Detection of Manipulated Pricing in Smart Energy CPS Scheduling

Reece Buckle
rtb1c17

May 2021
Department of Electronics and Computer Science
University of Southampton

1 Introduction

Energy companies will publish predictive guideline prices based on past usage, which consumers can use to schedule their own usage requirements over a 24 hour period. This is important because the unit cost per hour follows the actual usage for that hour, so it is in the consumer's best interest to schedule in off-peak times to save money. However, hackers can tamper with the physical security of an energy scheduling system by:

- Hacking the utility network's computer to physically change the guideline prices
- Hacking the smart meters by uploading malicious code during remote upgrading procedures
- Hacking the output of a smart meter to lie about energy usage

In this problem, there are 100 predictive guideline curves in which we're trying to discover abnormal activity (i.e., from a hacker manipulating the predictive guideline curve). Following this, we want to calculate the minimum cost solution for a community of 5 users - all completing 10 tasks per day.

The target of this problem is to:

- Reduce user bill
- Reduce PAR (peak to average ratio) of grid energy usage (by having a more balanced power load)
- Maximise renewable energy usage (e.g., during sunny days when Solar power is viable)

2 Statistical Classification

This section covers a statistical classification approach, developed by me and demonstrated in 'StatisticalMethods.py'. Table 1 presents a comparison of features that were calculated from 5000 normal and 5000 abnormal predictive guideline curves. Initially, I was looking for the frequency of low unit cost occurrences (less than 3) - especially around 8pm, as this is evidence of potential overloading attacks. However, it turned out that the normal curves had more low occurrences (3774), than the abnormal curves (2352). Of all measurements recorded, notable differences include the average PAR (peak-to-average ratio), as well as the average unit cost between hours 9-23 (during waking hours).

Feature	**Abnormal Curves**	**Normal Curves**
Average PAR	1.346	1.364
Max Unit Cost	6.969	6.969
Min Unit Cost	2.663	2.663
Occurrences Less Than 3	2352	3774
Average Unit Cost (Hours 0-8 Inclusive)	3.884	3.833
Average Unit Cost (Hours 9-23 Inclusive)	5.545	5.473

Table 1: Analysis of the Abnormal/Normal Curve Feature-Space

Initially, the training data-set was classified with a threshold value (1.355) based on the PAR; this value is the midpoint between the average normal PAR, and abnormal PAR. This method had an accuracy of 60.12%, and error rate of 39.88%.

As there was a significant difference between hours 9-23 for abnormal & normal curves, this value was also factored into the statistical algorithm. The figure below demonstrates this algorithm as pseudo-code. This method, using the most significant feature differences, resulted in an accuracy of 72.19% and error rate of 27.8%. This is clearly much better than guessing, however as linear SVM produced a much larger accuracy of 94.1%, this methodology was used for the final classification.

Pseudo-code of Statistical Classification

```
# Iterate through each individual guideline curve
# First compare the PAR to the threshold. Then compare the
# average cost between 9-23pm to the normal/abnormal average

if PAR > threshold:

    if average_cost < abnormal_average_cost:
        classify as normal
    else
        classify as abnormal

else if PAR < threshold:

    if average_cost > normal_average_cost:
        classify as abnormal
    else:
        classify as normal
```

3 Machine Learning Techniques

Table 2 compares a selection of classifiers provided by the scikit-learn library, which is demonstrated in 'Classifiers.py'. The highest accuracy was achieved from using the Guassian Process classifier (97.7%). However, as SVM is much faster and efficient, this classifier was used to optimise evaluate. All tests were completed with a standard 70-30 test/train split.

Classifier	Accuracy
Nearest Neighbors	0.789
SVM (Support Vector Machine)	0.954
Linear SVM	0.941
RBF (Radial Basis Function) SVM	0.940
Gaussian Process	0.977
Decision Tree	0.771
Random Forest	0.737
Neural Net	0.789
AdaBoost	0.924
Naive Bayes	0.948
QDA (Quadratic Discriminant Analysis)	0.949

Table 2: A Comparison of Classifier Accuracy

For a final classification, SVM (with a linear kernel) was used, which can be shown from executing 'SVMClassifier.py'. This achieved an accuracy of 94.1% when using a 70/30 test train split. To analyse the standard deviation and prevent over-fitting, 5-fold cross-validation was implemented (with a test/train split of 60/40). This resulted with an accuracy of 0.89%, and standard deviation of 0.12. To further analyse the bias-variance trade-off, the 'mlxtend' library was imported to provide an estimate by decomposing the linear SVM model. As it turns out, the linear SVM had a lower variance (0.009), and higher bias (0.056) which indicates the linear SVM model was more likely to be under-fitting the data. However, both values are relatively low and within the same order of magnitude, so a good balance was still achieved.

Linear SVM Analysis	Result
Accuracy (70/30 split)	0.941
Accuracy (5-Fold Cross Validation)	0.89
Standard Deviation	0.12
Mean Squared Error	0.065
Bias (estimate)	0.056
Variance (estimate)	0.009

Table 3: Analysis of Linear SVM Model

4 Linear Programming Problem

This energy scheduling problem is a minimisation linear programming problem. Given a predictive guideline curve, we need to decide which hours to schedule tasks. As each hour has a different predictive unit cost, an tasks span multiple hours, we need to find a minimum cost solution for completing all tasks. To complete this problem, the OR-Tools library was used. 'LPSolver.py' automates this problem on the classified testing data.

As it is too complicated to model all constraints, the following simplifications are made:

- Power usage for appliances is continuous (same rate per hour)
- No disruptions or variations in power supply
- No renewable energy usage
- Time-span is discrete (e.g., 1 hour, 2 hours, 3 hours, with nothing in between, e.g., 15 minutes)

Constraint 1: $0 <= Variable <= 1$

Each variable per hour is bounded by 0 and the maximum scheduled energy per hour (which is 1 for simplicity).

Constraint 2: $Var_{ready} + ... + Var_{deadline} = Energy\ Demand$

The sum of all variables in a task, between the ready and deadline time, has to equal the energy demand. No more, or less, energy can be used in the duration of that task. Furthermore, all hours outside the ready and deadline bounds are equal to 0. However, these variables are redundant and are not required in the linear programming solver.

Objective: $H_0(\sum_{T1}^{T10} Var_0) + ... + H_{23}(\sum_{T1}^{T10} Var_{23})$

This represents the objective cost function to minimise. It is calculated from the summation of all variables (in a given time task), multiplied by the unit cost for that hour. This includes all users and tasks within the community.

- H represents the unit cost for that hour (from hour 0 to hour 23)
- $T1 - T10$ represents the task ID for each user (from 1 to 10)
- Var represents all variables, across tasks 1 to 10, present in that hour

5 Analysis of Example Predictive Guideline Curves

A hacker could manipulate a predictive guideline curve to show 'cheaper prices' in time-slots that they don't intend to use, so other users do. As the real time price always follows actual energy usage, this would lead to a larger peak usage (and energy cost) in these hours. If a hacker was to show a really cheap cost around peak hours (such as 8pm), this would entice many community users to schedule at this point, thus 'overloading' the system. In more severe cases, overloading could lead to larger scale disruptions in the energy infrastructure.

Figure 1 presents the example normal, and abnormal, predictive guideline curves.

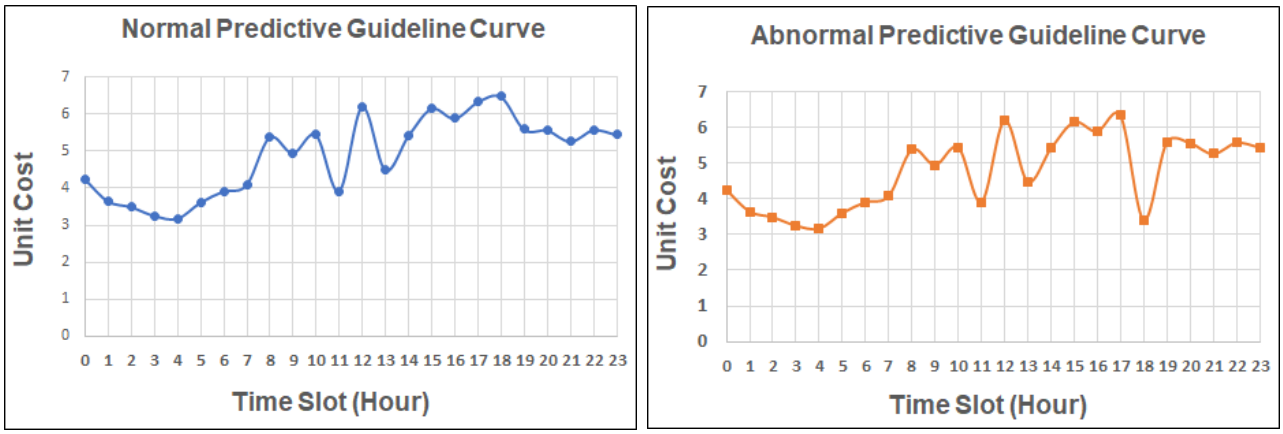


Figure 1: Abnormal & Normal Predictive Guideline Curves

Figure 2 presents the energy scheduling solution for these two curves, which is computed in 'LPExample.py'. The minimum cost solution for normal guideline curve is 454.44 whereas the abnormal curve is 420.52 (much lower).

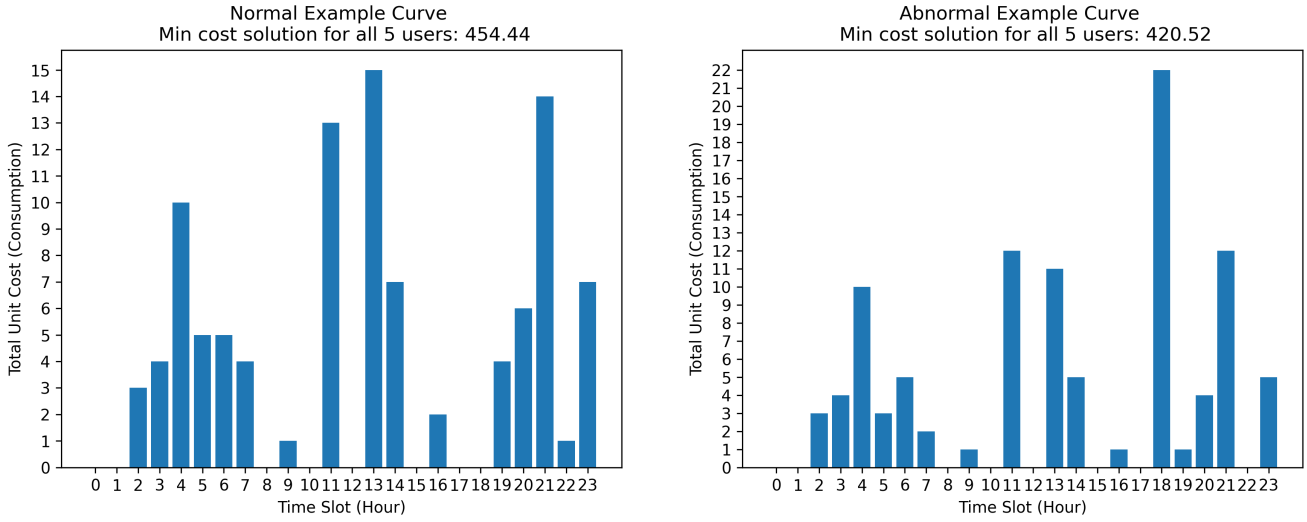


Figure 2: Energy Scheduling Solution of Predictive Guideline Curves

Figure 3 presents a comparison of the two guideline curves. Regarding the abnormal guideline curve, there is a clear dip at hour 18 - intentionally made by the hacker. The effects of this can be seen the linear programming solutions (Figure 2); there is a clear 'overloaded' peak at hour 18 in the abnormal solution in which 22 total units of energy are consumed - in comparison to 0 hours in the normal solution. If all community users followed this minimum programmed solution, this would cause an overload at this point, as it is clear this peak dwarfs the rest of the graph (in which only a maximum of 12 units are typically scheduled per hour).

