

Assignment 2

SPEECH SEGMENTATION TASK

Simon Chong Kai Yuen - A19EC3028

Clive Lai Yi Cheng - A19EC3019

Ong Yin Ren - A19EC0204

Ng Shu Yu - A19EC3031

Introduction

This assignment requires identifying the drawback of benchmark algorithm in **designing an algorithm** to segment the sound file accurately.

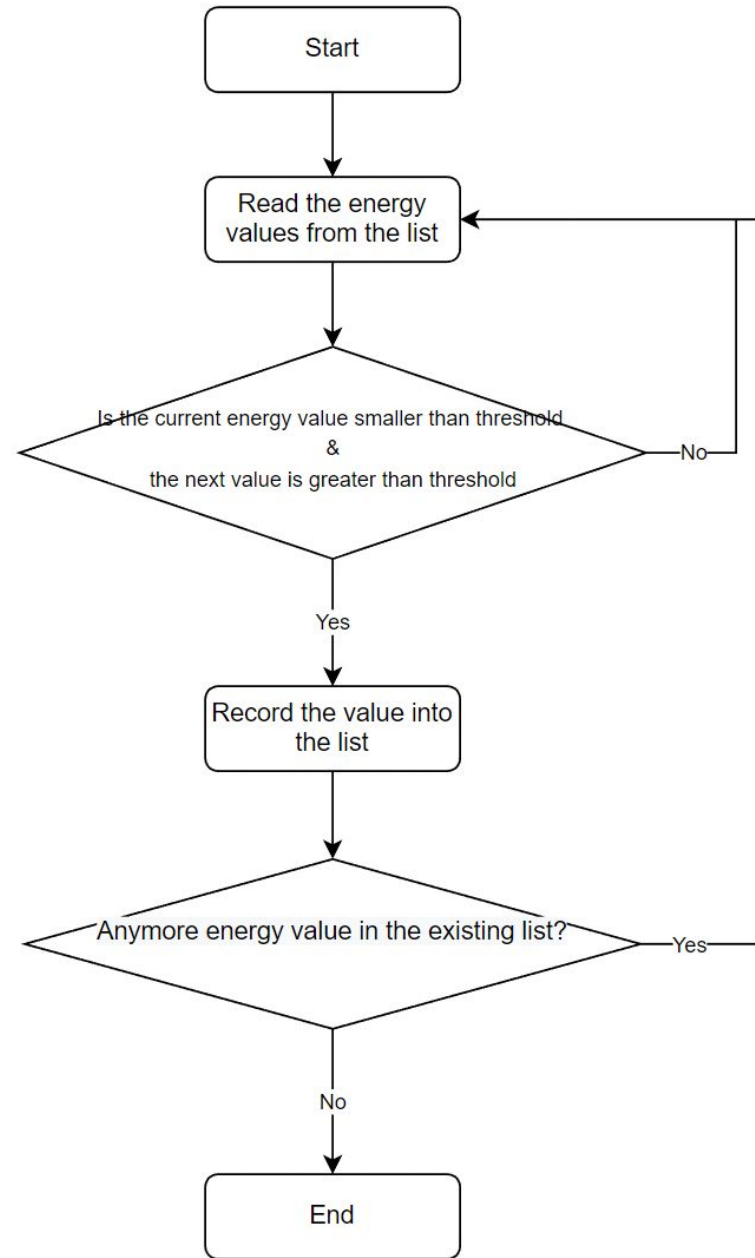
The given benchmark algorithm yield **a less promising result** due to it only check the incremental energy value which is crossing the threshold.

We **identify the issue and problem**, and come out with a **modified algorithm** that show a much more satisfying result in sound segmentation.

The data set

- Dataset used are Set-B And Set-G
- This dataset contains 20 sound patterns of connected digit, from 10 male speakers and 10 Female speakers
- These files are (0075, 1206, 2433, 3630, 4137, 5580, 6255, 7565, 8299, 9472)
- We found that the patterns have the best time tolerance, threshold and window size at 0.1, 1.4 and 250 respectively.

Benchmark Solution



Issue & Problem of the Benchmark Algorithm

The problem of the benchmark algorithm is it only identifies if the current energy value is smaller than the threshold and the next energy value is greater than the threshold (means incrementally crossing the threshold).

