รายงานหลักการทำงานของ Algorithm ในการแบ่งบรรทัดและตรวจจับตัวอักษร

จุดประสงค์ในการสร้าง Algorithm และ Function ต่างๆ เพื่อจะสามารถตรวจจับบรรทัด และตัวอักษรภายใต้ข้อความที่เขียนจาก Software Android ที่ชื่อ Quill Editor ซึ่งเป็น Application สำหรับการเขียนจดบันทึกบน Android

หลักการทำงานโดยคร่าวๆ จะทำงานโดยการวนลูปตัวแปร Y ทุกตัวที่ได้รับมา จากข้อมูลที่เก็บบันทึกมาจาก Application โดยเมื่อโปรแกรมทำงาน และมีการเขียนโดยลงปากกาไป 1 ครั้งจะนับการลงปากกาในครั้งนั้นไป 1 Segment และตัวแปร Y คือตัวแปรที่เป็นตัวแทนของแนวแกน Y จากการเขียนลงบน Application

ขั้นตอนหลักๆ ในการพัฒนา Algorithm ในแบบตั้งต้นแบ่งเป็น 5 ส่วนดังนี้

1. **แก้ไขโค้ด Python ของ Application Quill เพื่อลด Output ของ Text ที่ได้มาจาก Application**

```
(0.28892615437507629, 0.1283198744058609, 1.0)
(0,)
Version:2
255, 0, 0, 0
2
1
No of stroke:41
(0.31056588888168335, 0.22108486294746399, 1.0)
(0,)
(0.30854880809783936, 0.21931847929954529, 1.0000001192092896)
(0,)
```

*รูปที่ 1ตัวอย่าง **Output** ก่อนแก้โค้ด*

โดย Output ที่ลบออกไป คือ Output ที่เป็น timestamp และตัวเลขอื่นๆที่สุ่มเพิ่มออกมา โดยจะกำหนดให้ Output ทั้งหมดที่ออกมาจะเหลือแค่ จำนวนของ Stroke ที่ตวัดลงไปทั้งหมดว่ามีจำนวนในการตวัดทั้งหมดกี่ครั้ง ซึ่งจะนับว่าเป็น 1 Segment และในแต่ละ Segment จะประกอบไปด้วยส่วนประกอบหลักๆคร่าวๆ 4 ส่วนคือ จำนวนการตวัดปากกาในการกดลงไปแต่ละครั้ง และพิกัดแกน X, Y, Z แบ่งตาม comma ซึ่งเป็นพิกัดของการตวัดปากกาลงไปในครั้งนั้น โดย X จะหมายถึงพิกัดในแนวแกน X, Y หมายถึงพิกัดในแนวแกน Y และ Z จะหมายถึงความลึกในการกดปากกาไปในแต่ละครั้ง

```
No of stroke:29
(0.20406192541122437, 0.10483694076538086, 0.31640625)
(0.20403799414634705, 0.10624153912067413, 0.55485141277313232)
(0.20524875819683075, 0.10722740739583969, 0.5986943244934082)
(0.2066313624382019, 0.1073647066950798, 0.62639313936233521)
(0.20880044996738434, 0.10674357414245605, 0.65767967700958252)
```

รูปที่ 2 ตัวอย่างของ Output หลังแก้ไขโค้ดใน Python

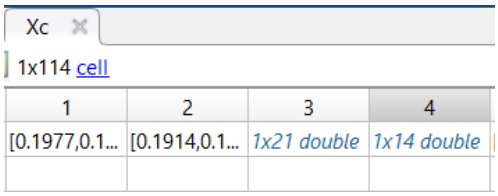
2. เขียน function readtfile.m เพื่อ return ค่าออกมาจาก text file

Output ที่ได้จาก function นี้มีทั้งหมด 5 outputs คือ Xc, Yc, Zc, txt_list และ segCount

```
%% readtfile.m
function [Xc, Yc, Zc, segCount, txt_list] = readtfile(tfile)
% READTFILE function will create a list of text coordinate
% which are x, y and z coordinate of text from quill editor
% and will output it cell and it segment count.
```