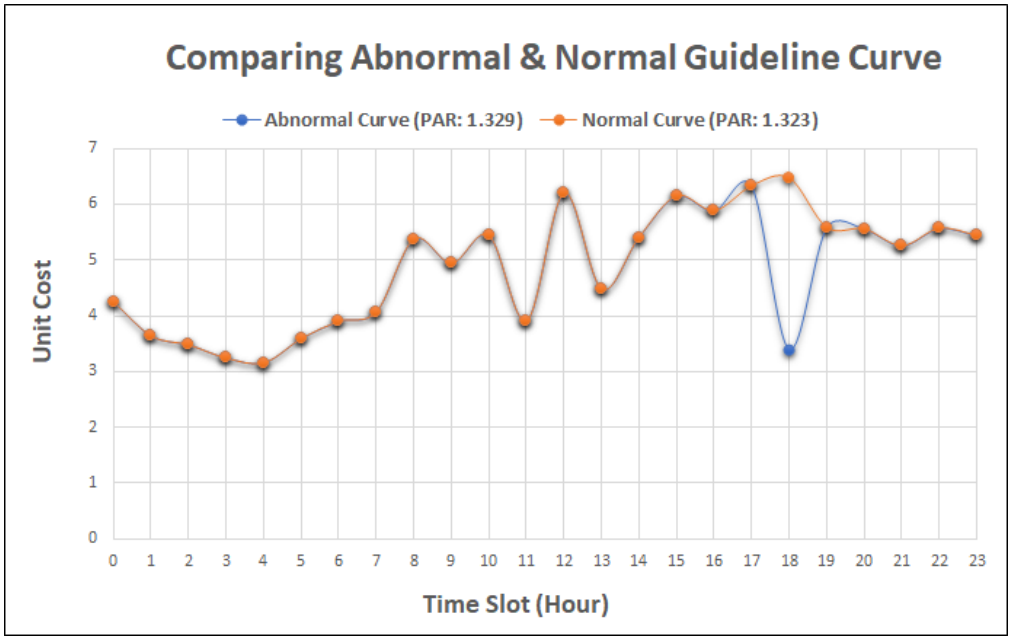


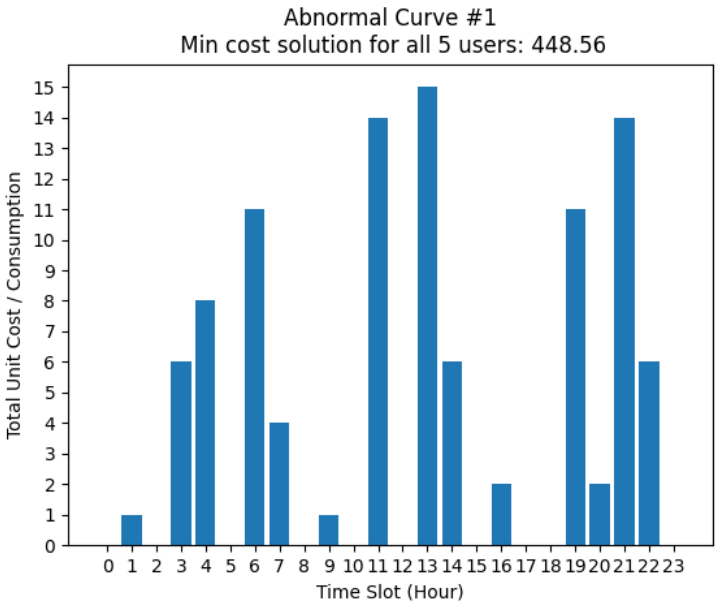
Figure 3: Comparison of Abnormal & Normal Predictive Guideline Curves

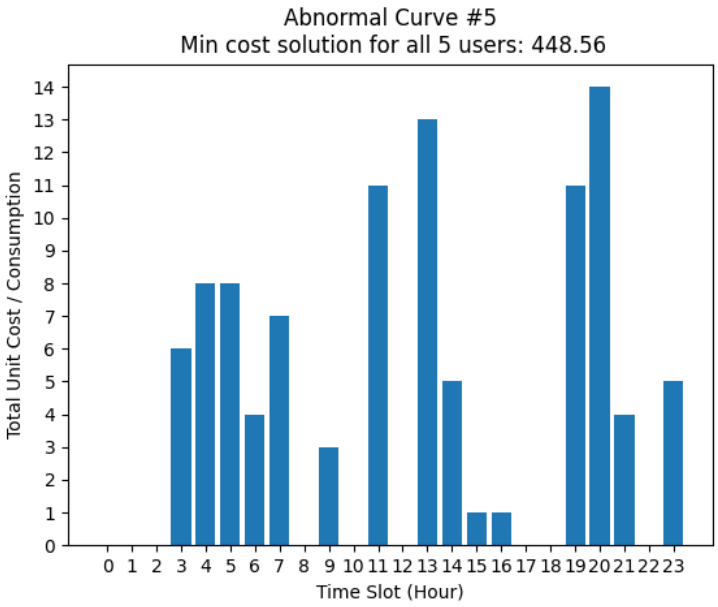
6 Abnormal Charts

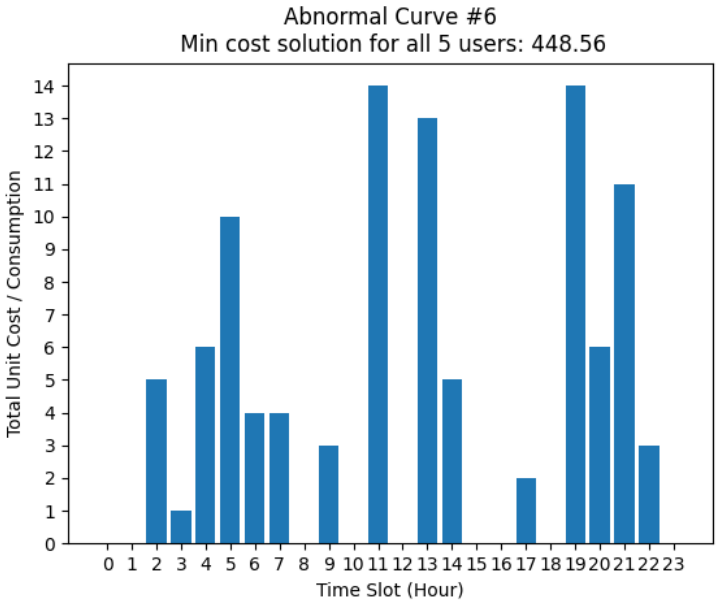
This section presents all abnormal graphs from guideline pricing curves 1-100. Each graph illustrates the minimum cost solution for each hour, as well as the minimum cost solution for all 5 users throughout the 24-hour period. To see all computed solutions for both the normal guideline curves, and abnormal guideline curves, please see their respective folders in 'Graphing Data'. With observation, the minimum cost solutions spans a range of 448.56 - 449.06 for all abnormal curves, and increases by 0.005 per pricing guideline curve number.

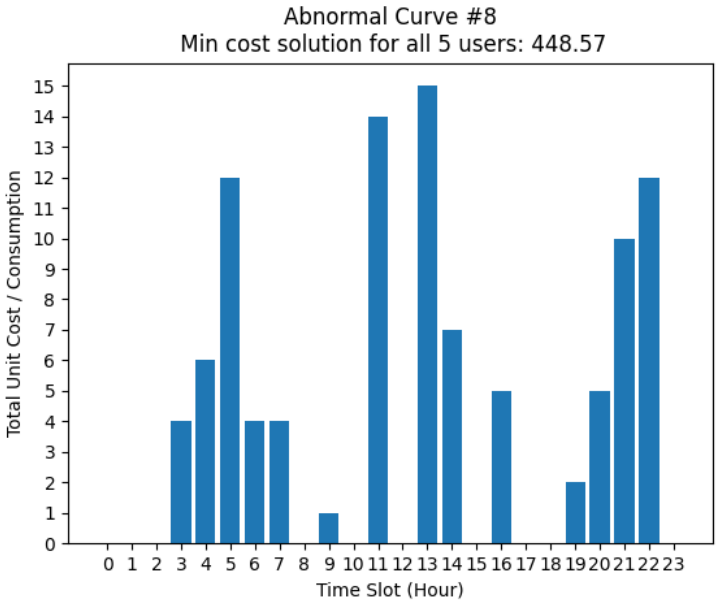
Computed labels (using the linear SVM classifier) is as follows:

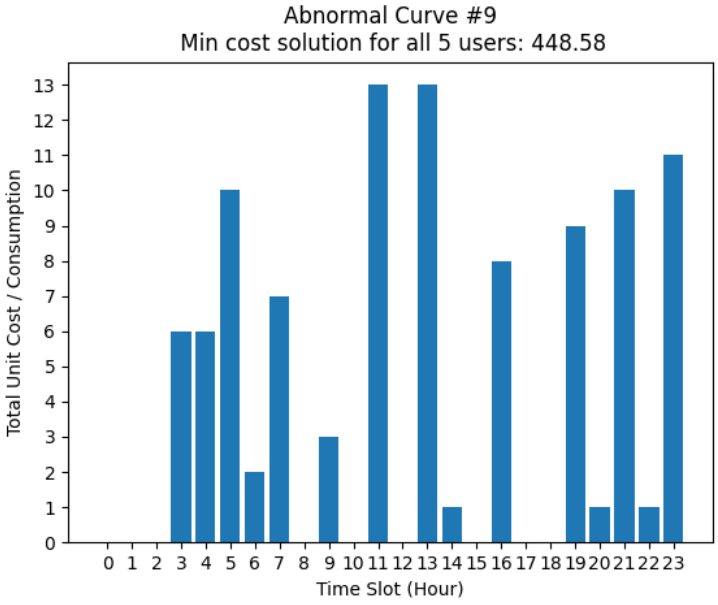
1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0