But it does not identify the energy value which is crossing the threshold decrementally by comparing if the current energy value is greater than the threshold and the next energy value is smaller than the threshold.

```
if (x(m)< Thres) & (x(m+1)>Thres)
    S(ind) = m-1;
    ind=ind+1;
end
```

Experimental Setup

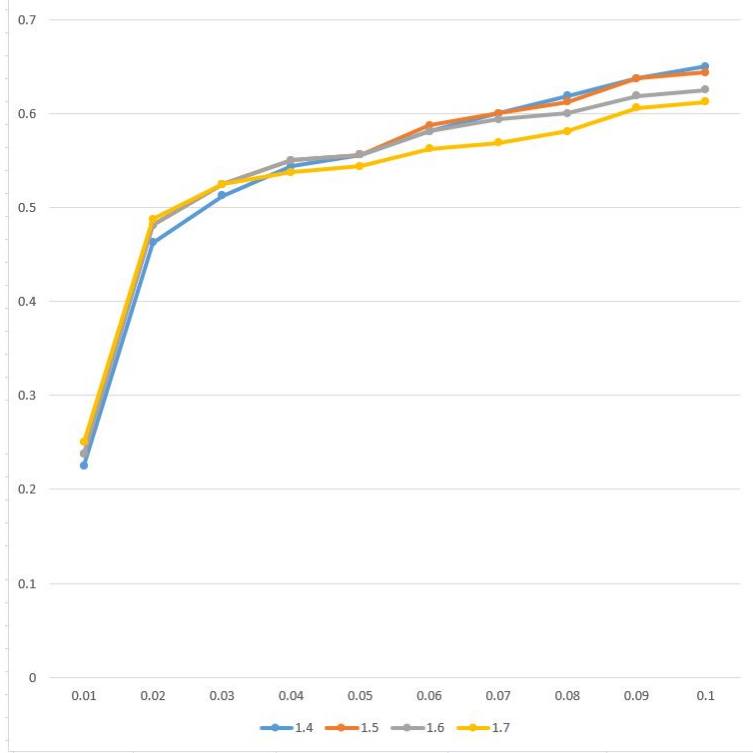
- Experimental Variable : Time tolerance, Threshold and win size
- Time tolerance: $t = \{0.00, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10\}$
- Threshold : $\text{Thres} = \{1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2\}$
- Win size : $W = \{250, 300, 350\}$
- The same set-up is used for the benchmark algorithm.
- Comparison between the proposed and benchmark are measure based on
 - Performance indicator: High match rate, low insertion and omission rate

* The values here are for examples. You may use them or choose your own that suit your solution

Issues & Problem (continue)

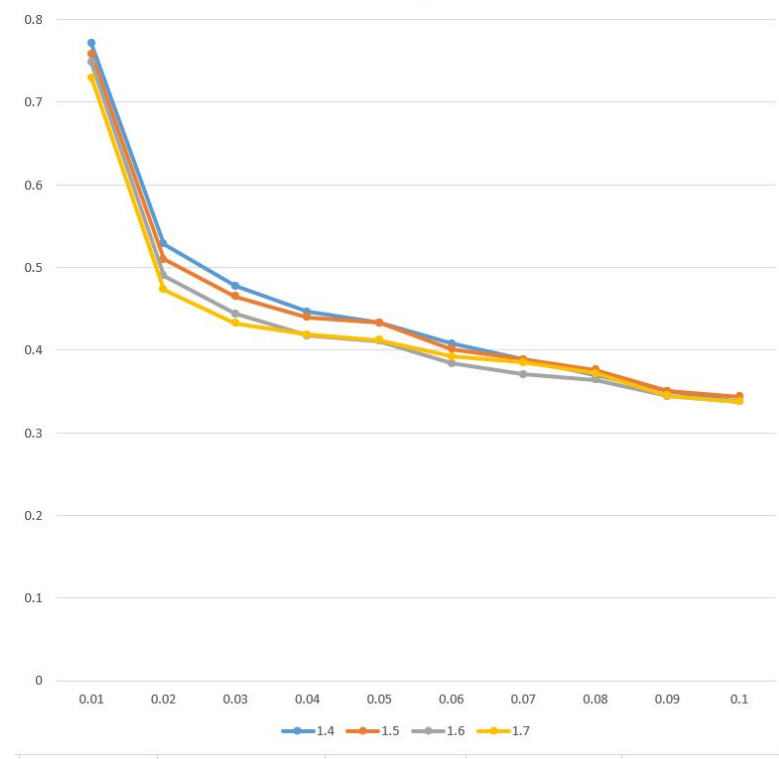
Match	Threshold	Threshold	Threshold	Threshold
	1.4	1.5	1.6	1.7
0.01	0.225	0.2375	0.2375	0.25
0.02	0.4625	0.4813	0.4813	0.4875
0.03	0.5125	0.525	0.525	0.525
0.04	0.5437	0.55	0.55	0.5375
0.05	0.5563	0.5563	0.5563	0.5437
0.06	0.5813	0.5875	0.5813	0.5625
0.07	0.6	0.6	0.5938	0.5687
0.08	0.6188	0.6125	0.6	0.5813
0.09	0.6375	0.6375	0.6188	0.6062
0.1	0.65	0.6438	0.625	0.6125

