Assignment 2

SPEECH SEGMENTATION TASK

TAN SEE JOU
A17CS0218

Introduction

This assignment requires the basic knowledge of speech processing especially speech segmentation and speech pattern understanding in designing an algorithm to segment the sound file accurately.

The given benchmark algorithm yield a less promising result due to the low match rate but high omission rate and insertion rate.

This assignment is to 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-A And Set-G
- This dataset contains 20 sound patterns of connected digit, from 10 male speakers and 10 Female speakers
- These files are

  (0075_a, 0075_g, 1206_a, 1206_g, 2433_a, 2433_g,

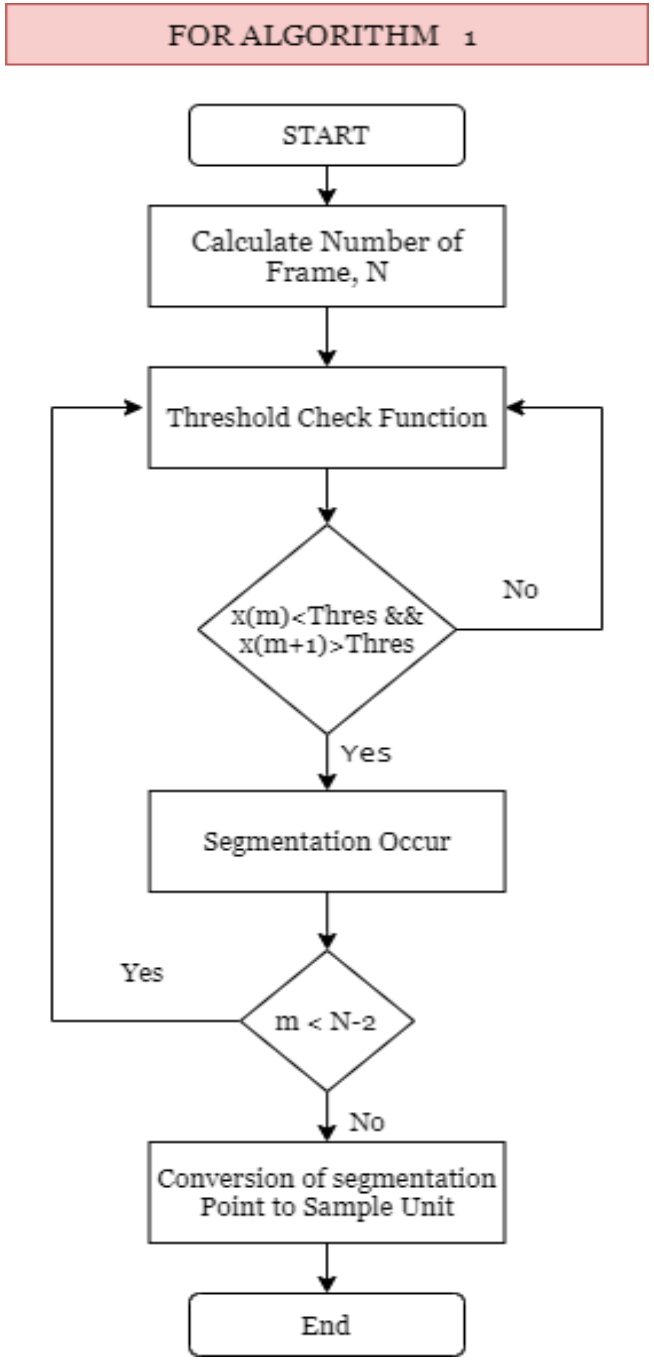
  3630_a, 3630_g, 4137_a, 4137_g, 5580_a, 5580_g,

  6255_a, 6255_g, 7565_a, 7565_g, 8299_a, 8299_g,

  9472_a, 9472_g)

- I found out that the patterns are come from 2 different people with different gender, Set A is come from a female speaker while Set G is come from a male speaker. Besides, the type of the speech is discrete as the speaker stop from a word to another. The vocabulary size is also considered small, which only cover to digits.

Benchmark Solution (Algorithm 1)



Issue & Problem of the Benchmark Algorithm

- For the benchmark algorithm, it only focus or make segmentation on the part of the signal that arise from the value smaller than threshold to higher than threshold.
- Therefore, from the figures, it is easily to notice that this algorithm miss a lot of segmentation on the end of the word (the part of signal that decrease from value higher than threshold to value smaller than threshold).
- Besides, the algorithm will insert the segmentation once it found out that there is signal that fulfil the requirement even though the last segmentation is just few frames ago.