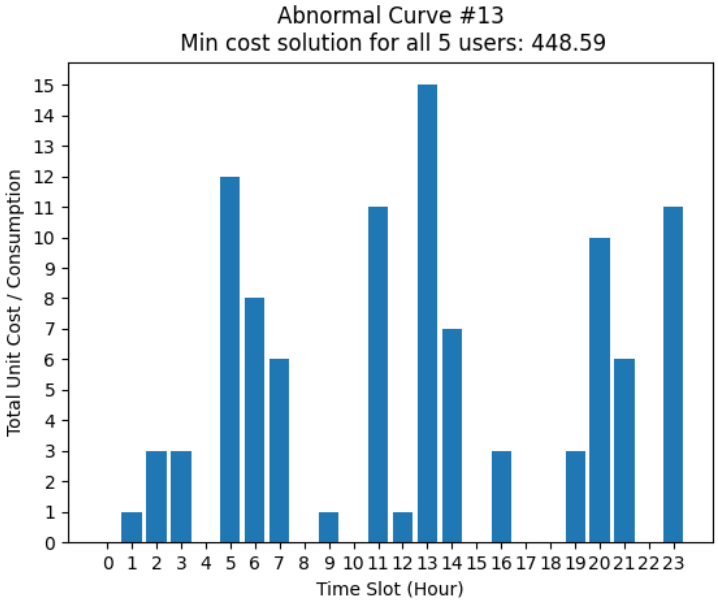


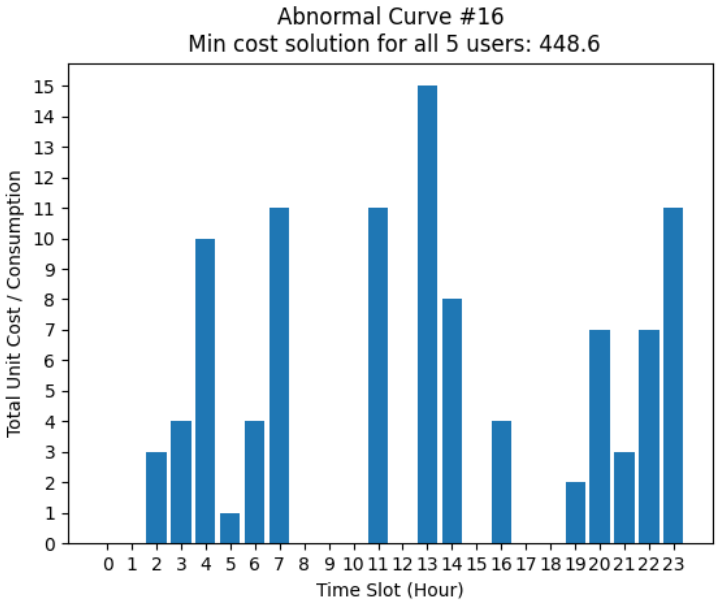


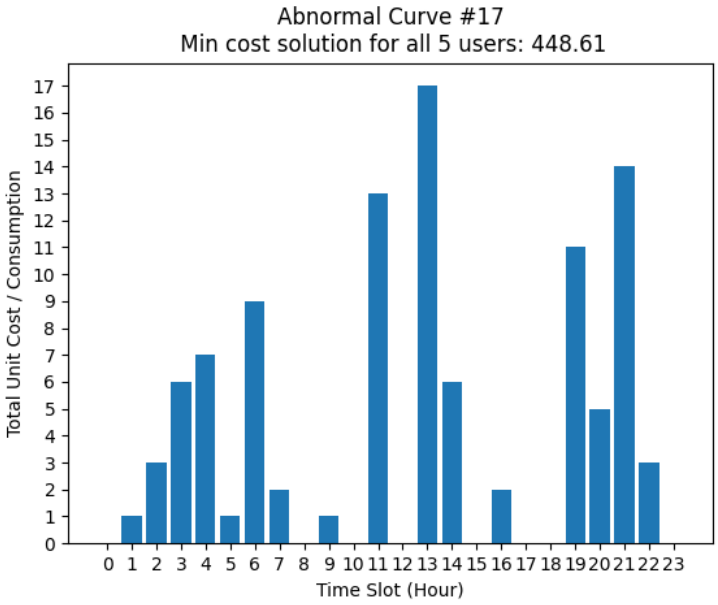

Abnormal Curve #6
Min cost solution for all 5 users: 448.56
Total Unit Cost / Consumption
Time Slot (Hour)

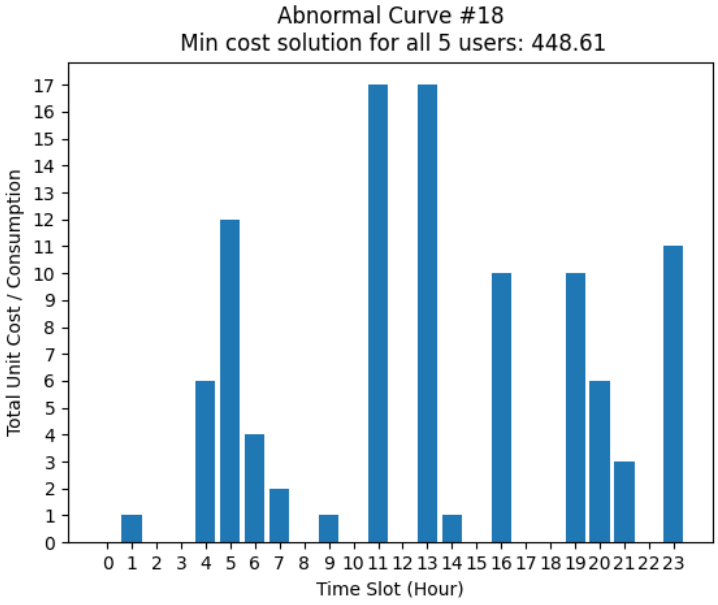

Abnormal Curve #8
Min cost solution for all 5 users: 448.57
Total Unit Cost / Consumption
Time Slot (Hour)

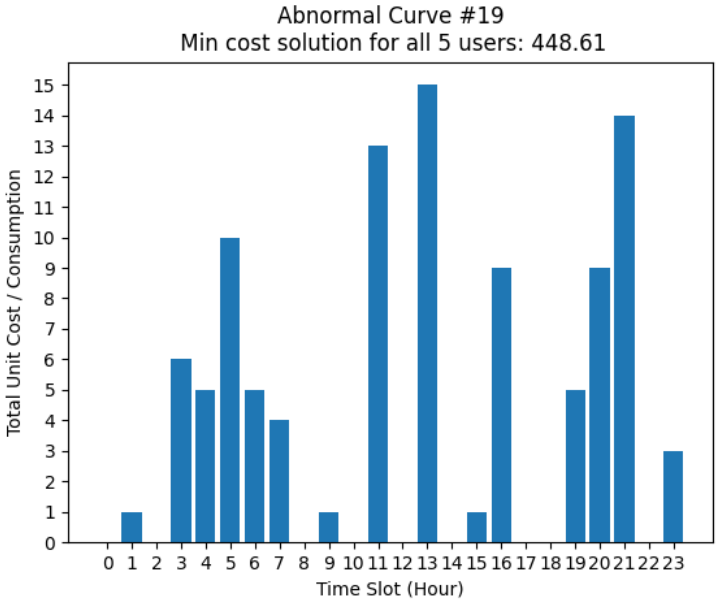

Abnormal Curve #9
Min cost solution for all 5 users: 448.58
Total Unit Cost / Consumption
Time Slot (Hour)

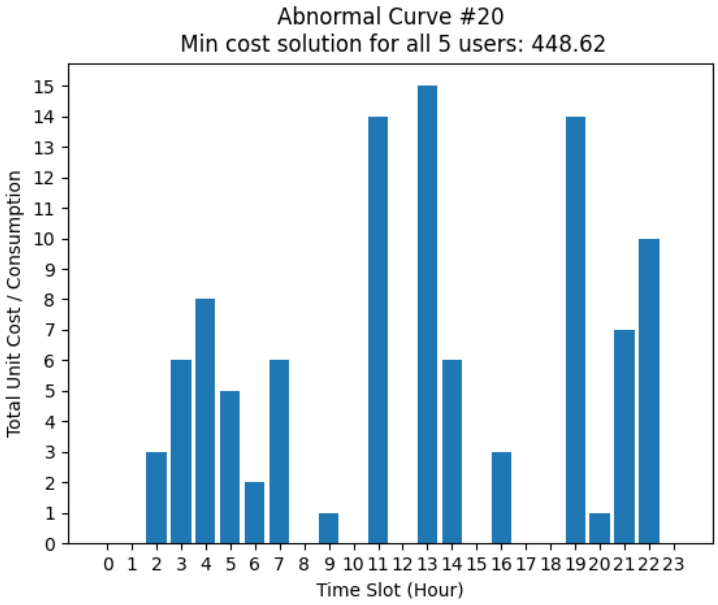

Abnormal Curve #13
Min cost solution for all 5 users: 448.59
Total Unit Cost / Consumption
Time Slot (Hour)

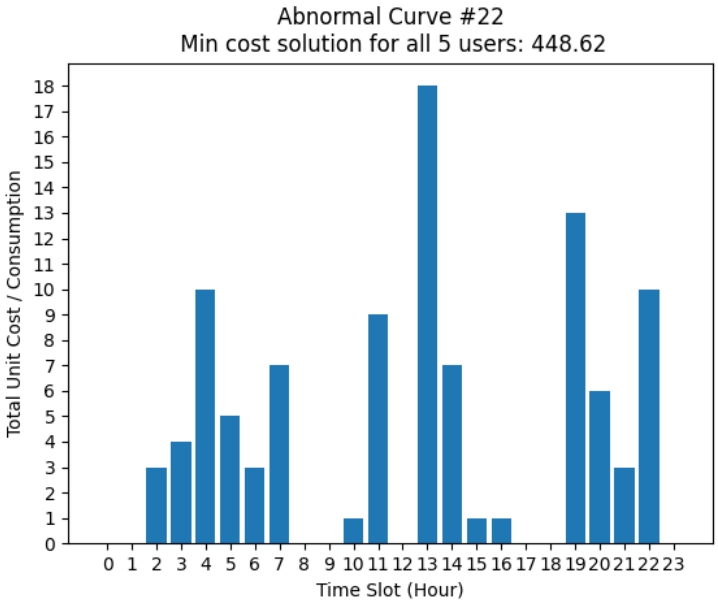

Abnormal Curve #16
Min cost solution for all 5 users: 448.6
Total Unit Cost / Consumption
Time Slot (Hour)

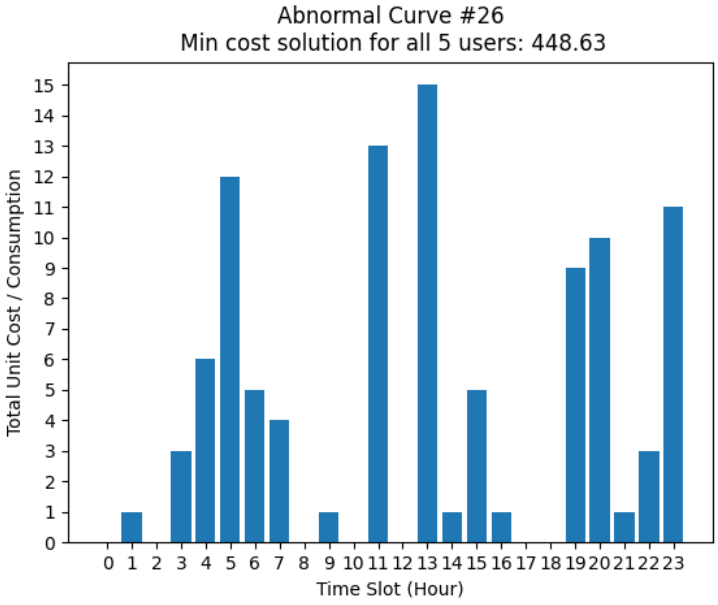

Abnormal Curve #17
Min cost solution for all 5 users: 448.61
Total Unit Cost / Consumption
Time Slot (Hour)