Match Rate by Threshold



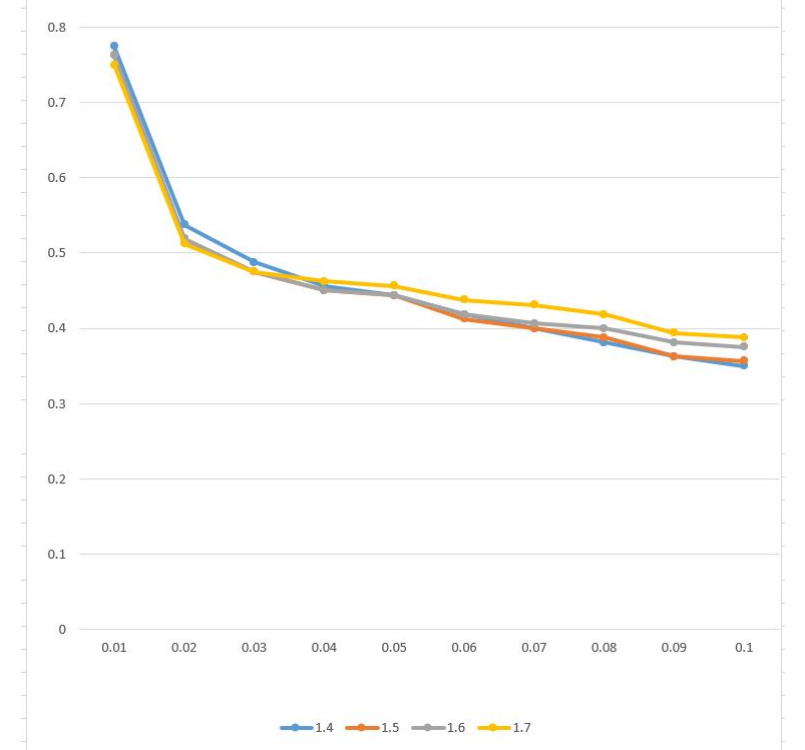
Insertion	Threshold	Threshold	Threshold	Threshold
	1.4	1.5	1.6	1.7
0.01	0.7707	0.758	0.7483	0.7297
0.02	0.5287	0.5096	0.4901	0.473
0.03	0.4777	0.465	0.4437	0.4324
0.04	0.4459	0.4395	0.4172	0.4189
0.05	0.4331	0.4331	0.4106	0.4122
0.06	0.4076	0.4013	0.3841	0.3919
0.07	0.3885	0.3885	0.3709	0.3851
0.08	0.3694	0.3758	0.3642	0.3716
0.09	0.3503	0.3503	0.3444	0.3446
0.1	0.3376	0.3439	0.3377	0.3378

Insertion Rate by Threshold



Omission	Threshold	Threshold	Threshold	Threshold
	1.4	1.5	1.6	1.7
0.01	0.775	0.7625	0.7625	0.75
0.02	0.5375	0.5188	0.5188	0.5125
0.03	0.4875	0.475	0.475	0.475
0.04	0.4563	0.45	0.45	0.4625
0.05	0.4437	0.4437	0.4437	0.4563
0.06	0.4187	0.4125	0.4187	0.4375
0.07	0.4	0.4	0.4063	0.4313
0.08	0.3812	0.3875	0.4	0.4187
0.09	0.3625	0.3625	0.3812	0.3938
0.1	0.35	0.3562	0.375	0.3875

Ommision Rate by Threshold



*Testing on different threshold and time tolerance, window size = 250

Issues & Problem (continue)