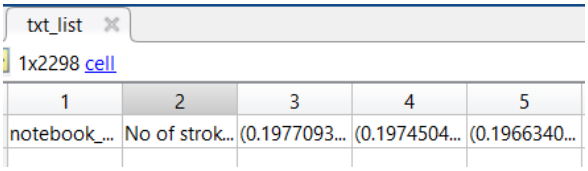
รูปที่ 3 Outputs และ parameter ของ function readtfile.m

- **Xc** : ค่าของตัวอักษรในแนวแกน X ทั้งหมดโดยจะถูกจัดเก็บเป็น cell array ใน matlab โดยที่ในแต่ละ cell column จะประกอบไปด้วยการตวัดปากกาในแนวแกน X แต่ละครั้ง



รูปที่ 4 cell array ที่เป็น output

- **Yc** : ค่าของตัวอักษรในแนวแกน Y ซึ่งสัมพันธ์กับค่าของตัวอักษรในแนวแกน X
- **Zc** : ค่าความลึกของการกดปากกาในแต่ละครั้ง
- **txt_list** : Cell array ที่เก็บค่าของ Text ทุกบรรทัดจาก file text ไว้ใน cell array แต่ละ column



รูปที่ 5 cell array ที่เก็บค่าแต่ละบรรทัดจาก text file

- **segCount** : นับจำนวน Segment ทั้งหมดของ text file ว่ามีทั้งหมดกี่ segment ซึ่งในแต่ละ segment จะเป็นการกดปากกาและยกออกในแต่ละครั้ง หรือก็คือเป็นการนับว่ามีตัวอักษรทั้งหมดกี่ตัวจากการนับการกดปากกาลงไป

3. เขียน Function LineFind.m เพื่อหาบรรทัดจาก text file ที่ใช้ และพยายามแยกแต่ละบรรทัดให้ออก

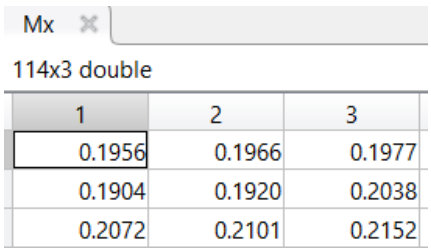
รับตัวแปร Xc และ Yci จาก main เพื่อนำมาคำนวณหาว่าเมื่อไหร่จะขึ้นบรรทัดใหม่ และเก็บค่าบรรทัดใหม่ไว้เพื่อบอกว่า ณ พิกัดนั้นๆ นับเป็นบรรทัดใหม่แล้ว

```
function [sepline, l] = LineFind(Xc, Yci)
% find min and max of y
```

รูปที่ 6 Outputs และ parameters ของ function LineFind.m

หลักการทำงานของ Algorithm ที่ใช้ในการแบ่งบรรทัด

1. วนลูปเพื่อหาเก็บค่า Min, Max และ Median ของตัวแปร Xc และ Yc มาเก็บไว้ในตัวแปร Mx และ My



รูปที่ 7 เก็บค่า min, median และ max ไว้ใน column

2. **เมื่อได้ Min และ Max แล้วก็ตั้งไว้ว่าค่า Min และ Max ของแกน Y เปรียบเสมือน boundary กั้นไว้ หากค่าไหนที่เยอะ Boundary ของบรรทัดนั้นทั้งหมดแสดงว่า อาจจะไม่ได้อยู่ในบรรทัดเดิมแล้ว**

```
miny = My(1, 1); % act as a first threshold (min)
maxy = My(1, 3); % act as a first threshold (max)
Ycvalue = [];
```