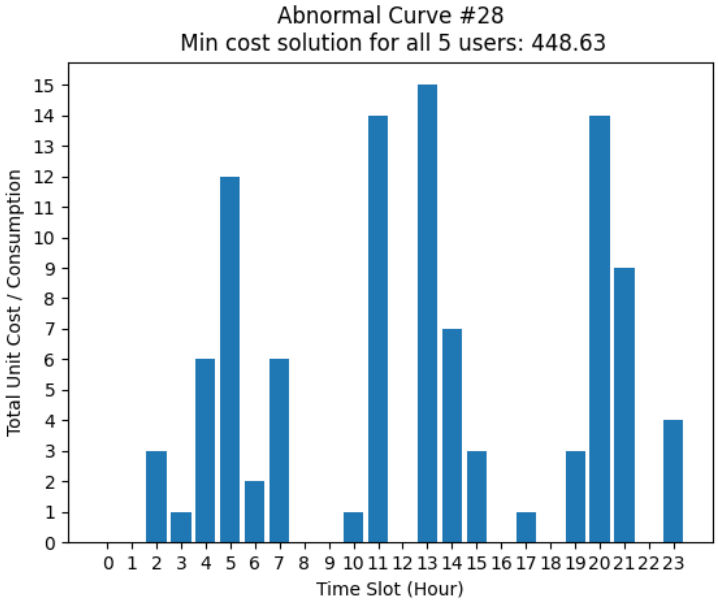

Abnormal Curve #18
Min cost solution for all 5 users: 448.61
Total Unit Cost / Consumption
Time Slot (Hour)

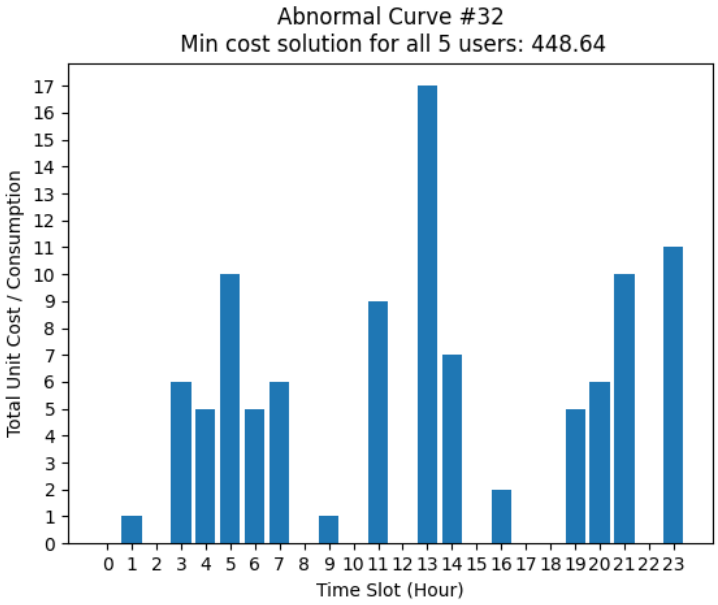

Abnormal Curve #19
Min cost solution for all 5 users: 448.61
Total Unit Cost / Consumption
Time Slot (Hour)

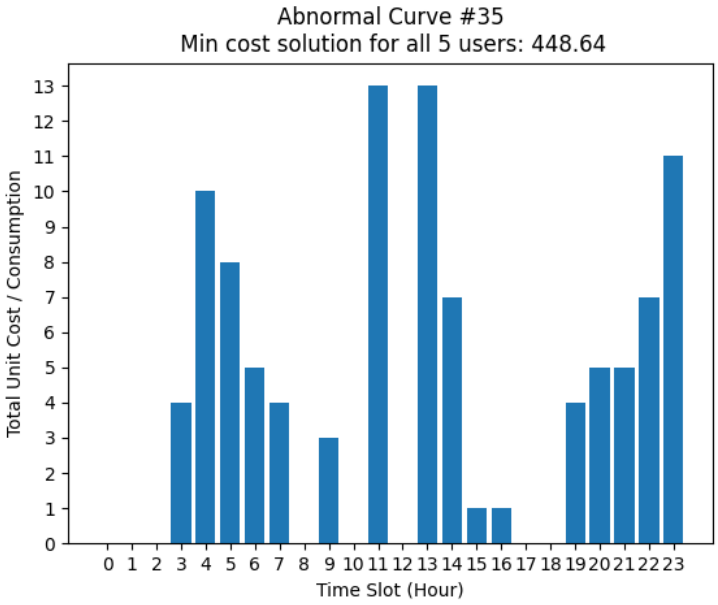

Abnormal Curve #20
Min cost solution for all 5 users: 448.62
Total Unit Cost / Consumption
Time Slot (Hour)

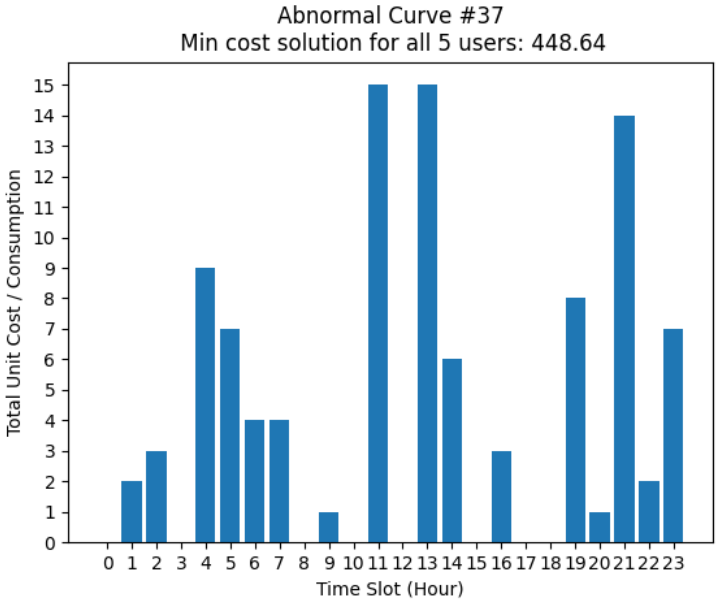

Abnormal Curve #22
Min cost solution for all 5 users: 448.62
Total Unit Cost / Consumption
Time Slot (Hour)

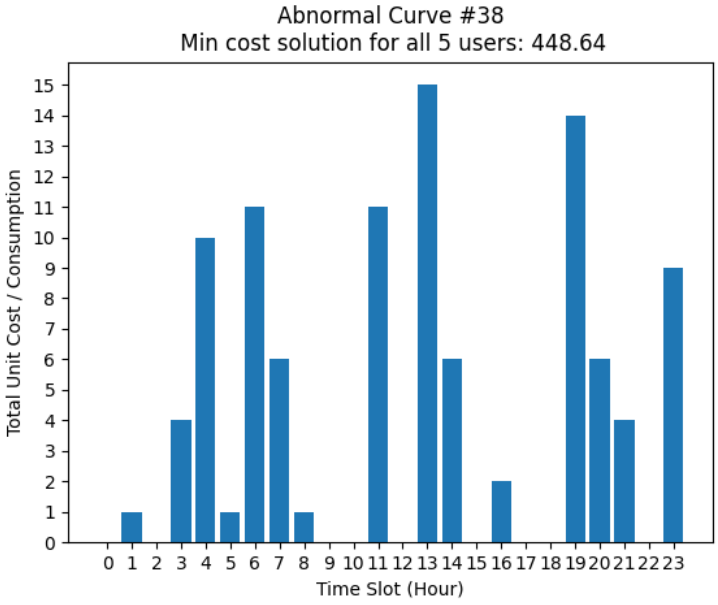

Abnormal Curve #26
Min cost solution for all 5 users: 448.63
Total Unit Cost / Consumption
Time Slot (Hour)

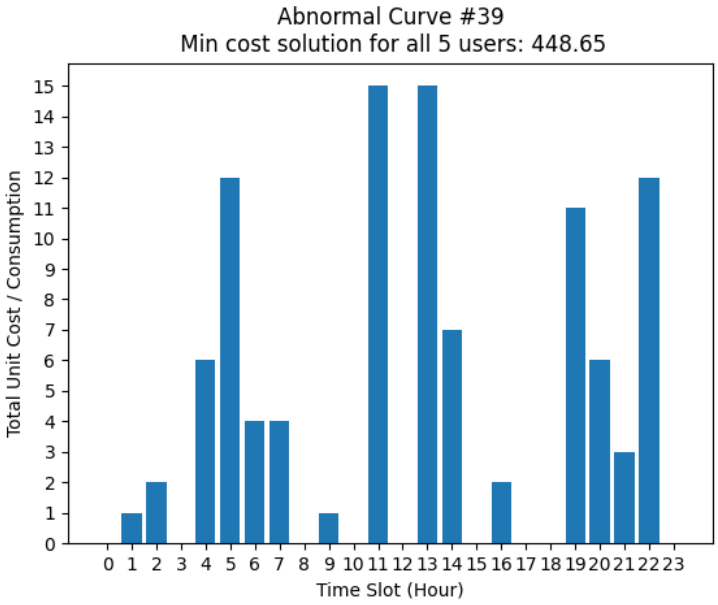

Abnormal Curve #28
Min cost solution for all 5 users: 448.63
Total Unit Cost / Consumption
Time Slot (Hour)

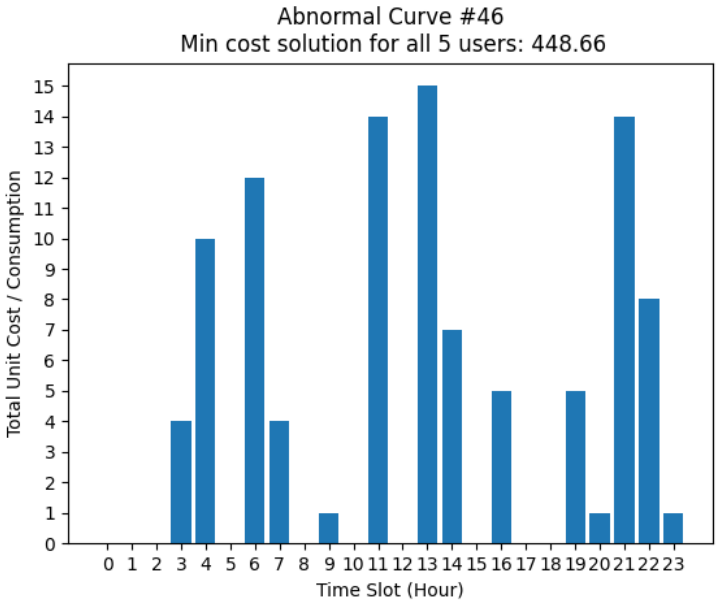

Abnormal Curve #32
Min cost solution for all 5 users: 448.64
Total Unit Cost / Consumption
Time Slot (Hour)

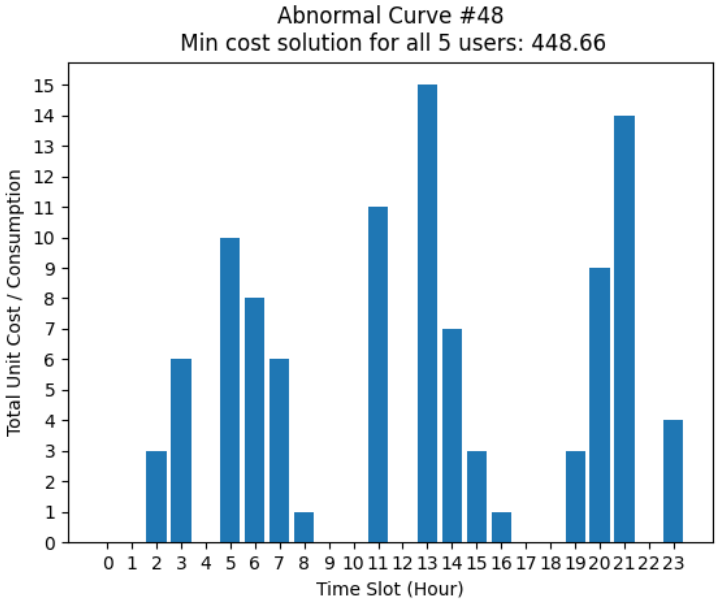

Abnormal Curve #35
Min cost solution for all 5 users: 448.64
Total Unit Cost / Consumption
Time Slot (Hour)

