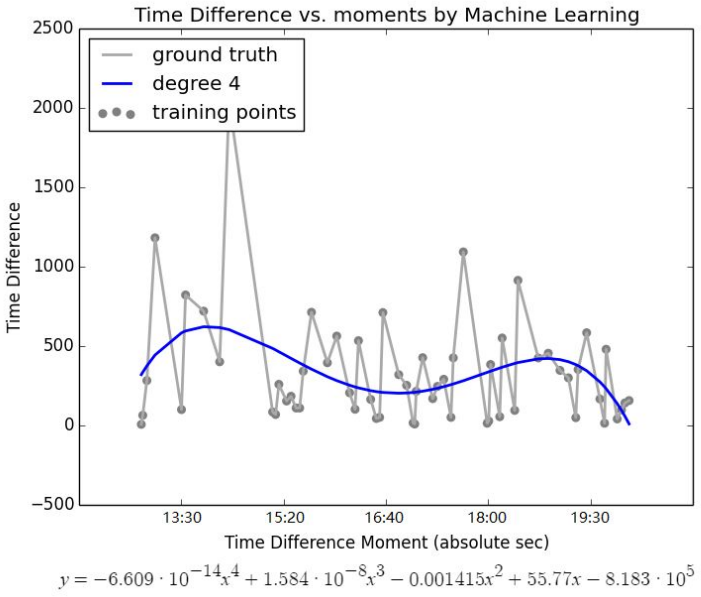
$$y = -6.682 \cdot 10^{-14}x^4 + 1.6 \cdot 10^{-8}x^3 - 0.001428x^2 + 56.28x - 8.255 \cdot 10^5$$

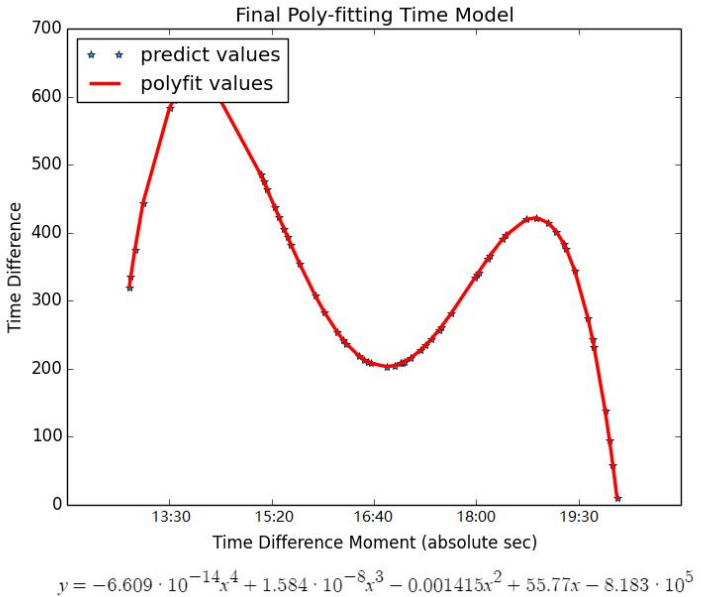


$$y = -6.682 \cdot 10^{-14}x^4 + 1.6 \cdot 10^{-8}x^3 - 0.001428x^2 + 56.28x - 8.255 \cdot 10^5$$

Experimental Results

- Weekend Area 4





Experimental Results

- Comparison among three strategies

weekend								
	Area 3				Area 4			
Sensor Mode Type	Period	Sensor Mode	Naive	Data Precontrol Model	Period	Sensor Mode	Naive	Data Precontrol Model
	13:17:53	38.42	54.36	10.96	**13:20:45**	35.41	53.32	11.22
	15:18:19	72.12	71.27	29.20	**15:18:48**	72.43	71.36	29.54
	17:57:46	59.53	54.28	16.54	**17:58:28**	60.41	54.01	17.54
Total Energy(WH)		**170.08**	**179.90**	**56.70**		**168.26**	**178.69**	**58.30**

Conclusion

- The result shows that Data Precontrol Model has a significant drop compared to the Naive model
- The Data Precontrol Model also have a benefit compared to the Sensor Mode
- However, Data Precontrol Model required more hardware equipment and maintenance than the Naive mode and Sensor more
    - Need maintain on the machine learning server
    - Sensor maintenances.

Future Work

- Getting more daily data to be iteratively computed in DPM
  - More daytime data
  - Fill Midnight data (0:00 - 6:00am)
- Add more features in machine learning model
  - E.g. Temperature, Schedules of office/class hours in the testing floor
- Measure the real power consumption of fluorescent light with physical devices
- Change better smart sensors with faster reaction and LED for better energy saving
- Expand the sensing area (use long distance sensor) to cover and manage more different areas

Future Work

- Currently only use the SmartThings related devices to sense and transmit data for validation purpose
  - In the future we can use microcontroller to replace the SmartThings Hub
    - cost vs complexity

Reference

[1] “Does Turning Fluorescent Lights Off Use More Energy Than Leaving Them On?”. Retrieved from: https://www.scientificamerican.com/article/turn-fluorescent-lights-off-when-you-leave-room/

[2] “Robust linear estimator fitting”. Retrieved from: http://scikit-learn.org/stable/auto_examples/linear_model/plot_robust_fit.html#sphx-glr-auto-examples-linear-model-plot-robust-fit-py

[3] “Code reference for logging data to google sheet” https://community.smartthings.com/t/log-events-to-google-sheets-see-post-154-for-current-github-repo-and-v1-1/36719