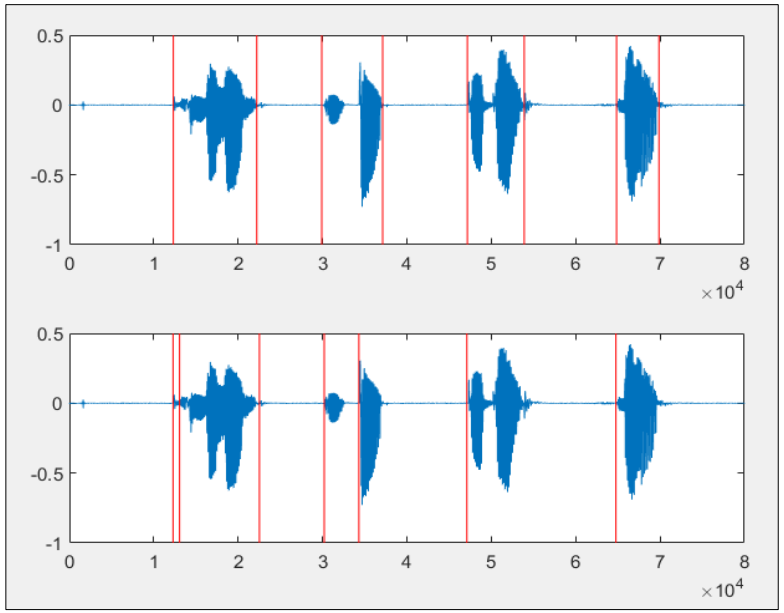


Figure 1: Speech Segmentation for data 2947_g

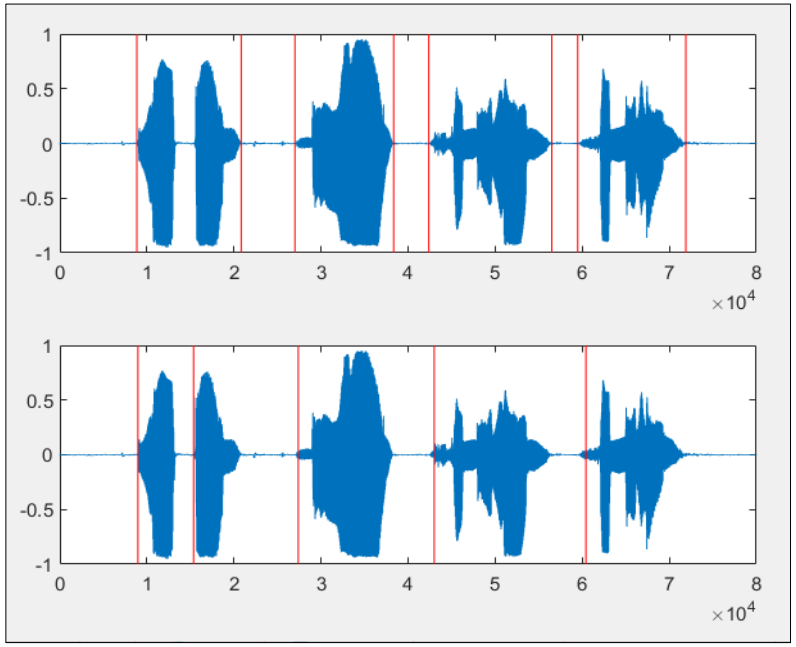


Figure 2: Speech Segmentation for data 8299_a

Issues & Problem (continue)