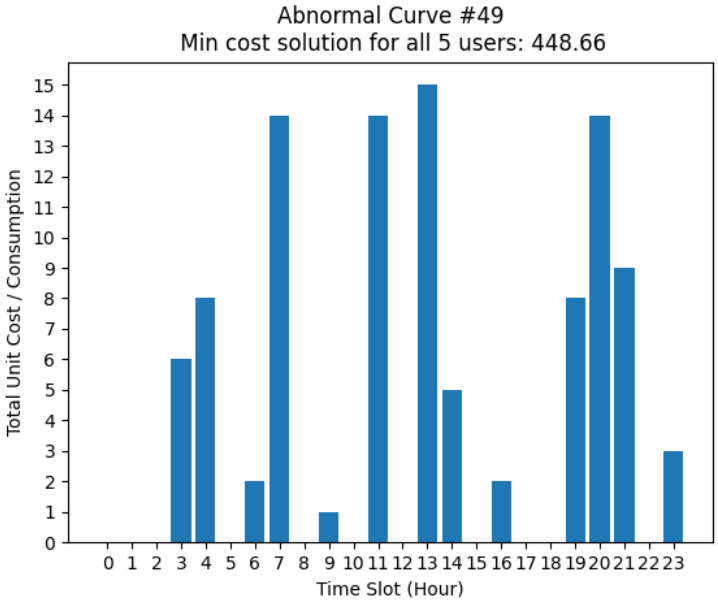

Abnormal Curve #37
Min cost solution for all 5 users: 448.64
Total Unit Cost / Consumption
Time Slot (Hour)

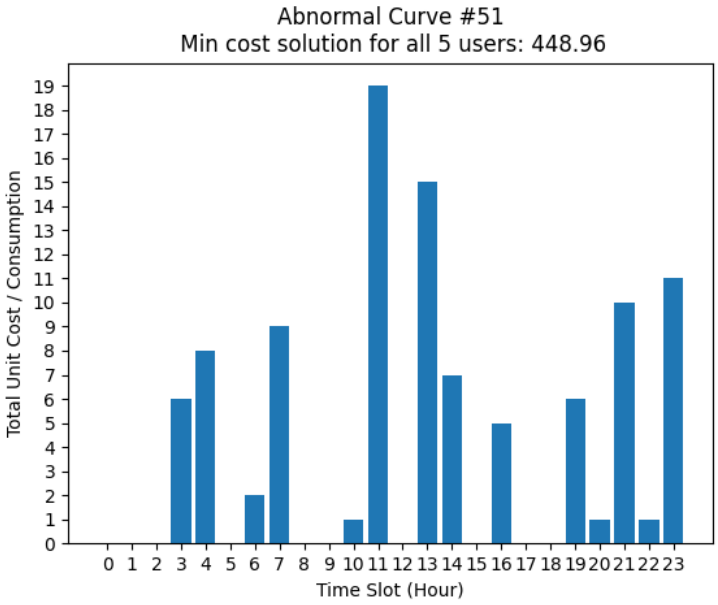

Abnormal Curve #38
Min cost solution for all 5 users: 448.64
Total Unit Cost / Consumption
Time Slot (Hour)

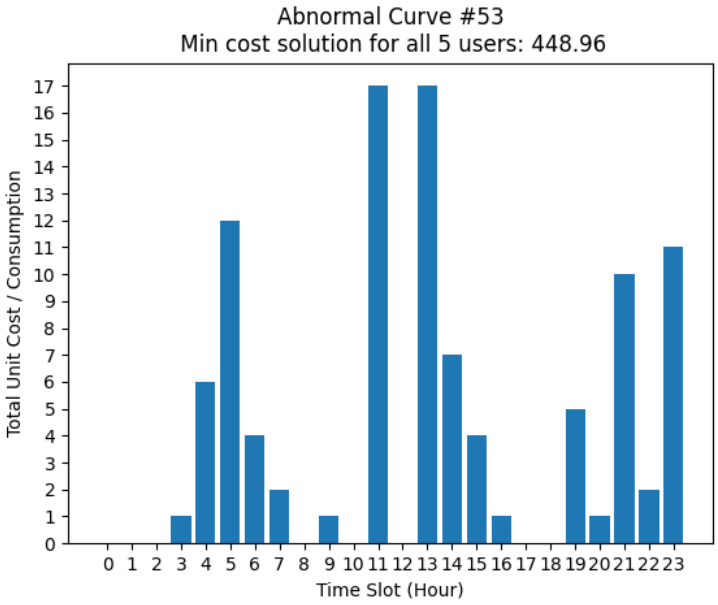

Abnormal Curve #39
Min cost solution for all 5 users: 448.65
Total Unit Cost / Consumption
Time Slot (Hour)

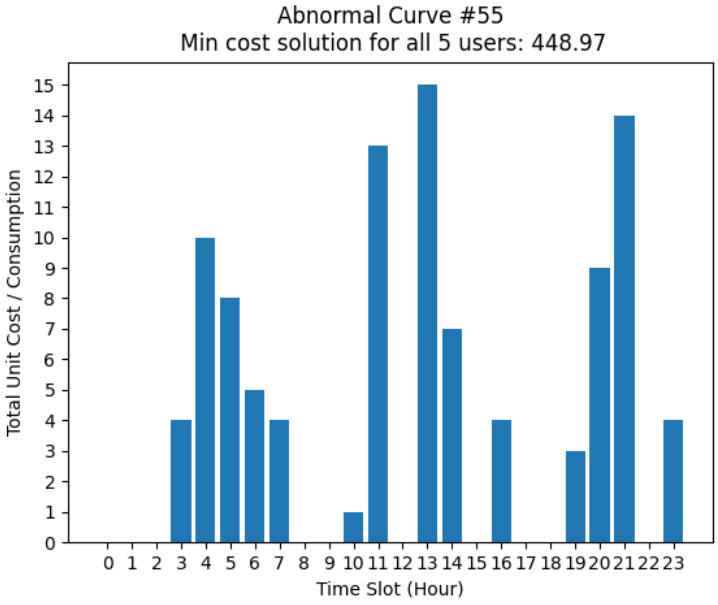

Abnormal Curve #46
Min cost solution for all 5 users: 448.66
Total Unit Cost / Consumption
Time Slot (Hour)

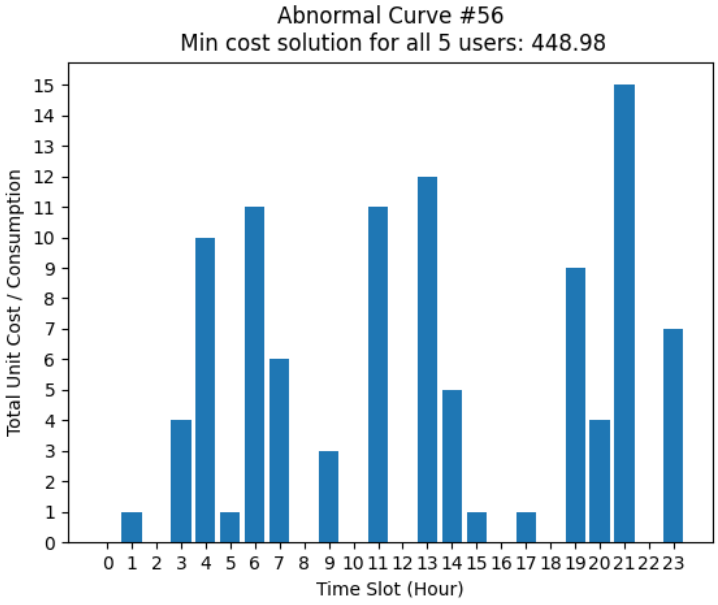

Abnormal Curve #48
Min cost solution for all 5 users: 448.66
Total Unit Cost / Consumption
Time Slot (Hour)

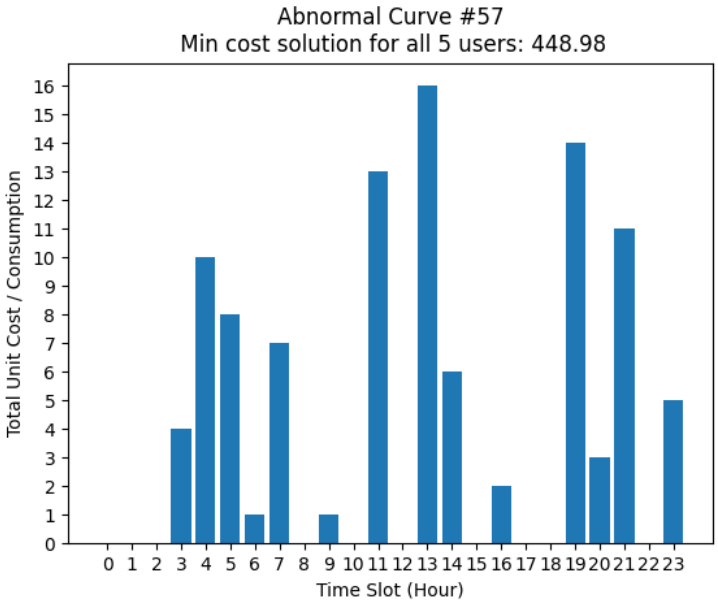

Abnormal Curve #49
Min cost solution for all 5 users: 448.66
Total Unit Cost / Consumption
Time Slot (Hour)

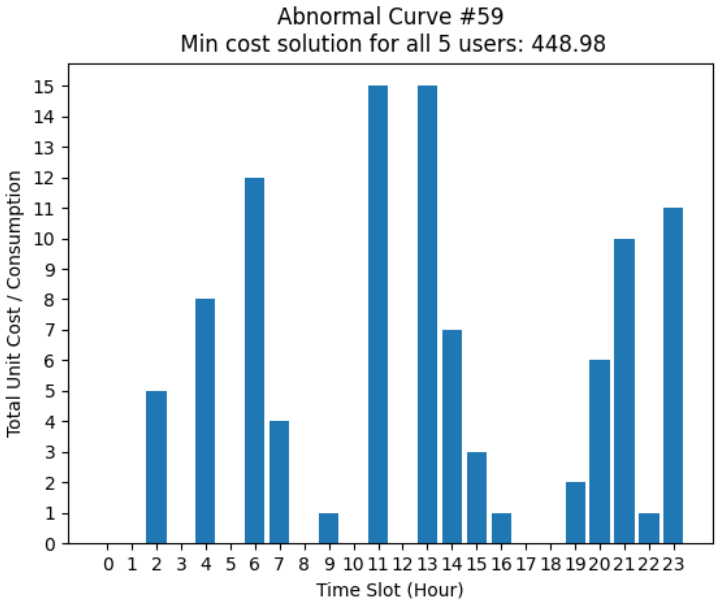

Abnormal Curve #51
Min cost solution for all 5 users: 448.96
Total Unit Cost / Consumption
Time Slot (Hour)

