

การจัดกลุ่มลำดับย่อยของข้อมูลอนุกรมเวลาแบบกระแสอย่างมีความหมาย



นายวิชญ์ เนียรนาทระกุล

ศูนย์วิทยพัทพยาบาล  
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2553

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

MEANINGFUL SUBSEQUENCE CLUSTERING FOR TIME SERIES DATA STREAM



Mr. Vit Niennattrakul

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

A Dissertation Submitted in Partial Fulfillment of the Requirements  
for the Degree of Doctor of Philosophy Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

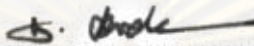
Academic Year 2010

Copyright of Chulalongkorn University

Thesis Title           MEANINGFUL SUBSEQUENCE CLUSTERING FOR TIME  
SERIES DATA STREAM  
By                        Mr. Vit Niennattrakul  
Field of Study         Computer Engineering  
Thesis Advisor        Assistant Professor Chotirat Ann Ratanamahatana, Ph.D.

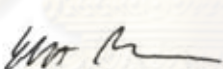
---

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of  
the Requirements for the Doctoral Degree

  
..... Dean of the Faculty of Engineering  
(Associate Professor Boonsom Lerthirunwong, Dr.Ing.)

THESIS COMMITTEE

  
..... Chairman  
(Professor Boonserm Kijisirikul, Ph.D.)

  
..... Thesis Advisor  
(Assistant Professor Chotirat Ann Ratanamahatana, Ph.D.)

  
..... Examiner  
(Professor Prabhas Chongstitvattana, Ph.D.)

  
..... Examiner  
(Assistant Professor Sukree Sinthupinyo, Ph.D.)

  
..... External Examiner  
(Assistant Professor Charnyote Pluempitiwiriyaewej, Ph.D.)

วิษญ์ เนียรนาทตระกูล: การจัดกลุ่มลำดับย่อยของข้อมูลอนุกรมเวลาแบบกระแสดังมี  
 ความหมาย. (MEANINGFUL SUBSEQUENCE CLUSTERING FOR TIME SERIES  
 DATA STREAM) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผู้ช่วยศาสตราจารย์ ดร. โชติรัตน์  
 รัตนามัทธนะ, 192 หน้า.

การจัดกลุ่มลำดับย่อยสำหรับข้อมูลอนุกรมเวลาแบบกระแสเป็นหนึ่งในปัญหาที่ท้าทายมาก  
 ที่สุดของการทำเหมืองข้อมูลอนุกรมเวลาดังแต่การจัดกลุ่มลำดับย่อยได้ถูกแสดงให้เห็นว่าการจัด  
 กลุ่มจะให้คำตอบที่ไร้ความหมายในเชิงการทดลองและทฤษฎี การจัดกลุ่มลำดับย่อยของข้อมูล  
 อนุกรมเวลาที่ถูกใช้ในหลายร้อยงานวิจัยนั้นจะให้คลื่นไซน์เป็นตัวแทนกลุ่มเสมอ ถ้าให้ข้อมูล  
 อนุกรมเวลาหนึ่ง ๆ การจัดกลุ่มลำดับย่อยของข้อมูลอนุกรมเวลาควรคืนค่าตัวแทนกลุ่มที่เป็น  
 ลักษณะของทุกลำดับย่อยในข้อมูลอนุกรมเวลา สาเหตุที่ทำให้เกิดความไร้ความหมายถูกระบุ  
 ไว้มาจากสองสาเหตุได้แก่ การใช้ระยะทางยุคลิดเป็นตัววัดระยะทางที่ไม่เหมาะสมและการใช้  
 การเฉลี่ยค่าตามแอมพลิจูดเป็นฟังก์ชันการเฉลี่ยที่ไม่เหมาะสม เพื่อที่จะได้มาซึ่งคำตอบของการ  
 จัดกลุ่มที่มีความหมาย ในวิทยานิพนธ์นี้การจัดกลุ่มลำดับย่อยของข้อมูลอนุกรมเวลาดังตามรูปได้  
 ถูกเสนอโดยใช้ระยะทางไดนามิกโทมัสวอร์ปปีงและการเฉลี่ยค่าตามรูปแทนระยะทางยุคลิดและ  
 การเฉลี่ยค่าตามแอมพลิจูดตามลำดับ ดังนั้นการจัดกลุ่มลำดับย่อยของข้อมูลอนุกรมเวลาดังตาม  
 รูปจะคืนผลลัพธ์ที่มีความหมายที่มากกว่าการจัดกลุ่มลำดับย่อยของข้อมูลอนุกรมเวลาแบบเดิม  
 แต่อย่างไรก็ตามการจัดกลุ่มลำดับย่อยของข้อมูลอนุกรมเวลาดังตามรูปไม่สามารถประยุกต์ใช้กับ  
 ข้อมูลแบบกระแสได้ เนื่องจากการจัดกลุ่มลำดับย่อยของข้อมูลอนุกรมเวลาดังตามรูปใช้เวลาในการ  
 ประมวลผลนานโดยคำนวณลำดับย่อยที่ผ่านมาทั้งหมดเมื่อมีจุดข้อมูลใหม่เข้ามา การจัดกลุ่ม  
 ลำดับย่อยของข้อมูลอนุกรมเวลาแบบกระแสดังตามรูปจึงถูกเสนอให้รองรับกรณีข้อมูลแบบกระแส  
 โดยคำนวณบนชุดข้อมูลขนาดเล็กของลำดับย่อยที่เก็บไว้แทนที่จะคำนวณจากลำดับย่อยทั้งหมด  
 ซึ่งชุดข้อมูลของลำดับย่อยที่เก็บไว้ถูกปรับปรุงสำหรับทุกๆจุดข้อมูลเพื่อรักษาจำนวนลำดับย่อย  
 ในชุดข้อมูลไม่ให้เกินกว่าจำนวนมากที่สุดที่อนุญาต ดังนั้นการจัดกลุ่มลำดับย่อยของข้อมูลอนุกรม  
 เวลาแบบกระแสดังตามรูปจึงเร็วกว่าการจัดกลุ่มลำดับย่อยของข้อมูลอนุกรมเวลาดังตามรูปอย่างมาก