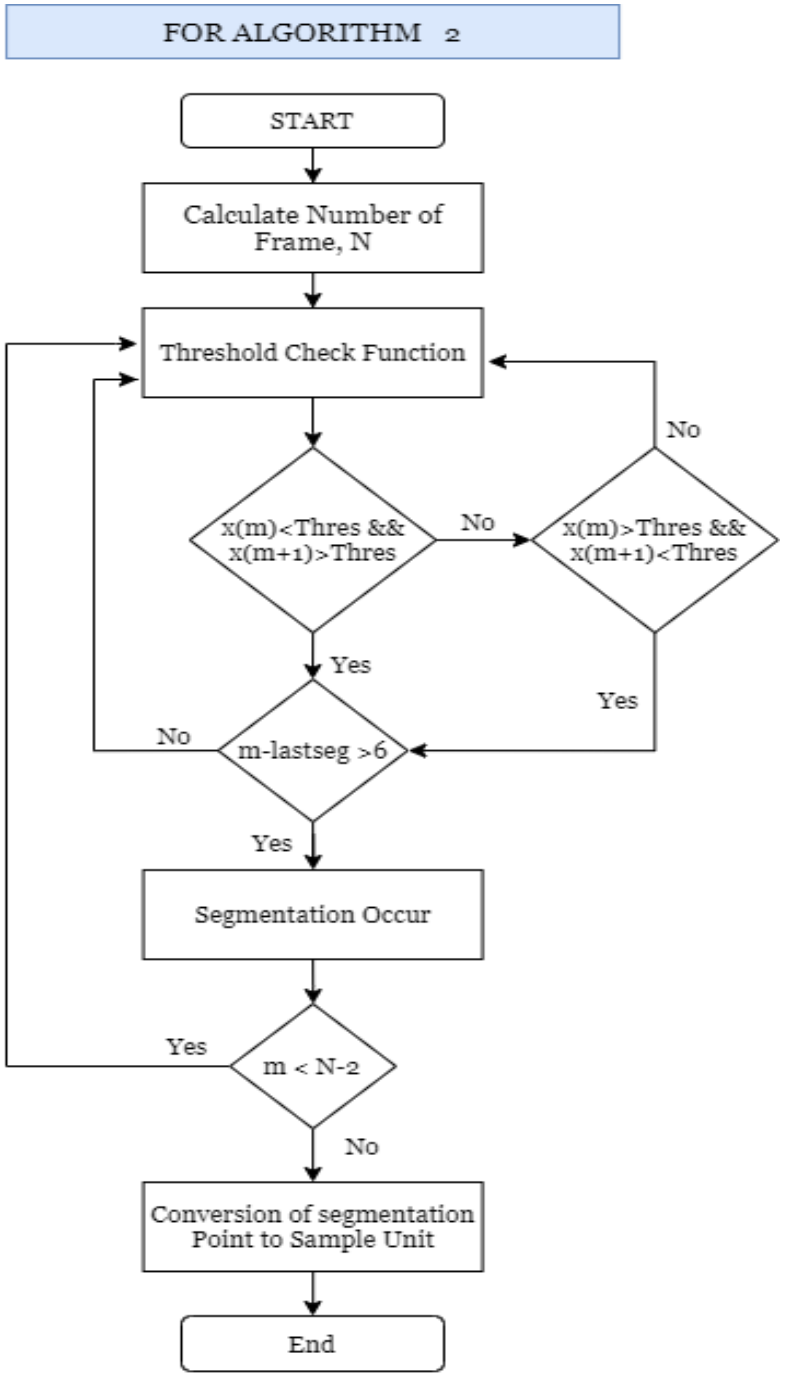
Table 1: Result of Benchmark Algorithm with Constant Threshold=2.0

Time Tolerance	Threshold	Match Rate	Omission Rate	Insertion Rate
0.01	2.0	0.36	0.64	0.53
0.02	2.0	0.46	0.54	0.39
0.03	2.0	0.49	0.51	0.36
0.04	2.0	0.49	0.51	0.35
0.05	2.0	0.51	0.49	0.32
0.06	2.0	0.52	0.48	0.31
0.07	2.0	0.52	0.48	0.31
0.08	2.0	0.53	0.47	0.30
0.09	2.0	0.54	0.46	0.29
0.10	2.0	0.56	0.44	0.26

Table 2: Result of Benchmark Algorithm with Constant Time Tolerance = 0.1

Time Tolerance	Threshold	Match Rate	Omission Rate	Insertion Rate
0.1	1.2	0.61	0.39	0.33
0.1	1.4	0.57	0.42	0.31
0.1	1.6	0.57	0.42	0.29
0.1	1.8	0.57	0.43	0.27
0.1	2.0	0.56	0.44	0.26
0.1	2.2	0.56	0.44	0.26
0.1	2.4	0.56	0.44	0.27
0.1	2.6	0.54	0.46	0.27
0.1	2.8	0.53	0.47	0.27
0.1	3.0	0.51	0.49	0.29

- Thus, the omission rate of benchmark algorithm is quite high as it miss or ignore a lot of segmentation especially the part at the end of the word. From the Table 1, when the time tolerance increase, the match rate is increase, while both the omission and insertion rate is decrease. This is because the higher the time tolerance, it is easier to reach the match requirement.
- According to the Table 2, the time tolerance is constant at 0.1, when the threshold value increase, the match rate and the insertion rate is decrease while the omission rate is increase. This is because when the threshold increase, it is more difficult to fulfil the segmentation condition, therefore, the omission rate is increase.