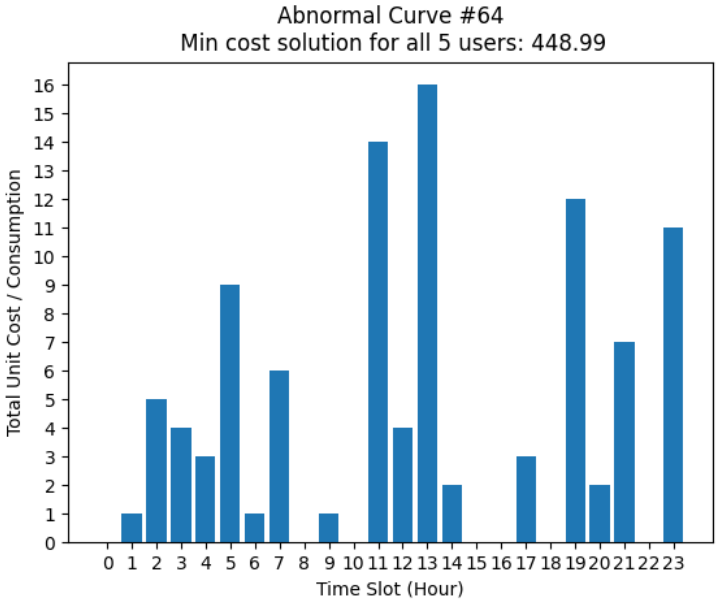

Abnormal Curve #53
Min cost solution for all 5 users: 448.96
Total Unit Cost / Consumption
Time Slot (Hour)

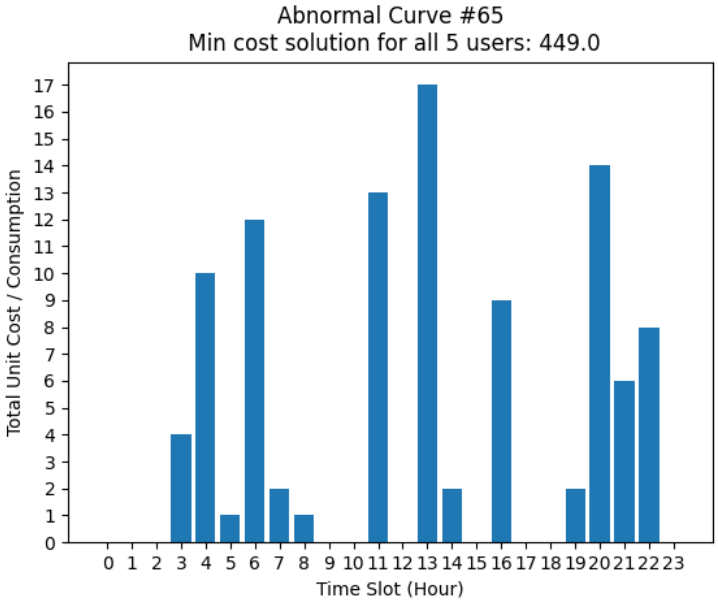

Abnormal Curve #55
Min cost solution for all 5 users: 448.97
Total Unit Cost / Consumption
Time Slot (Hour)

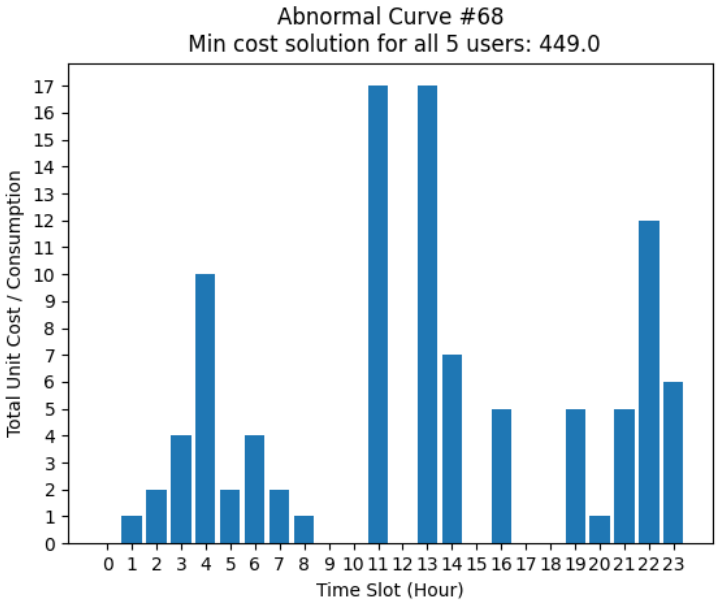

Abnormal Curve #56
Min cost solution for all 5 users: 448.98
Total Unit Cost / Consumption
Time Slot (Hour)

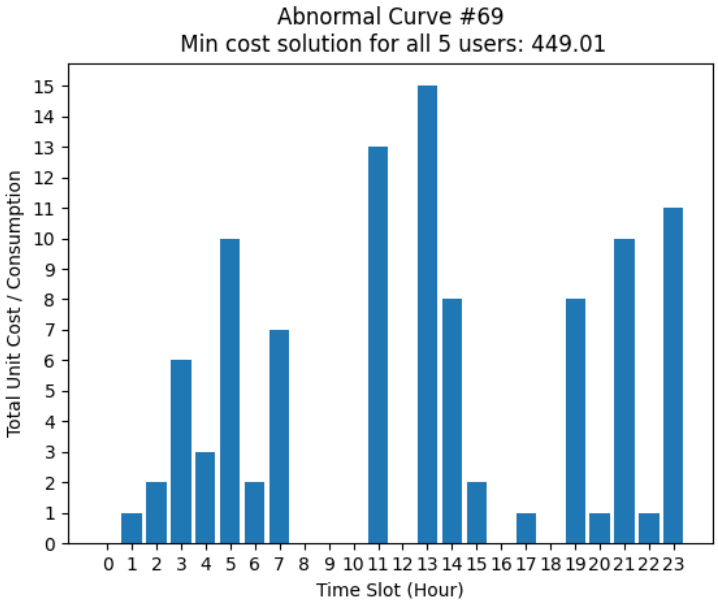

Abnormal Curve #57
Min cost solution for all 5 users: 448.98
Total Unit Cost / Consumption
Time Slot (Hour)

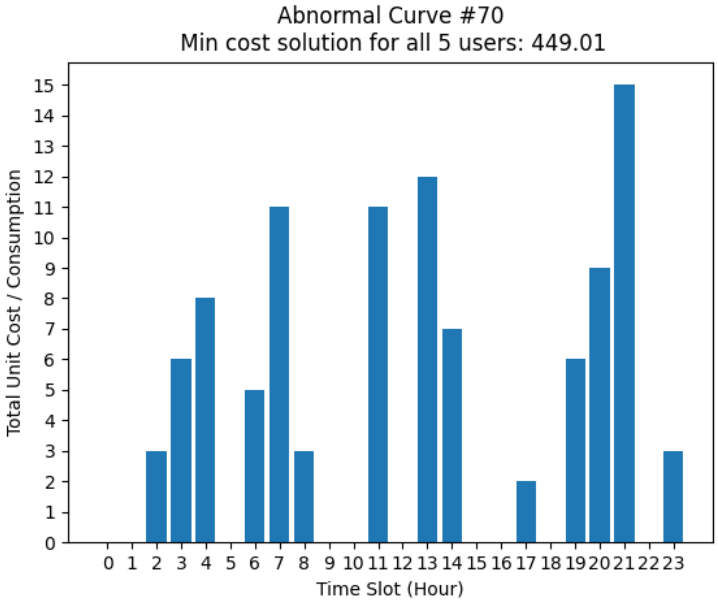

Abnormal Curve #59
Min cost solution for all 5 users: 448.98
Total Unit Cost / Consumption
Time Slot (Hour)

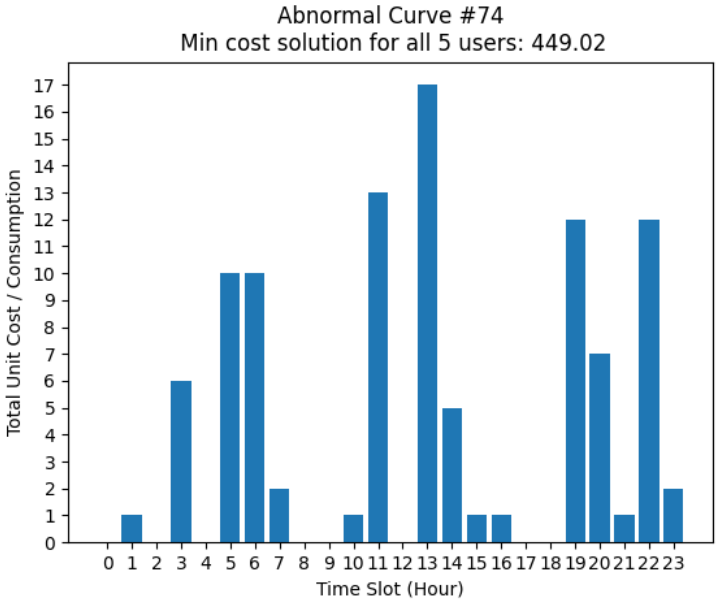

Abnormal Curve #64
Min cost solution for all 5 users: 448.99
Total Unit Cost / Consumption
Time Slot (Hour)

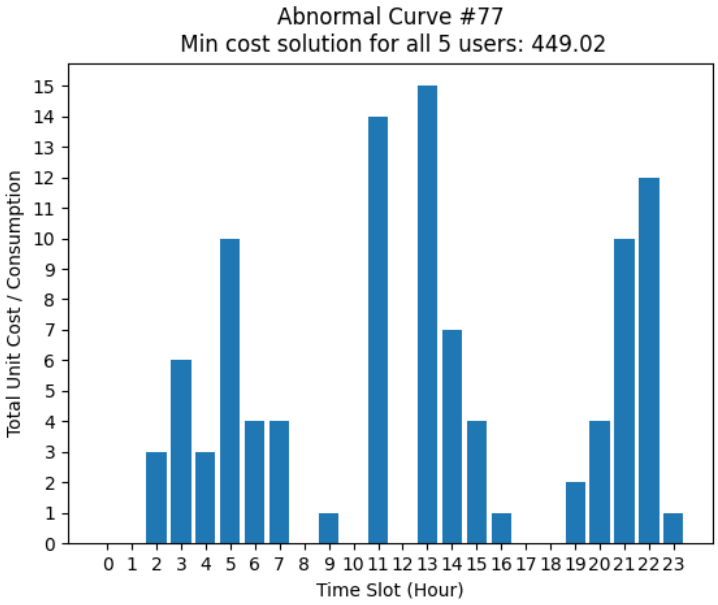

Abnormal Curve #65
Min cost solution for all 5 users: 449.0
Total Unit Cost / Consumption
Time Slot (Hour)

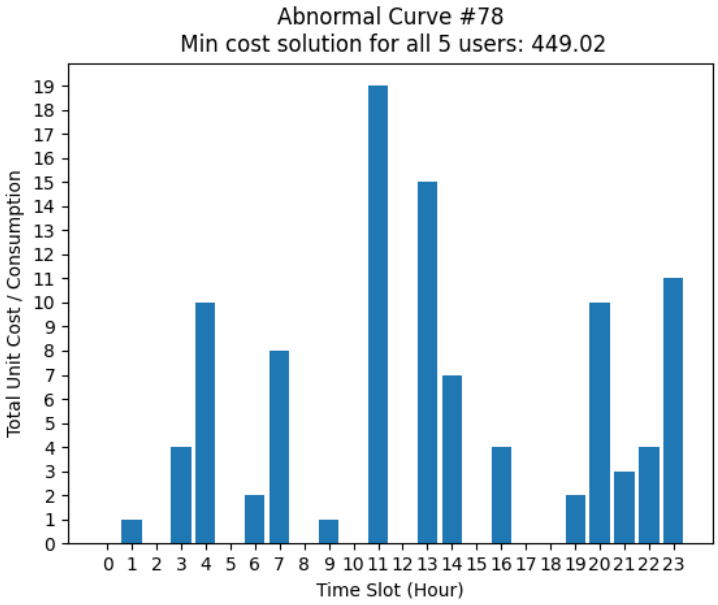

Abnormal Curve #68
Min cost solution for all 5 users: 449.0
Total Unit Cost / Consumption
Time Slot (Hour)