*รูปที่ 8 ใส่ **boundary** ไว้สำหรับแกน **X** และ **Y** ไว้คร่าวๆ*

3. **วน for loop ตามจำนวน Segment ทั้งหมดของตัวแปร X หรือ Y เพื่อหาว่ายังอยู่ในบรรทัดนั้นๆมั้ย จาก segment ที่มี**
   - ถ้าค่า min ของ y ที่ segment นั้นๆน้อยกว่าค่า max ของ segment นั้นๆอยู่ 0.3 จะสันนิษฐานว่าค่า y ณ segment นั้นๆมี outlier อยู่ซึ่งอาจจะเกิดจากการที่มีการพลาดแล้วมือไปโดน เพราะฉนั้นเราจะไม่สันนิษฐานว่านั่นเป็นบรรทัดใหม่และยังไม่เปลี่ยนบรรทัด

```
for i = 2:lx % start from the second Y value

    % the condition if it still in the same line or next
    if My(i, 1) - maxy >= 0.3
        newline = 0; % this mean to sep outlier first
        miny = My(i, 1);
```

รูปที่ 9 วนลูปเช็คว่ายังอยู่บรรทัดเดิมอยู่รึเปล่า

   - แต่ถ้าเป็นอย่างอื่นก็ยังมีอีก 2 กรณีคือ เป็นบรรทัดใหม่ หรือก็เป็นบรรทัดสุดท้ายแล้ว
     - **กรณีเป็นบรรทัดใหม่**

       ถ้าค่า max ของ Y ณ segment นั้นๆน้อยกว่าค่า min ของ segment ที่แล้วอยู่ 0.01 เราจะสันนิษญานว่านั่นเป็นบรรทัดใหม่ แต่ยังไม่เปลี่ยนบรรทัด เพราะยังไม่แน่ใจ จึงทำแบบเดิมไปอีกตัวนึงว่าถ้าค่า max ของ ณ segment อันต่อไปยังคงน้อยกว่าค่า min เดิมอยู่ เราจึงจะมั่นใจได้ว่าเป็นการขึ้นบรรทัดใหม่แล้วจริงๆ จึงเปลี่ยนไปยังบรรทัดต่อไป และเก็บค่า ณ segment นั้นๆไว้ที่ตัวแปร sepline เพื่อเป็นการบอกว่าการเปลี่ยนเป็นบรรทัดใหม่ที่เกิดขึ้น อยู่ที่ segment ใด

```
else
    if My(i, 3) < miny - 0.01
        newline = newline + 1;
        if newline > 2 % if newline is greater than 3 that mean
                       % we are so sure that that is a new line
            l = l+1;
            newline = 0; % restart parameter if sure its a new line
            miny = My(i-2, 1);
            LineNo(i-2: i, 1) = l;
            sepline = horzcat(sepline, i-2);
```

รูปที่ 10 วนลูปอีกรอบ ในกรณีที่อาจจะปเลี่ยนเป็นบรรทัดใหม่แล้ว

- **กรณีเป็นบรรทัดสุดท้ายแล้ว**

  กรณีที่เป็นบรรทัดสุดท้ายแล้ว เราจะใส่กรณีนี้ไว้ใน else เลยและนับค่าเพื่อบอกว่าไม่มีบรรทัดที่อยู่น้อยกว่านี้อีกแล้ว และเก็บค่าที่เป็นบรรทัดสุดท้ายไว้เพื่อนำมาตรวจสอบในภายหลัง

4. **นำค่าของตัวอักษรที่เราสันนิษฐานว่าเป็นบรรทัดใหม่ทุกตัว มาใส่ไว้ใน list เพื่อเก็บค่านั้นๆไว้บอกไว้ว่าพิกัด X และ Y ณ segment นั้นๆถูกนับเป็นตัวขึ้นต้นบรรทัดใหม่**

```
sepline = horzcat(sepline, lx(end))
```

รูปที่ 11 Output ที่เป็น list เพื่อบอกว่าบรรทัดใหม่เริ่มต้นที่ segment ใด

4. **Function ที่ใช้ในการแยกแต่ละบรรทัดออกจากกัน เพื่อนำไปใช้แยกหาแต่ละตัวอักษรต่อไป**