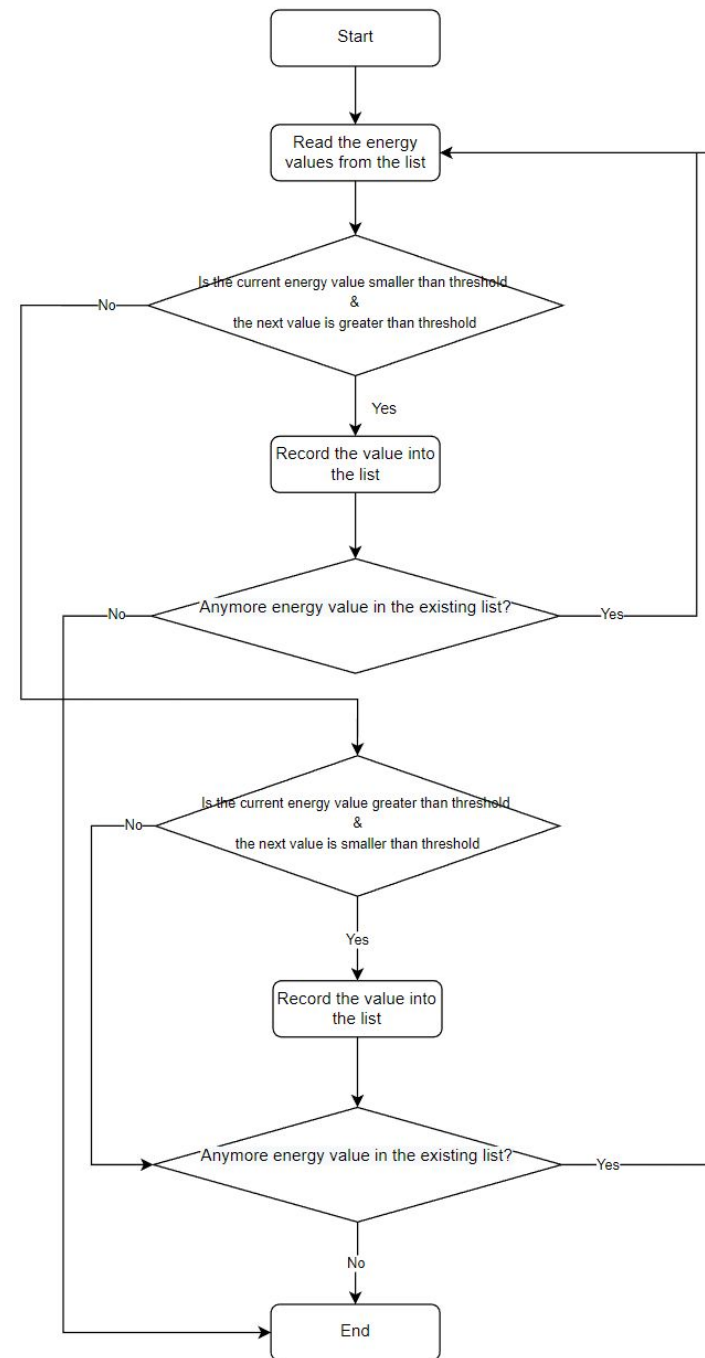
From the result, we concluded that the best result from these experiment variables:

- Time tolerance is 0.1
- Threshold is 1.4
- Window Size is 250

But...

The highest match rate is only 0.65 (65%), this is quite low from the experiment result for benchmark algorithm 1.

Proposed Algorithm



Proposed Algorithm

The proposed algorithm will be identifying the decremental value in the energy form. If the current energy value is not smaller than threshold, and the next energy value is also not greater than threshold (means there is no incrementally crossing the threshold) then the system will check if the current energy value is greater than the threshold and the next energy value is smaller than the threshold (means there is decrementally crossing the threshold).