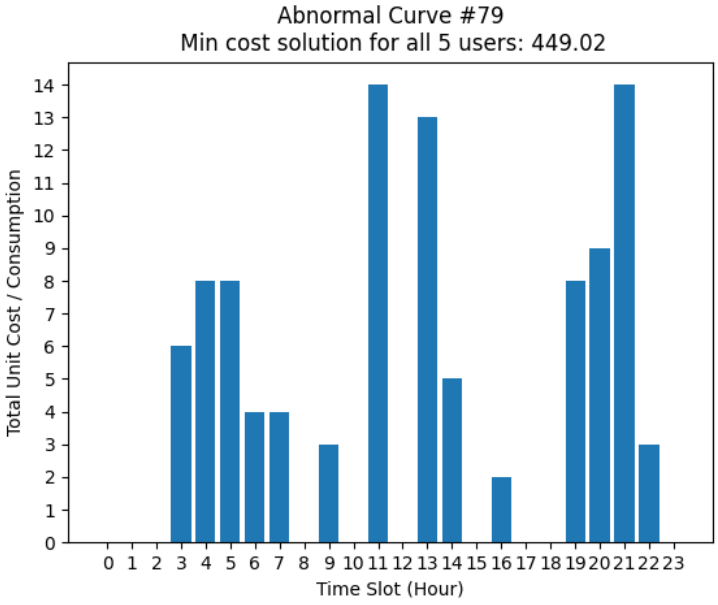

Abnormal Curve #69
Min cost solution for all 5 users: 449.01
Total Unit Cost / Consumption
Time Slot (Hour)

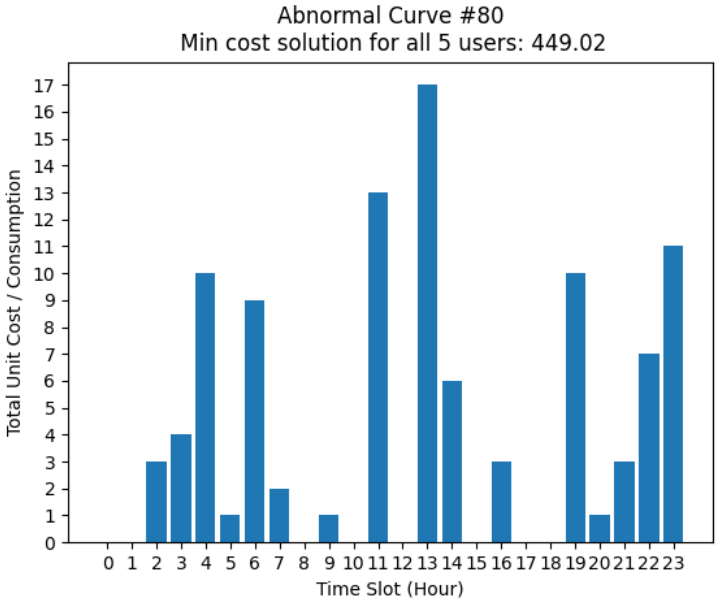

Abnormal Curve #70
Min cost solution for all 5 users: 449.01
Total Unit Cost / Consumption
Time Slot (Hour)

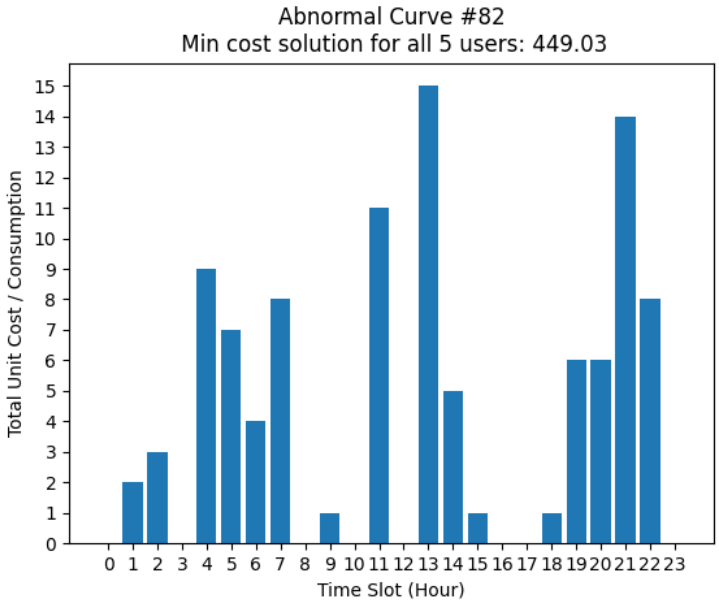

Abnormal Curve #74
Min cost solution for all 5 users: 449.02
Total Unit Cost / Consumption
Time Slot (Hour)

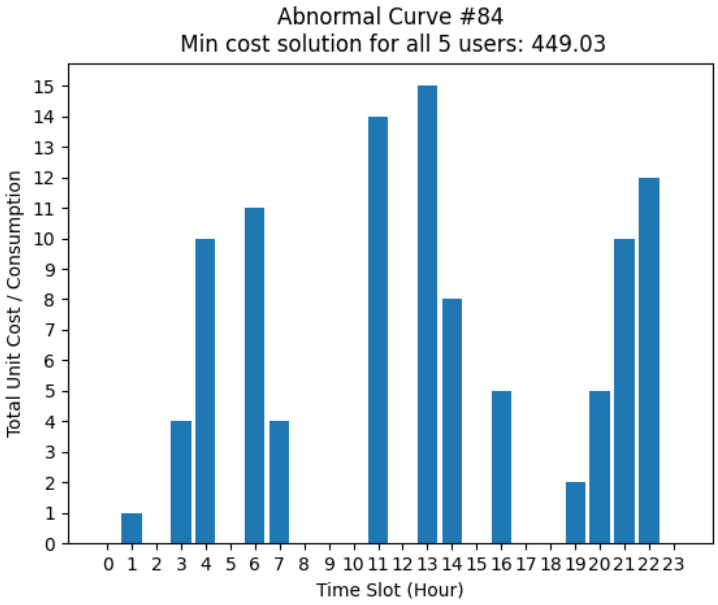

Abnormal Curve #77
Min cost solution for all 5 users: 449.02
Total Unit Cost / Consumption
Time Slot (Hour)

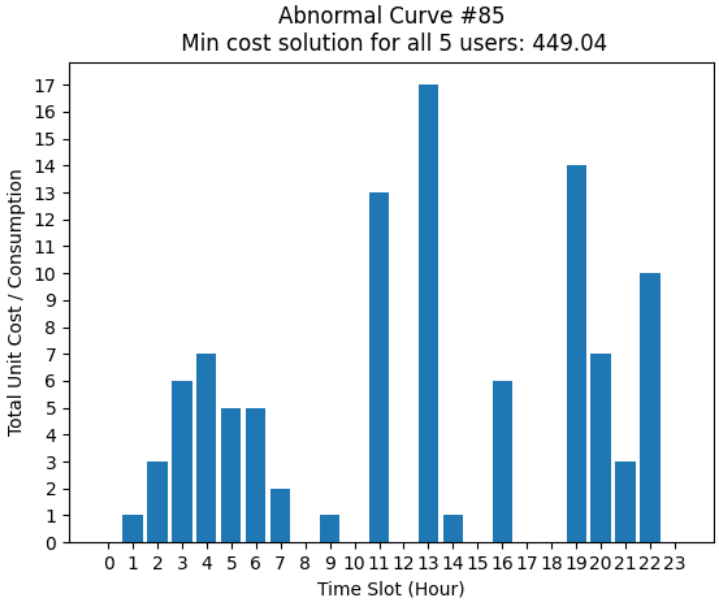

Abnormal Curve #78
Min cost solution for all 5 users: 449.02
Total Unit Cost / Consumption
Time Slot (Hour)

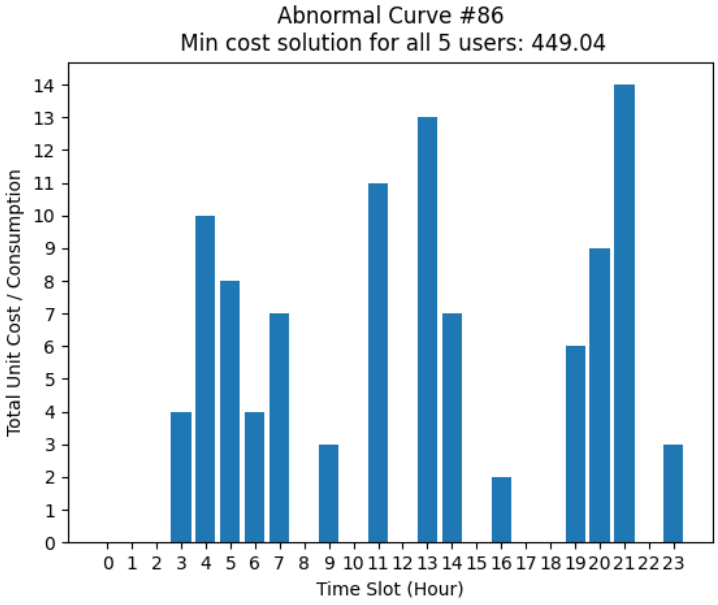

Abnormal Curve #79
Min cost solution for all 5 users: 449.02
Total Unit Cost / Consumption
Time Slot (Hour)

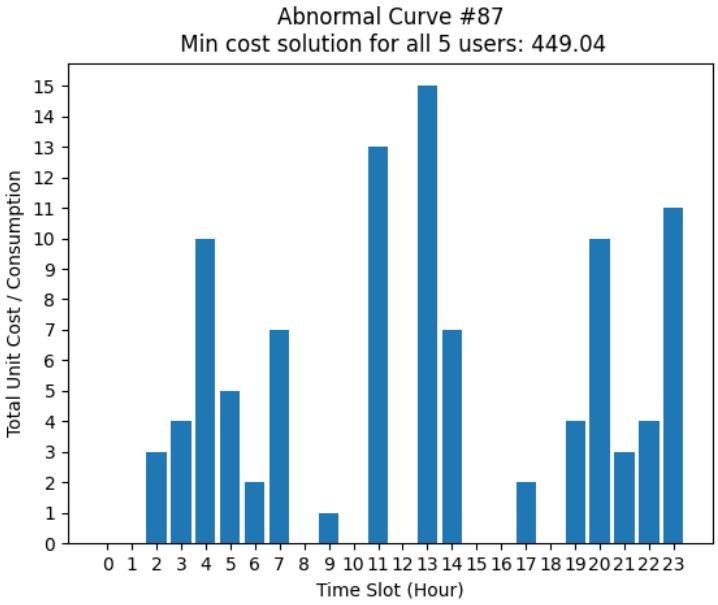

Abnormal Curve #80
Min cost solution for all 5 users: 449.02
Total Unit Cost / Consumption
Time Slot (Hour)