เมื่อโปรแกรมรู้แล้วว่ามีทั้งหมดกี่บรรทัด และรู้ว่าแต่ละบรรทัดเริ่มต้นที่ segment ใด จึงเขียน function linesegment.m ขึ้นมาเพื่อแยกทุกบรรทัดออกจากกันและเก็บค่าของ segment X, Y และ Z แต่ละบรรทัดไว้ใน row cell array โดยแต่ละ row ใน cell จะหมายถึงแต่ละบรรทัด

```
function [SegX, SegY, SegZ] = linesegment(l, sepline, Xc, Yci, Zc)
SegX = cell(l);
SegY = cell(l);
SegZ = cell(l);
```

รูปที่ 12 Outputs และ parameters ของ function linesegment.m

การทำงานของ algorithm ที่ใช้ในการแยกบรรทัดถูกทำงานด้วยการวน for loop ตามจำนวนบรรทัดที่หาได้และใส่ไว้ในตัวแปร ‘l’ แบ่งเป็น 3 กรณีคือบรรทัดแรก บรรทัดสุดท้าย และบรรทัดอื่นๆ

- **บรรทัดแรก**

วน for loop และในกรณีที่ i=0 หรือเป็นบรรทัดแรก และวน for loop ต่ออีกครั้งจนถึงค่าของ segment ตัวแรกที่ถูกเก็บไว้ในตัวแปร sepline และเก็บค่าที่ได้ไว้ใน row cell array

```
for i = 0:l
    % First Line Case
    if i == 0
        for j = 1:sepline(1)-1
            SegX{1, j} = Xc{j};
            SegY{1, j} = Yci{j};
            SegZ{1, j} = Zc{j};
        end
```

รูปที่ 13 กรณีบรรทัดแรก

- **บรรทัดสุดท้าย**

วน for loop ในกรณีที่ i=l(end)-1 (หรือค่าสุดท้าย) หรือบรรทัดสุดท้าย และวน for loop ต่ออีกครั้งจาก sepline ตัวก่อนสุดท้ายจนถึงตัวสุดท้าย เพื่อเก็บค่า segment X และ Y ทุกตัวไว้ใน row cell array ที่ row สุดท้าย และเมื่อถึงบรรทัดสุดท้ายจะ break เพื่อไม่ให้ function ทำงานต่อจากเดิมอีก

```
elseif i >= l(end)-1
    if i == l(end)-1
        for j = sepline(i):sepline(i+1)
            SegX{i+1, j-(sepline(i)-1)} = Xc{j};
            SegY{i+1, j-(sepline(i)-1)} = Yci{j};
            SegZ{i+1, j-(sepline(i)-1)} = Zc{j};
        end
    elseif i == l(end)
        break
    end
```

รูปที่ 14 กรณีบรรทัดสุดท้าย

- **บรรทัดอื่นๆ**

วนลูปตามปกติ และวนลูปอีกครั้ง ณ sepline ที่ i ไปจนถึง sepline ที่ i+1 เพื่อเก็บค่าระหว่างบรรทัดนั้นไว้ตาม row cell array แต่ละ row ซึ่งแต่ละ row จะหมายถึงแต่ละบรรทัด

```
% Normal Case
else
    for j = sepline(i):sepline(i+1)
        SegX{i+1, j-(sepline(i)-1)} = Xc{j};
        SegY{i+1, j-(sepline(i)-1)} = Yci{j};
        SegZ{i+1, j-(sepline(i)-1)} = Zc{j};
    end
```