```
if (x(m)< Thres) && (x(m+1)>Thres)
    S(ind) = m-1;
    ind=ind+1;
elseif (x(m)> Thres) && (x(m+1)<Thres)
    S(ind) = m-1;
    ind=ind+1;
end
```

Experimental Setup

- Experimental Variable : Time tolerance, Threshold and win size
- Time tolerance: $t = \{0.00, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10\}$
- Threshold : $\text{Thres} = \{1.4\}$
- Win size : $W = \{250\}$
- The same set-up is used for the benchmark algorithm.
- Comparison between the proposed and benchmark are measure based on
 - Performance indicator: High match rate, low insertion and omission rate

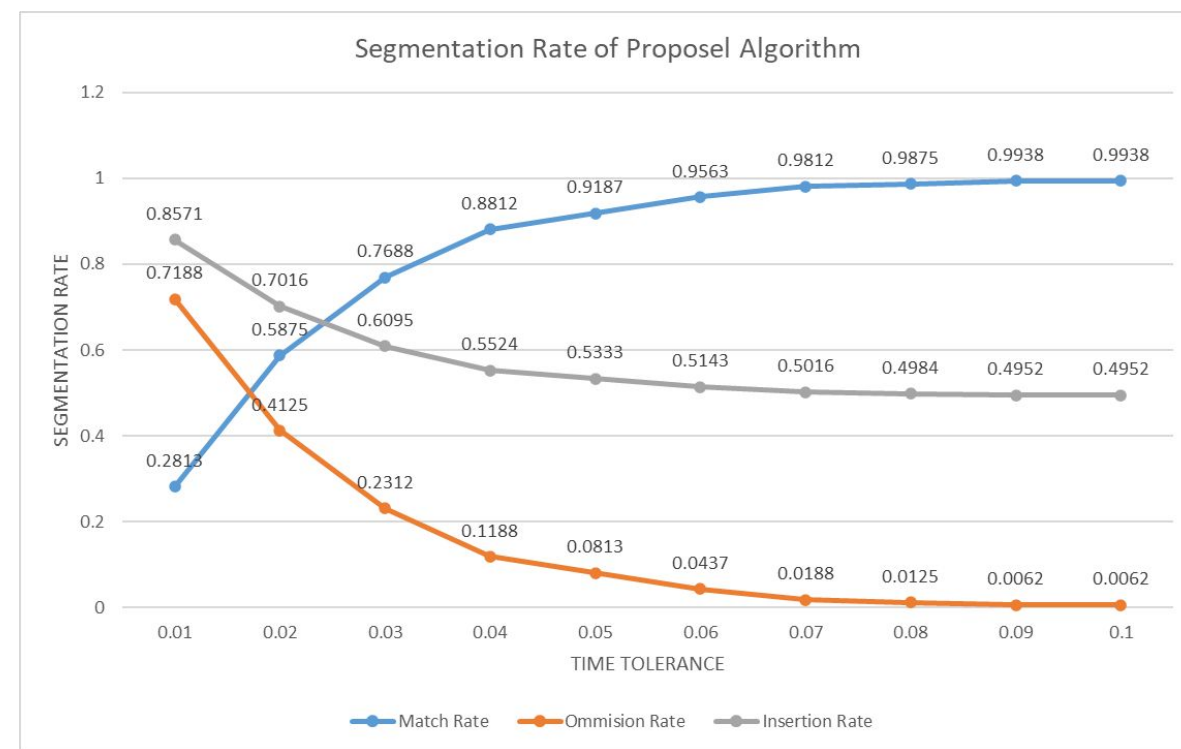
* The values here are for examples. You may used t heor choose your own that suit your solution