ภาควิชา ..... วิศวกรรมคอมพิวเตอร์.....

สาขาวิชา ..... วิศวกรรมคอมพิวเตอร์.....

ปีการศึกษา ..... 2553 .....

ลายมือชื่อนิสิต .....

ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

## 4971828021: MAJOR COMPUTER ENGINEERING

KEYWORDS: DATA MINING / SUBSEQUENCE CLUSTERING / TIME SERIES / DATA STREAM

VIT NIENNATTRAKUL : MEANINGFUL SUBSEQUENCE CLUSTERING FOR TIME SERIES DATA STREAM. ADVISOR : ASSISTANT PROFESSOR CHOTIRAT RATANAMAHATANA, PH.D., 192 pp.

Subsequence clustering for time series data streams is one of the most challenging issues of time series data mining since subsequence clustering has been proven both theoretically and empirically that it produces meaningless clustering results, where hundreds of research works that utilize Subsequence Time Series Clustering (STSC) as a preprocessing step and a subroutine are all affected. Given a time series sequence, subsequence clustering should return cluster representatives which represent characteristics of all subsequences in time series. Therefore, if cluster representatives are always sine waves regardless of inputs, clustering results are meaningless since they do not reflect characteristics of the subsequences. The causes of meaninglessness are identified in twofold, i.e., inappropriate uses of Euclidean distance as a distance measure and Amplitude Averaging as an averaging function. To achieve meaningful clustering results, in this thesis, Shape-based Subsequence Time Series Clustering (2STSC) is proposed to use Dynamic Time Warping (DTW) distance measure and Shape-based Averaging function. Therefore, 2STSC returns more meaningful results than those from STSC. However, 2STSC cannot directly apply to data streams since 2STSC consumes large computational complexity by considering all previous subsequences for every new incoming data point. Shape-based Streaming Subsequence Time Series Clustering (3STSC) is then proposed to handle the streaming case by calculating a clustering result on a small set of stored subsequences instead of calculating from all previous subsequences. The small set of stored subsequences is updated for every new incoming data point to maintain the number of stored subsequences not to exceed the maximum allowance. 3STSC, therefore, is much faster than 2STSC, while 3STSC returns small distortions of clustering results.

Department: ..... Computer Engineering .....

Student's Signature ..  .....

Field of Study: .... Computer Engineering .....

Advisor's Signature ..  .....

Academic Year: .....2010.....

## Acknowledgments

I would like to express my sincere gratitude to my thesis advisor, Dr. Chotirat Ann Ratanamahatana for her invaluable guidance and support during my graduate studies at Chulalongkorn University. She always encourages my progress, new ideas, as well as motivates my research work. She is always full of energy and untiringly available for giving me insightful advices. In addition, I have learned precious lessons through her past and present remarkable research work, as well as her exceptional presentation. I truly consider it a great privilege in having the opportunity to work with her as my graduate advisor.