รูปที่ 15 กรณีที่เป็นบรรทัดอื่นๆปกติ

SegX

6x27 cell

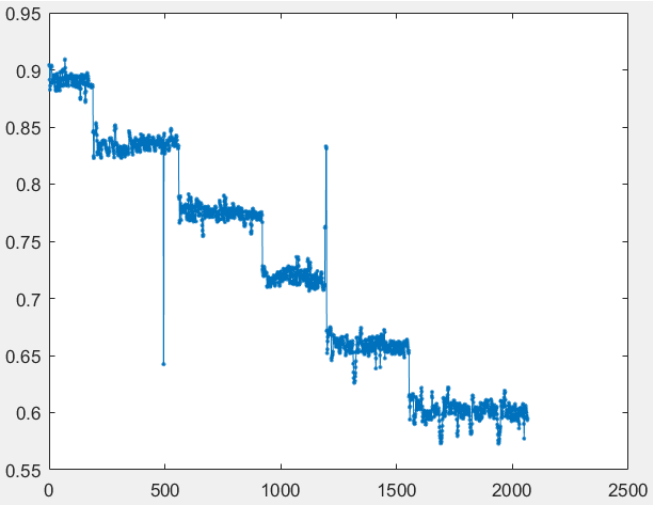
1	2	3	4	5
[0.1977,0.1...	[0.1914,0.1...	1x21 double	1x14 double	[0.2354,0.2...
[0.1963,0.1...	[0.1880,0.1...	1x22 double	[0.2217,0.2...	1x12 double
1x21 double	1x35 double	1x17 double	1x34 double	1x12 double
1x23 double	1x25 double	1x14 double	1x45 double	1x53 double
1x20 double	[0.1821,0.1...	[0.1999,0.1...	1x17 double	1x32 double
1x19 double	[0.1862,0.1...	[0.2037,0.2...	1x17 double	1x52 double

รูปที่ 16 ตัวอย่างผลลัพธ์ที่ได้จากการแบ่งแต่ละบรรทัดเก็บไว้ใน row cell array โดยในแต่ละ row จะหมายถึงแต่ละบรรทัด

5. Plot เพื่อสำรวจผลลัพธ์

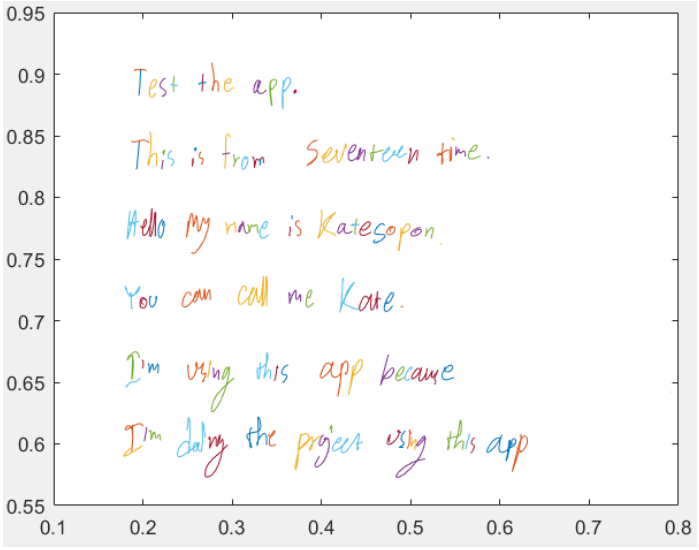
Plot กราฟเพื่อสำรวจผลลัพธ์จากขั้นตอนต่างๆ plot ในไฟล์ Mainpot.m จาก file text “6.txt”