Results

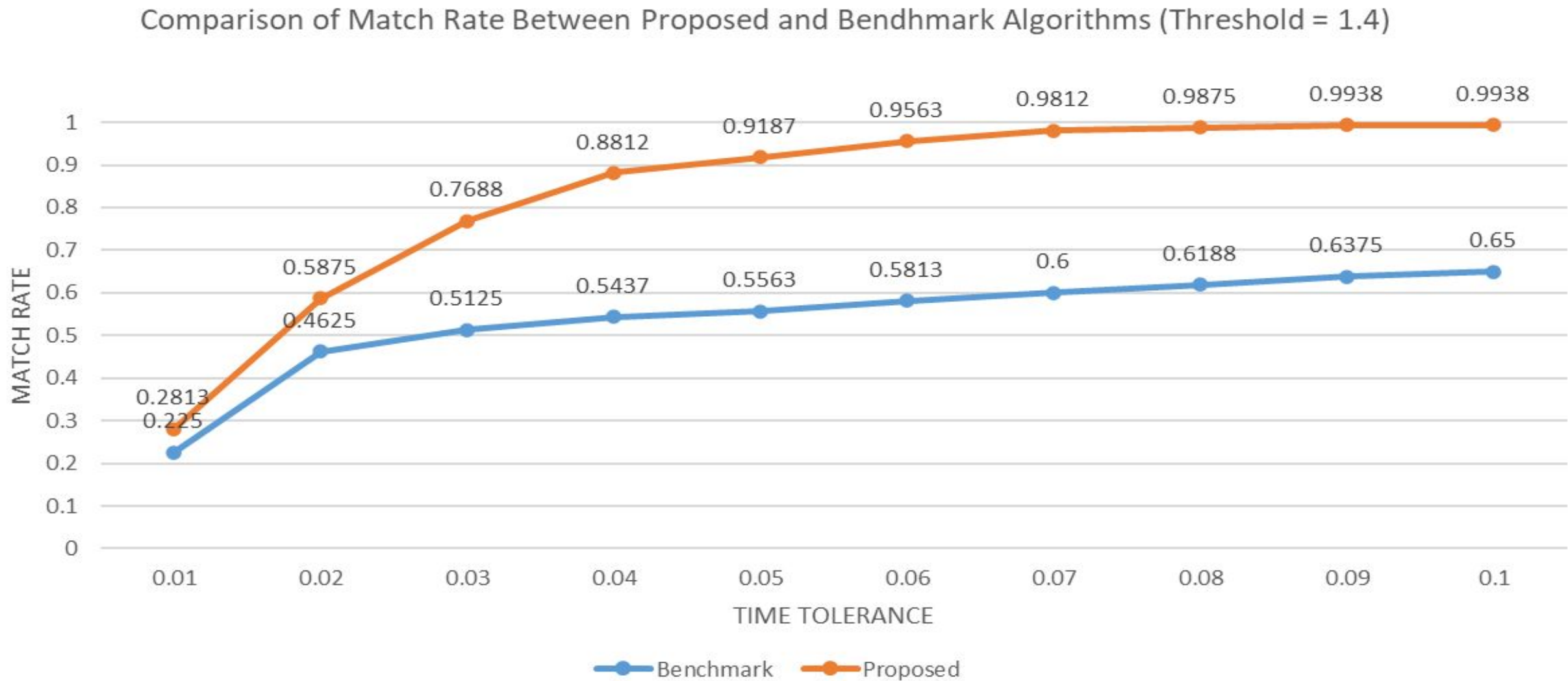
Testing with:

- Time tolerance = 0.1
- Threshold = 1.4
- Window Size = 250

	Match Rate	Ommision Rate	Insertion Rate
0.01	0.2813	0.7188	0.8571
0.02	0.5875	0.4125	0.7016
0.03	0.7688	0.2312	0.6095
0.04	0.8812	0.1188	0.5524
0.05	0.9187	0.0813	0.5333
0.06	0.9563	0.0437	0.5143
0.07	0.9812	0.0188	0.5016
0.08	0.9875	0.0125	0.4984
0.09	0.9938	0.0062	0.4952
0.1	0.9938	0.0062	0.4952

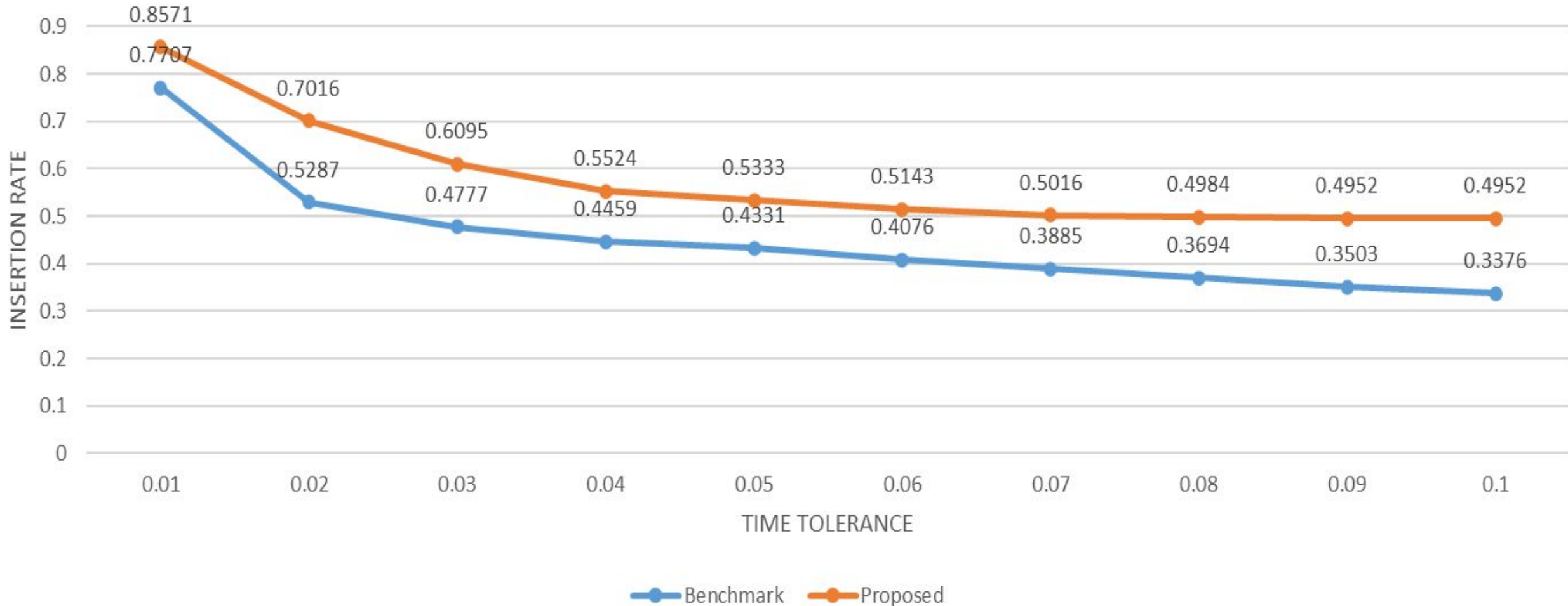


Results



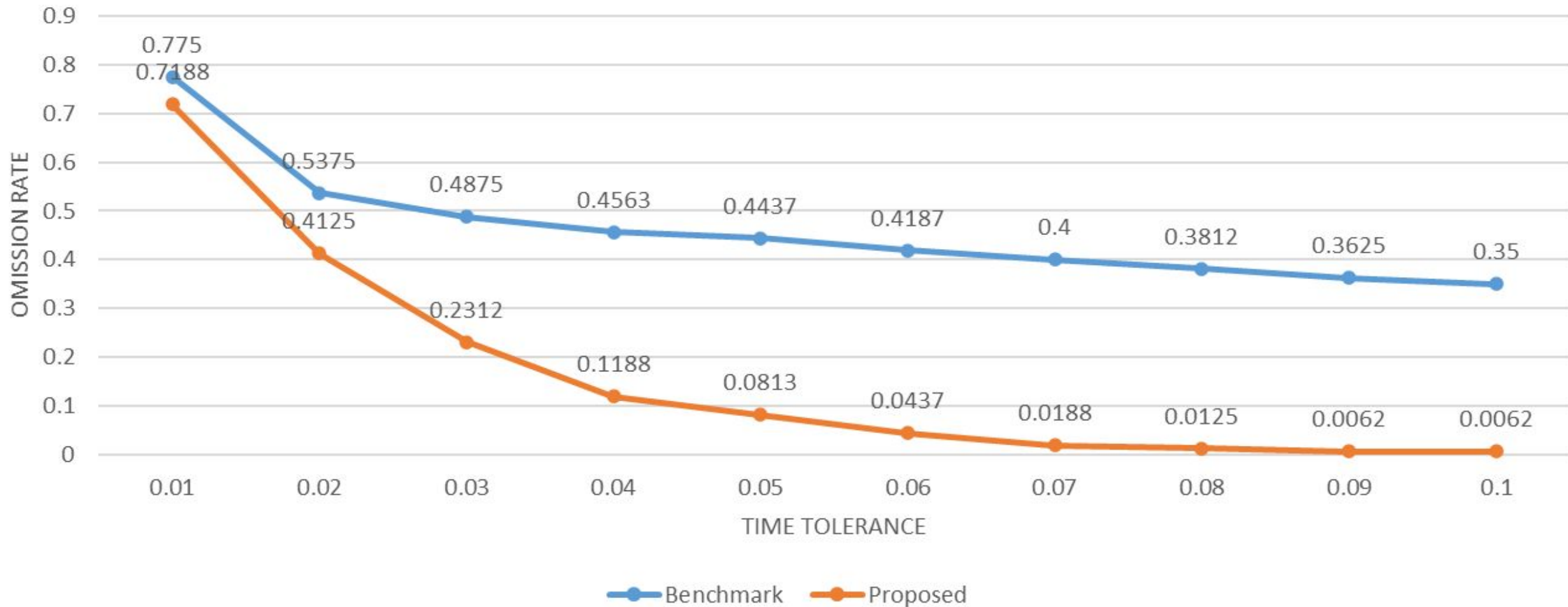
Results

Comparison of Insertion Rate Between Proposed and Benchmark Algorithms (Threshold = 1.4)