Proposed Algorithm (draw your algorithm process flow or Psedocode)

Proposed Algorithm

- To solve the issue from the benchmark algorithm, some conditions has been added to it. The proposed algorithm has added one more condition for the segmentation, which is when the signal value is decrease from the higher than threshold to value lower than threshold, then the segmentation will be occur. With this condition, the end of the word can be track correctly.
- Besides, one more condition is if the last segmentation is too close to the current, then the current signal value will not have segmentation occur.

Experimental Setup

- Experimental Variable : Time tolerance, Threshold
- Time tolerance:

  t = {0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10}

- Threshold :

  Thres = { 1.2,1.4, 1.6, 1.8, 2.0, 2.2 , 2.4, 2.6, 2.8, 3.0}

- The same set up are 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

Results

Table 3: Result of Proposed Algorithm with Constant Threshold=2.0

Time Tolerance	Threshold	Match Rate	Omission Rate	Insertion Rate
0.01	2.0	0.47	0.53	0.61
0.02	2.0	0.78	0.23	0.36
0.03	2.0	0.87	0.13	0.28
0.04	2.0	0.89	0.11	0.26
0.05	2.0	0.93	0.07	0.23
0.06	2.0	0.95	0.05	0.21
0.07	2.0	0.97	0.03	0.20
0.08	2.0	0.98	0.02	0.19
0.09	2.0	0.99	0.01	0.18
0.10	2.0	1.00	0.00	0.17

Table 4: Result of Proposed Algorithm with Constant Time Tolerance = 0.1

Time Tolerance	Threshold	Match Rate	Omission Rate	Insertion Rate
0.1	1.2	0.99	0.01	0.20
0.1	1.4	0.99	0.01	0.19
0.1	1.6	1.00	0.00	0.18
0.1	1.8	1.00	0.00	0.18
0.1	2.0	1.00	0.00	0.17
0.1	2.2	1.00	0.00	0.18
0.1	2.4	0.99	0.01	0.18
0.1	2.6	0.99	0.01	0.17
0.1	2.8	0.99	0.01	0.17
0.1	3.0	0.98	0.02	0.17

- From Table 3, the match rate increase when the time tolerance is increase. Besides, with the proposed algorithm, the omission rate is lower compare to the benchmark algorithm as a lot of segmentation point are being notice.
- From Table 4, the match rate and the omission rate is the best when the threshold is at optimized value.

Results

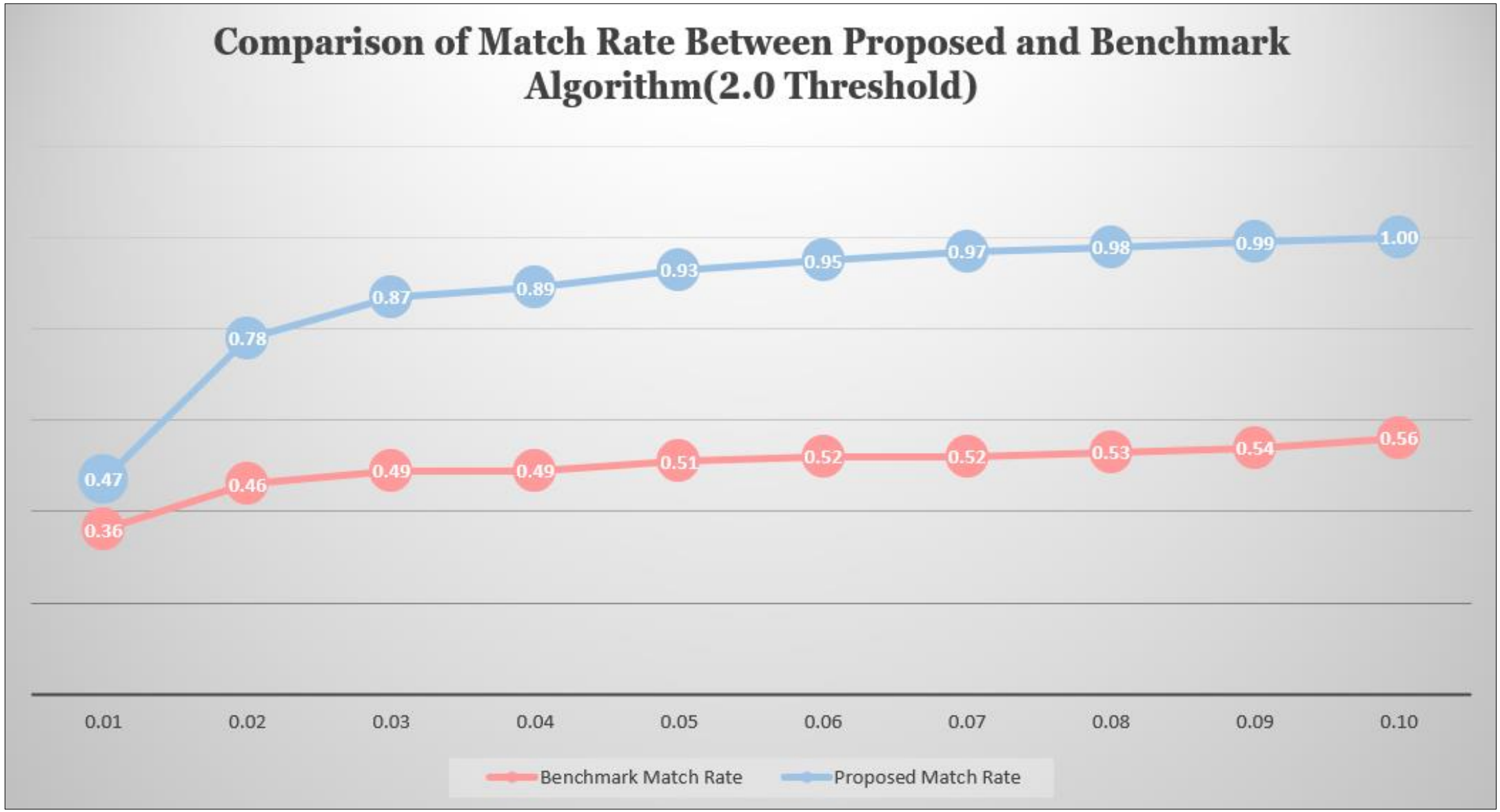
Table 5: Comparison for Both Algorithm with Constant Threshold=2.0