5.1 Plot ดูการกระจายของตัวแปร Y เพื่อดูการเปลี่ยนแปลงของ Y ตามเวลาการเขียน



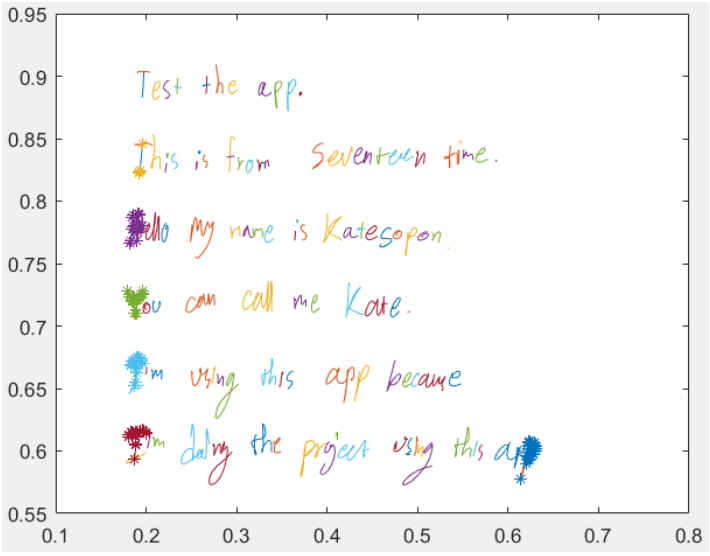
รูปที่ 17 Plot การกระจายตัวของ Y

5.2 Plot ข้อมูล X และ Y จากไฟล์ text ที่มาจาก quill



รูปที่ 18 Plot แกน X และ Y จากไฟล์

5.3 Plot ทับ figure(2) เป็นการ plot เพื่อดูว่าค่าที่หามาได้จาก algorithm ในการแยกบรรทัดแยกได้ดีแค่ไหน

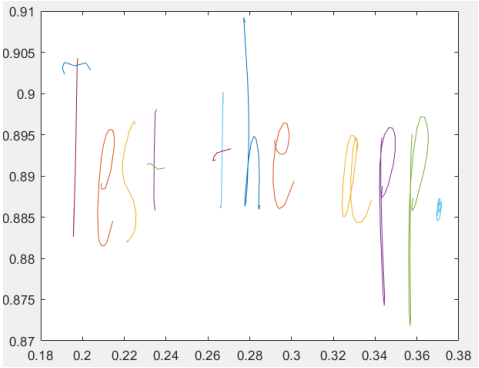


รูปที่ 19 Plot ทับ figure 2

จะเห็นได้ว่า plot ทับไว้ได้ดี แสดงว่า algorithm ที่ใช้ในการแบ่งบรรทัดสามารถใช้กับการแบ่งไฟล์นี้ได้

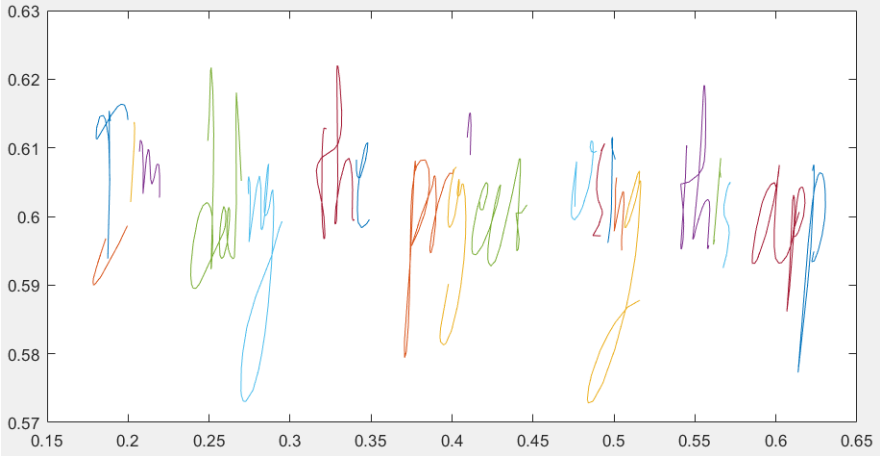
5.4 ทดสอบ plot บรรทัดแรกและบรรทัดสุดท้าย