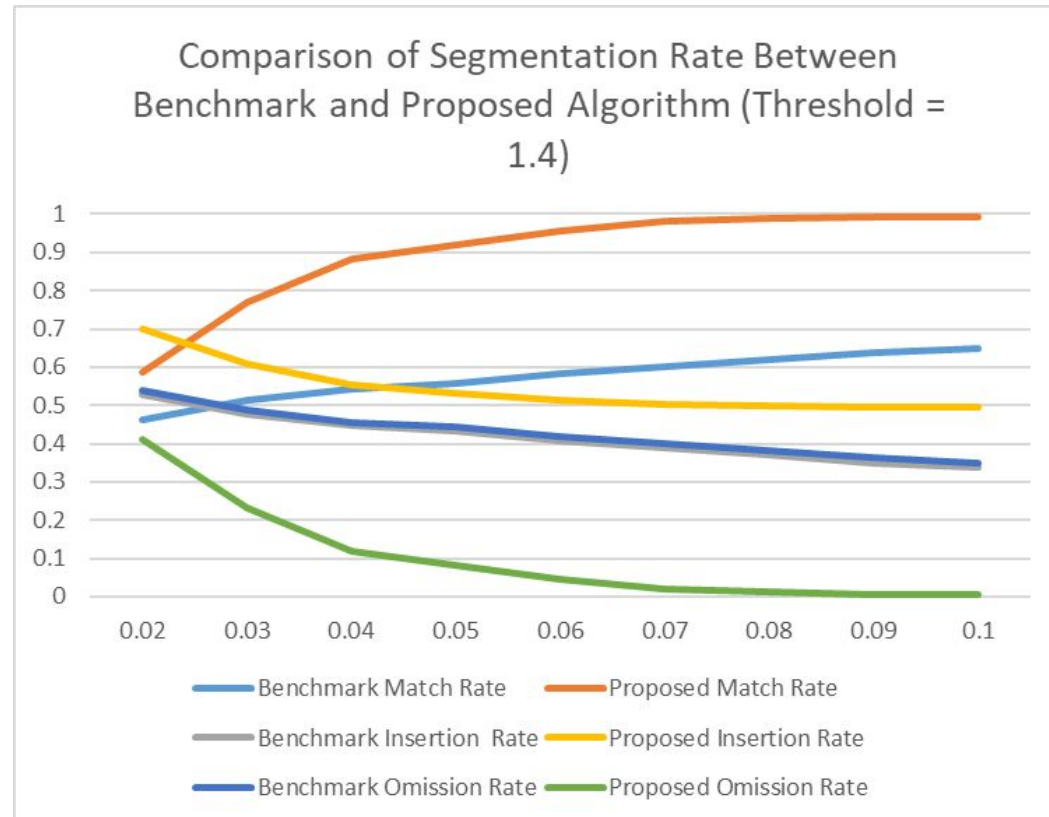


Results

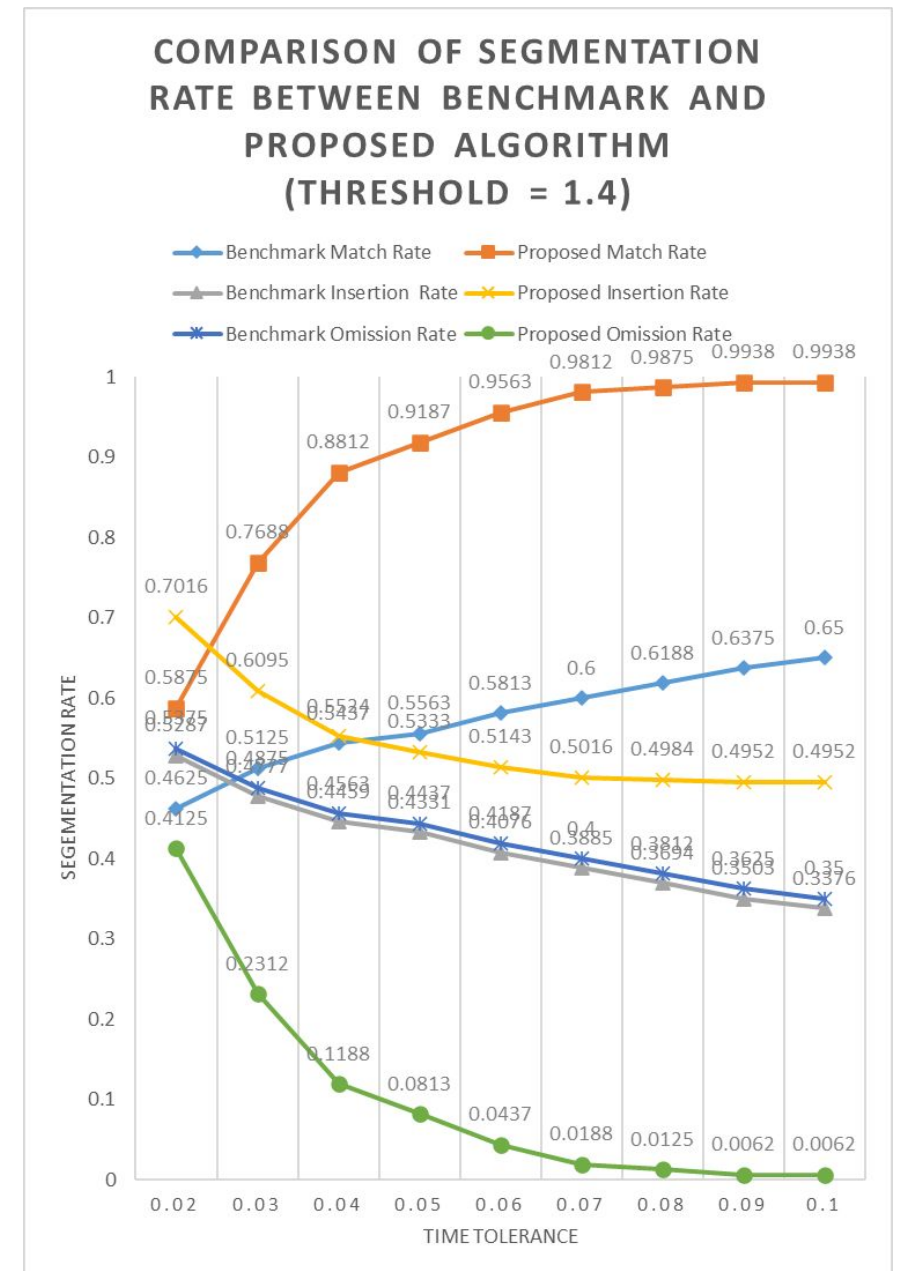
Comparison of Omission Rate Between Proposed and Benchmark Algorithms (Threshold = 1.4)



Results



General View



Detail View

Conclusion

- The best tolerance rate is 0.1, threshold is 1.4 and the window size is 250.
- The benchmark algorithm is only checking the incremental between the threshold.
- The proposed algorithm check both incremental and decremental between the threshold.
- Although the insertion rate for proposed algorithm is slightly higher than the benchmark algorithm, but we think it is alright because it is not much different.