Time	Threshold	Benchmark			Proposed		
Tolerance	Value	Match Rate	Omission Rate	Insertion Rate	Match Rate	Omission Rate	Insertion Rate
0.01	2.0	0.36	0.64	0.53	0.47	0.53	0.61
0.02	2.0	0.46	0.54	0.39	0.78	0.23	0.36
0.03	2.0	0.49	0.51	0.36	0.87	0.13	0.28
0.04	2.0	0.49	0.51	0.35	0.89	0.11	0.26
0.05	2.0	0.51	0.49	0.32	0.93	0.07	0.23
0.06	2.0	0.52	0.48	0.31	0.95	0.05	0.21
0.07	2.0	0.52	0.48	0.31	0.97	0.03	0.20
0.08	2.0	0.53	0.47	0.30	0.98	0.02	0.19
0.09	2.0	0.54	0.46	0.29	0.99	0.01	0.18
0.10	2.0	0.56	0.44	0.26	1.00	0.00	0.17

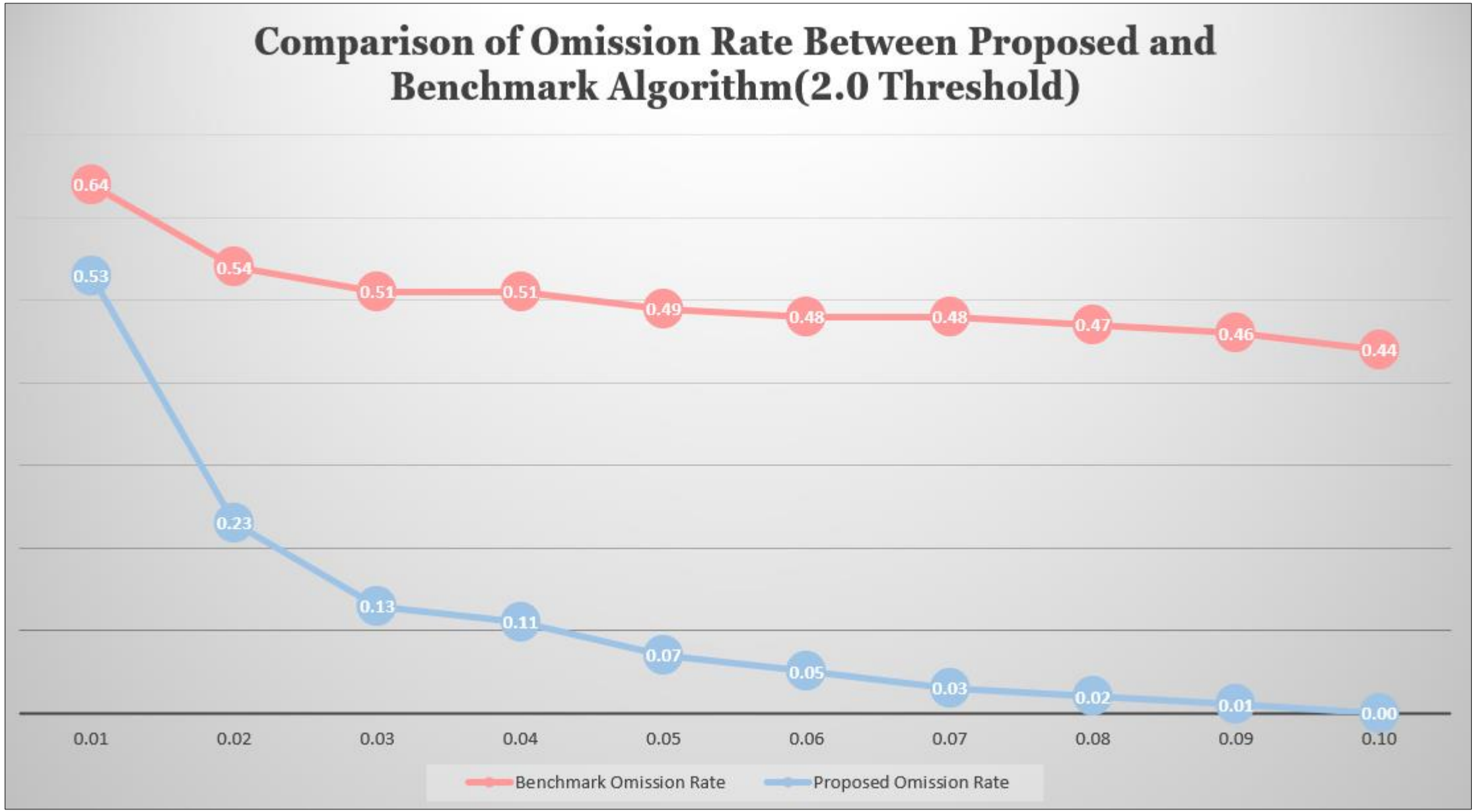
Table 6: Comparison for Both Algorithm with Constant Time Tolerance=0.1

Time	Threshold	Benchmark			Proposed		
Tolerance	Value	Match Rate	Omission Rate	Insertion Rate	Match Rate	Omission Rate	Insertion Rate
0.1	1.2	0.61	0.39	0.33	0.99	0.01	0.20
0.1	1.4	0.57	0.42	0.31	0.99	0.01	0.19
0.1	1.6	0.57	0.42	0.29	1.00	0.00	0.18
0.1	1.8	0.57	0.43	0.27	1.00	0.00	0.18
0.1	2.0	0.56	0.44	0.26	1.00	0.00	0.17
0.1	2.2	0.56	0.44	0.26	1.00	0.00	0.18
0.1	2.4	0.56	0.44	0.27	0.99	0.01	0.18
0.1	2.6	0.54	0.46	0.27	0.99	0.01	0.17
0.1	2.8	0.53	0.47	0.27	0.99	0.01	0.17
0.1	3.0	0.51	0.49	0.29	0.98	0.02	0.17