5.4.1 บรรทัดแรก



*รูปที่ 20 **plot** บรรทัดแรก*

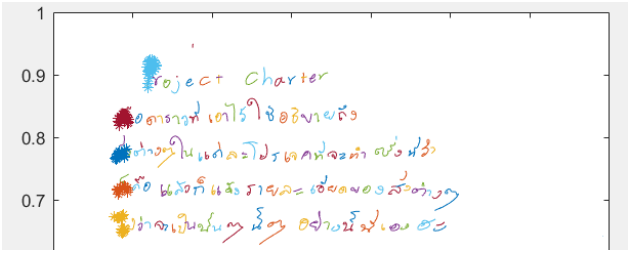
5.4.2 บรรทัดสุดท้าย



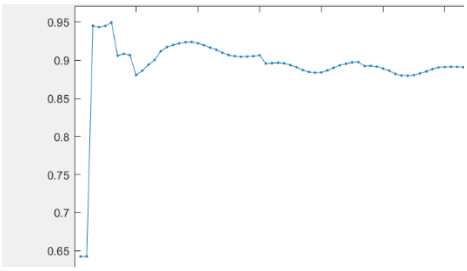
*รูปที่ 21 **Plot** บรรทัดสุดท้าย*

ปัญหาที่พบ

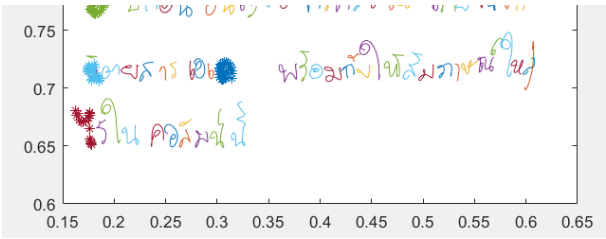
จากการเปลี่ยนไฟล์ไปมา จะเจอปัญหาจากบางไฟล์เช่นไฟล์ “7.txt” จะเจอปัญหาการเจอ outlier ตั้งแต่ตัวแรกเลยทำให้จับตัวอักษรตัวแรกคลาดเคลื่อน และไฟล์ “9.txt” ที่เจอว่าเหมือนมีการลบแล้วกลับมาเขียนใหม่ทำให้ตัว algorithm ไม่สามารถจับได้ว่า เป็น segment ที่อยู่ในบรรทัดก่อนหน้า



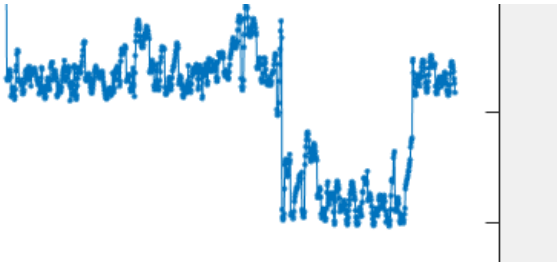
รูปที่ 25 รูปจากไฟล์ "7.txt"



รูปที่ 23 เส้นกราฟที่ plot ออกมาเพื่อตรวจสอบไฟล์ "7.txt"



รูปที่ 24 รูปภาพจากไฟล์ "9.txt"



รูปที่ 22 เส้นกราฟที่ plot ออกมาเพื่อตรวจสอบไฟล์ "9.txt"

สรุปผล

ผลที่ได้ออกมาได้ดีกว่ารอบที่แล้วมาก โดยจัดการปัญหาที่มี outlier ระหว่างบรรทัดซึ่งเกิดจากมือไปโดนระหว่างเขียน และปัญหาตัวอักษรภาษาอังกฤษที่อยู่ห่างๆกันออกไปได้

แต่ยังคงมีปัญหา หากมีการลบแล้วกลับมาเขียนตัวที่อยู่บรรทัดบนๆ เนื่องจากเวลาบันทึก text file ไว้และ import เข้ามาแก้ไขไม่ได้มีการบันทึก timestamp ออกมาด้วย และอีกปัญหาคือ เมื่อมีตัวแปรที่เป็น outlier ถูกกำกับไว้ที่ segment แรกจะทำให้จับว่าเป็น outlier ไม่ได้แล้วจะทำให้เวลาเก็บค่ามาใส่ cell ตัวที่เป็น outlier นั้นจะถูกนำเข้ามาเก็บด้วย