

Введение

Основная идея на котором будет базироваться решение то, что в разные промежутки времени существует разных характер потребления данных. Затем с помощью методов интерполяции будут восстановлены пропущенные точки.

Также берется за основу то, что данные для обучения подобраны качественные, так как здесь не будет учитываться праздничные дни, мировые катастрофы и тд, в период которых характер потребления интернет ресурсов меняется. И также делается предположение, что предоставленная статистика собрана с одного региона, где один часовой пояс.

Модель данных

Модель будет базироваться на предоставленном графике.

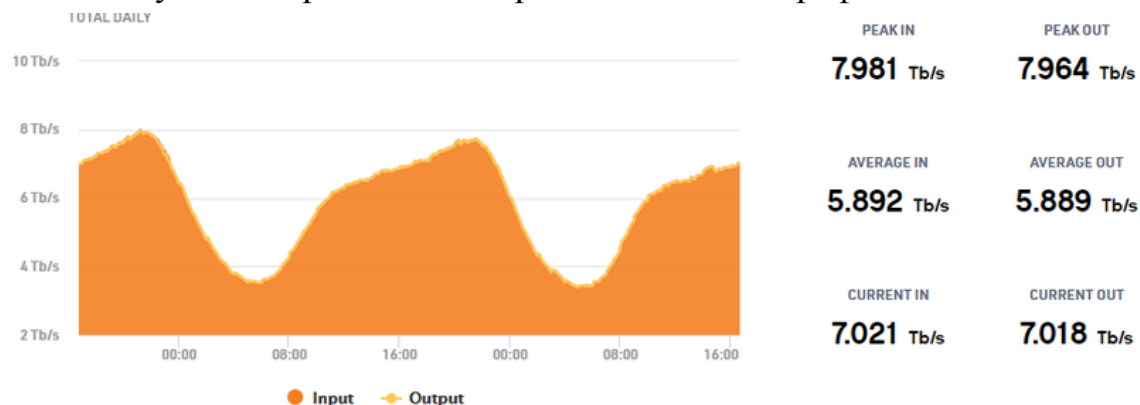


Рис1. Суточное изменение (пульсация) интернет-трафика провайдера

Исходя из графика можно предположить следующий характер данных

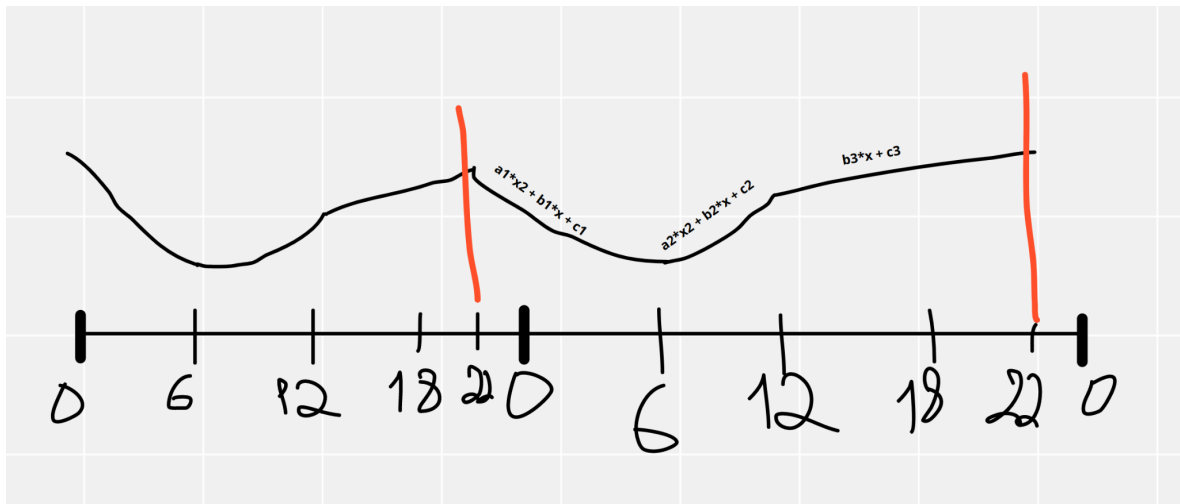


Рис2. Модель трафика данных

Модель будет выглядеть следующим образом:

$$22:00-6:00 - a_1 \cdot x^2 + b_1 \cdot x + c_1$$

$$6:00-12:00 - a_2 \cdot x^2 + b_2 \cdot x + c_2$$

$$12:00-22:00 - b_3 \cdot x + c_3$$

Дальше осталось найти только нужные коэффициенты. Для этого будем использовать метод наименьших квадратов. Перед аппроксимацией все данные будут слиты в один день.

После того, как коэффициенты будут найдены следует применить заданную погрешность из задания при подсчете ответа.

Инструменты для решения

Для решения целесообразно выбрать язык python, так как он содержит много открытых библиотек для аппроксимации. Предлагается использовать библиотеку `scipy` и `numpy`.

Реализация

```
from scipy.optimize import curve_fit

# Load data
data # data loaded from csv and fitted within one day
firstPartData # 22:00-6:00
secondPartData # 6:00-12:00
thirdPartData # 12:00-22:00
```

```
# Find coeffs
def linearRegression(x, a, b):
    y = a*x + b
    return y

def squareRegression(x, a, b, c):
    y = a*x*x + b*x + c
    return y

firstPart = curve_fit(squareRegression, xdata = firstPartData, ydata =
y)[0]
secondPart = curve_fit(squareRegression, xdata = secondPartData, ydata =
y)[0]
thirdPart = curve_fit(linearRegression, xdata = thirdPartData, ydata =
y)[0]
print(firstPart, secondPart, thirdPart)
```

Тестирование

Для тестирования нужны тестовые данные, которые не были предоставлены.

Заключение

В данном задании была построена модель трафика, но со многими допущениями. Из за нехватки времени не были подобраны полноценно коэффициенты для модели.