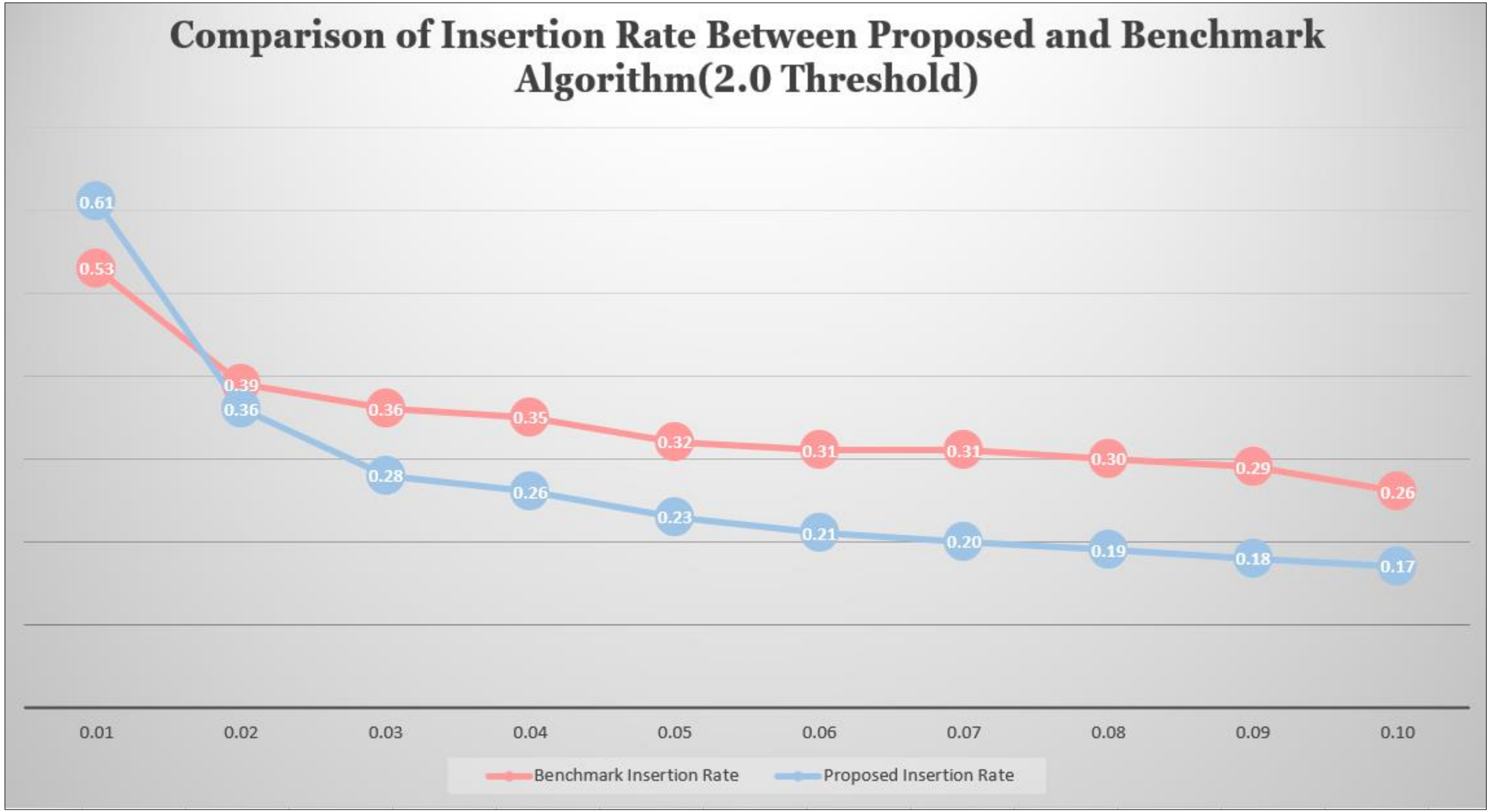
Results


Comparison of Match Rate Between Proposed and Benchmark Algorithm(2.0 Threshold)
0.47
0.78
0.87
0.89
0.93
0.95
0.97
0.98
0.99
1.00
0.36
0.46
0.49
0.49
0.51
0.52
0.52
0.53
0.54
0.56
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
Benchmark Match Rate
Proposed Match Rate