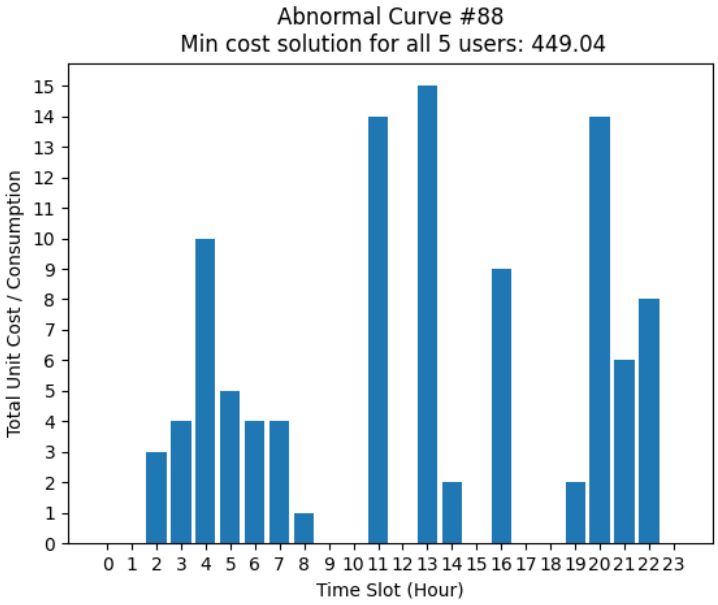

Abnormal Curve #82
Min cost solution for all 5 users: 449.03
Total Unit Cost / Consumption
Time Slot (Hour)

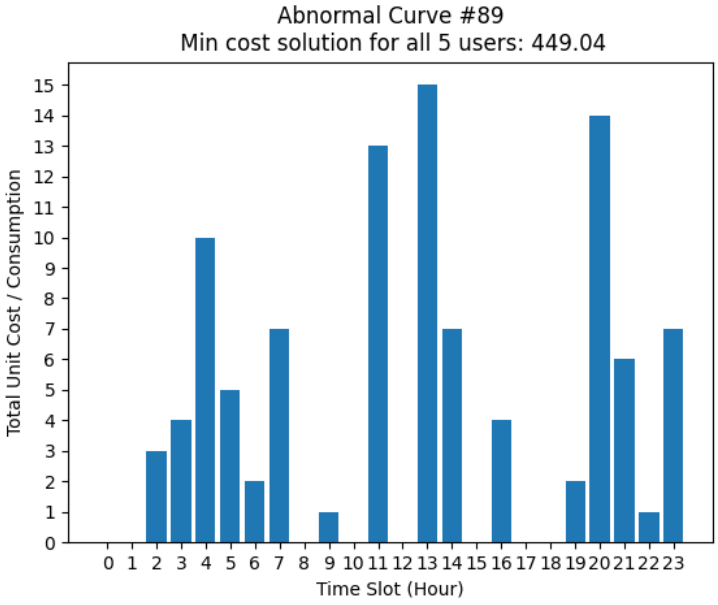

Abnormal Curve #84
Min cost solution for all 5 users: 449.03
Total Unit Cost / Consumption
Time Slot (Hour)

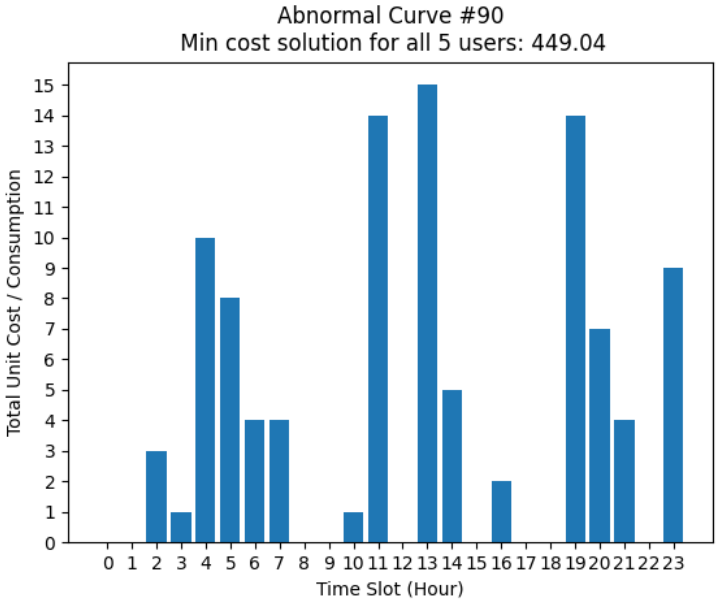

Abnormal Curve #85
Min cost solution for all 5 users: 449.04
Total Unit Cost / Consumption
Time Slot (Hour)

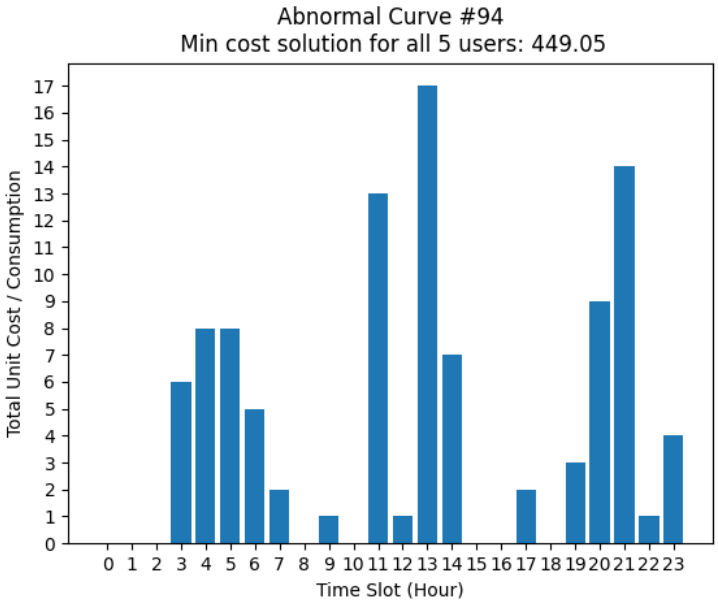

Abnormal Curve #86
Min cost solution for all 5 users: 449.04
Total Unit Cost / Consumption
Time Slot (Hour)

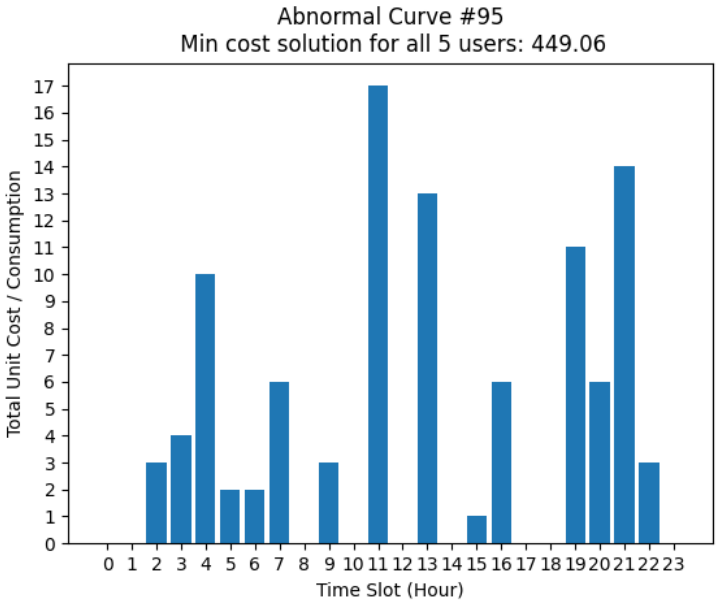

Abnormal Curve #87
Min cost solution for all 5 users: 449.04
Total Unit Cost / Consumption
Time Slot (Hour)

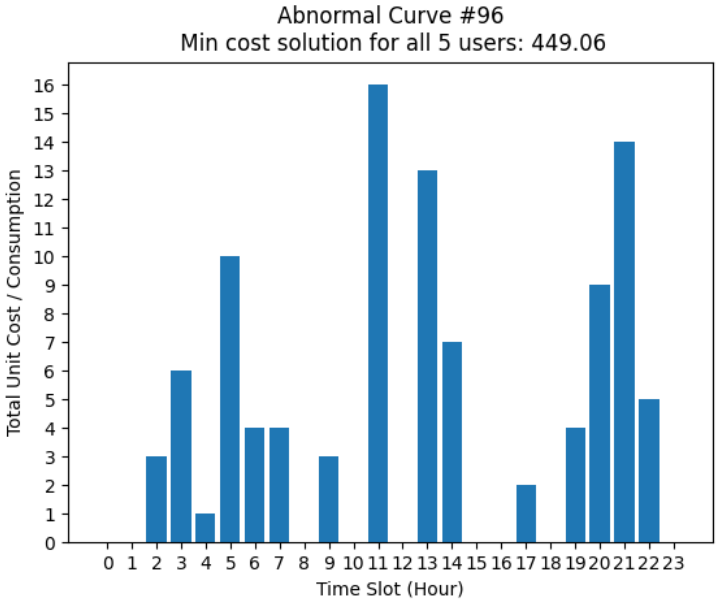

Abnormal Curve #88
Min cost solution for all 5 users: 449.04
Total Unit Cost / Consumption
Time Slot (Hour)

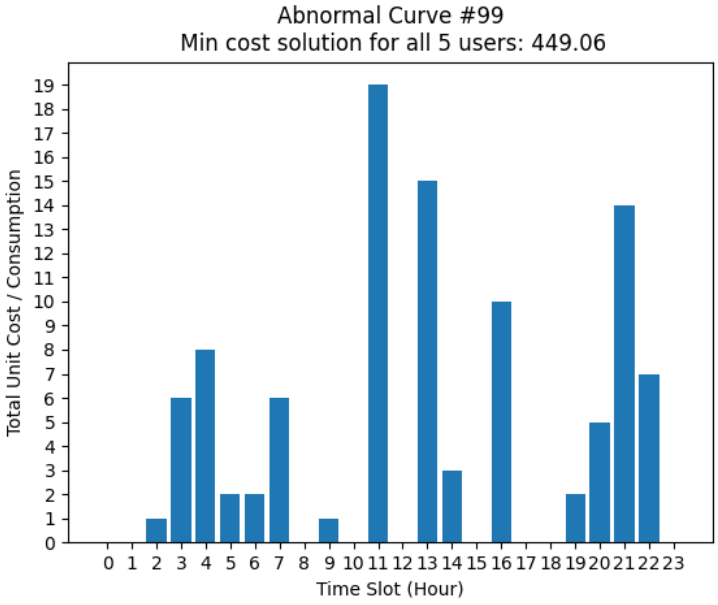

Abnormal Curve #89
Min cost solution for all 5 users: 449.04
Total Unit Cost / Consumption
Time Slot (Hour)


Abnormal Curve #90
Min cost solution for all 5 users: 449.04
Total Unit Cost / Consumption
Time Slot (Hour)


Abnormal Curve #94
Min cost solution for all 5 users: 449.05
Total Unit Cost / Consumption
Time Slot (Hour)


Abnormal Curve #95
Min cost solution for all 5 users: 449.06
Total Unit Cost / Consumption
Time Slot (Hour)


Abnormal Curve #96
Min cost solution for all 5 users: 449.06
Total Unit Cost / Consumption
Time Slot (Hour)


Abnormal Curve #99
Min cost solution for all 5 users: 449.06
Total Unit Cost / Consumption
Time Slot (Hour)