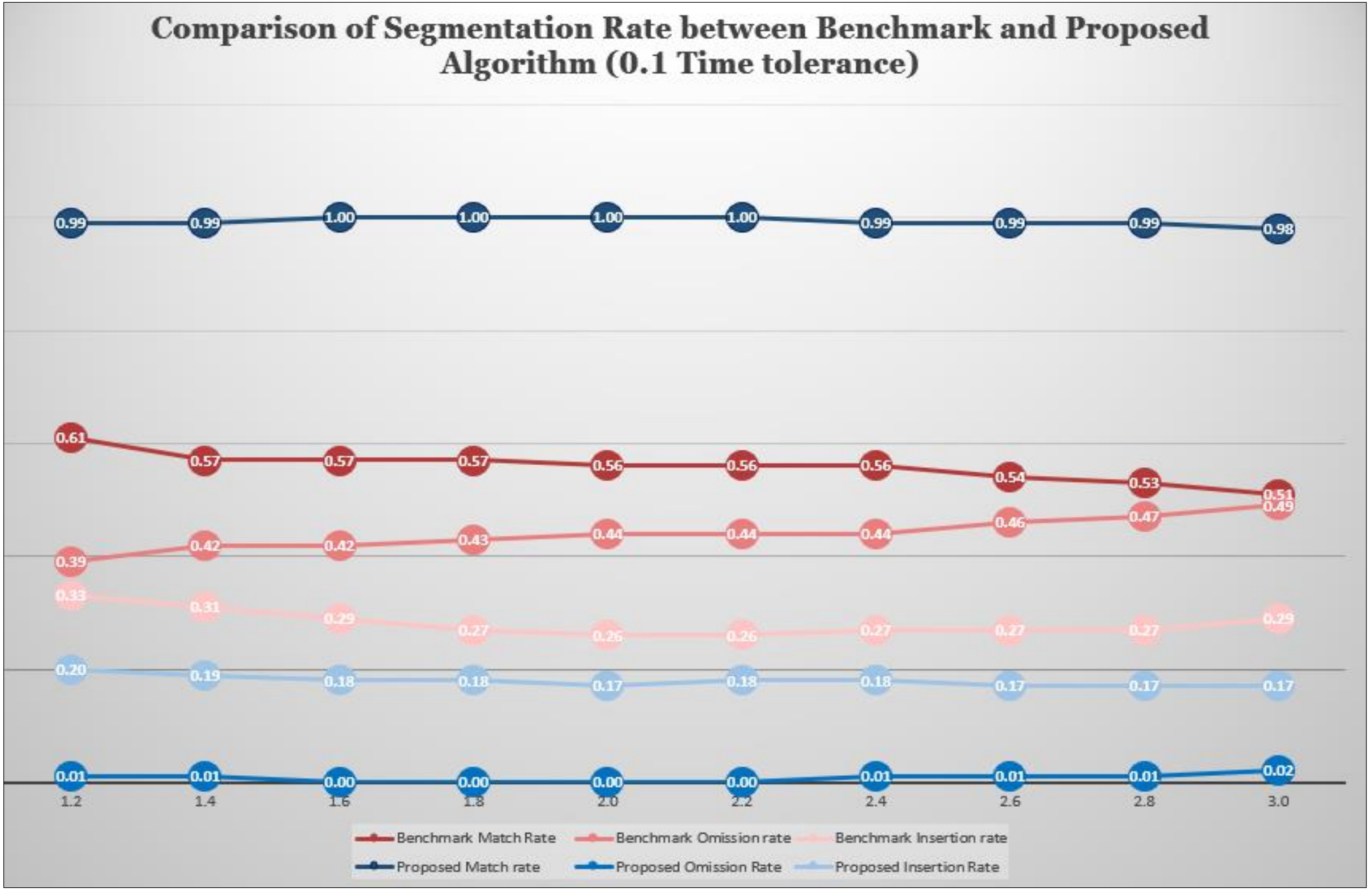
Results


Comparison of Omission Rate Between Proposed and Benchmark Algorithm(2.0 Threshold)
0.64
0.54
0.51
0.51
0.49
0.48
0.48
0.47
0.46
0.44
0.53
0.23
0.13
0.11
0.07
0.05
0.03
0.02
0.01
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
Benchmark Omission Rate
Proposed Omission Rate

Results


Comparison of Insertion Rate Between Proposed and Benchmark Algorithm(2.0 Threshold)
0.61
0.53
0.39
0.36
0.36
0.28
0.35
0.26
0.32
0.23
0.31
0.21
0.31
0.20
0.30
0.19
0.29
0.18
0.26
0.17
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
Benchmark Insertion Rate
Proposed Insertion Rate

Results


Comparison of Segmentation Rate between Benchmark and Proposed Algorithm (0.1 Time tolerance)
0.99 0.99 1.00 1.00 1.00 1.00 0.99 0.99 0.99 0.98
0.61 0.57 0.57 0.57 0.56 0.56 0.56 0.54 0.53 0.51
0.39 0.42 0.42 0.43 0.44 0.44 0.44 0.46 0.47 0.49
0.33 0.31 0.29 0.27 0.26 0.26 0.27 0.27 0.27 0.29
0.20 0.19 0.18 0.18 0.17 0.18 0.18 0.17 0.17 0.17
0.01 0.01 0.00 0.00 0.00 0.00 0.01 0.01 0.01 0.02
1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0
Benchmark Match Rate
Benchmark Omission rate
Benchmark Insertion rate
Proposed Match rate
Proposed Omission Rate
Proposed Insertion Rate

Conclusion

In short, the algorithm can only be considered as good if it is able to achieve a high match rate but low omission and insertion rate. From the above experiment, it show that the higher the time tolerance with constant threshold, the higher the match rate as there is higher possibility for the automatic point to match with the manual segmentation point.

Besides, the threshold value must be optimized depend on case as if the threshold value is too low, the insertion rate will be increase; if the threshold value is too high, the omission rate will be increase. This is because high threshold value will cause a lot of segmentation point to be ignore, while low threshold value will cause a lot of signal to be mistreat as the segmentation point.

Therefore, to achieve a high matching rate, both the time tolerance and threshold value have to be considered.