I am grateful to Prof. Eamonn Keogh who supported me when I was in the United States. I also express my thankfulness to my dissertation committee: Dr. Prabhas Chongstitvatana, Dr. Boonserm Kitsirikul, Dr. Sukree Sinthupinyo, and Dr. Charnyote Pluempitwiriwaj. I am indebted to every teacher, especially, Dr. Proadpran, Dr. Atiwong, Dr. Athasit, Dr. Pizzanu, Dr. Vishnu, Dr. Somchai, Ajarn Mandhana for introducing me a rabbit hole of computer engineering for nine years since my undergraduate years. I would like to thank my lovely friends, Soung, Nart, Aim, Ji, Ping, Heng, Ton, Nui, Yong, Jen, Rote, Pick, Pong, Kwang, Guk, N’Pam, N’Bird, N’Bim, N’Pao, N’Au, N’Rong, N’Pop, N’Pun, Kook, Tohn, Poo, P’Lin, P’Komate, P’Woon, P’Petch, P’Jung, P’Yui, P’Nan, P’Noot, P’Ae, P’O, P’Woot, and P’Ko for their wonderful friendships, encouragement, and many valuable discussions. I also would like to thank Took, Moo, Nakorn, P’Tookta, P’Lek, P’Fad, P’Pang, P’Art, P’Dao, P’Hao, P’May, P’Pui, P’Tee-Guay, P’Ae, P’O, N’Parn, N’Neoy, N’Ping, N’Nanah, and N’Giffy for making my stay in the U.S. wonderful; without them, my visit would be colorless. Additionally, I am thankful to the administrative staffs for always being helpful during my whole time attending the Department of Computer Engineering.

I greatly appreciate the financial support from the Thailand Research Fund given through the Royal Golden Jubilee Ph.D. Program (PHD/0141/2549) and the Chulalongkorn University Graduate Scholarship to Commemorate the 72<sup>nd</sup> Anniversary of His Majesty King Bhumibol Adulyadej for giving me the invaluable opportunity and providing financial support during my precious a half decade of years studying in Ph.D. program and going abroad to the United States. I also greatly appreciate the research fund from the 90<sup>th</sup> Anniversary of Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund) to make my research significantly forward and my idea becomes reality.

Finally, with my utmost gratitude, this dissertation is dedicated to my beloved parents for shaping my life for what it is today, and to my sisters for always being there. Without their love, encouragement, understanding, and support, this research could not have been completed.

# Contents

	Page
<b>Abstract (Thai)</b> . . . . .	iv
<b>Abstract (English)</b> . . . . .	v
<b>Acknowledgments</b> . . . . .	vi
<b>Contents</b> . . . . .	vii
<b>List of Tables</b> . . . . .	x
<b>List of Figures</b> . . . . .	xi
<b>Chapter</b>	
<b>I Introduction</b> . . . . .	<b>1</b>
1.1 Objective of the Thesis . . . . .	5
1.2 Scopes of the Thesis . . . . .	5
1.3 Contributions of the Thesis . . . . .	6
1.4 Research Methodology . . . . .	6
<b>II Meaninglessness of Subsequence Time Series Clustering</b> . . . . .	<b>8</b>
2.1 Background . . . . .	8
2.1.1 Subsequence Time Series Clustering (STSC) . . . . .	8
2.1.2 $K$ -Hierarchical Clustering . . . . .	10
2.1.3 $K$ -Means Clustering . . . . .	12
2.1.4 Euclidean Distance . . . . .	13
2.1.5 Amplitude Averaging . . . . .	14
2.1.6 $Z$ -Normalization . . . . .	15
2.2 Related Work . . . . .	16
2.3 Experiments . . . . .	19
2.3.1 First Experiment . . . . .	20
2.3.2 Second Experiment . . . . .	22
2.4 Causes of Meaninglessness . . . . .	24
2.5 Conclusion . . . . .	27
<b>III Shape-based Averaging</b> . . . . .	<b>28</b>
3.1 Background . . . . .	29
3.1.1 Dynamic Time Warping (DTW) Distance . . . . .	29
3.1.2 Dynamic Time Warping (DTW) Averaging . . . . .	30
3.2 Related Work . . . . .	31

Chapter	Page
3.3 Shape-based Averaging . . . . .	33
3.3.1 Cubic-Spline Dynamic Time Warping (CDTW) Averaging . . . . .	33
3.3.2 Iterative Cubic-Spline Dynamic Time Warping (ICDTW) Averaging . . . . .	35
3.4 Experimental Evaluation . . . . .	37
3.5 Averaging Trivial-Matched Subsequences . . . . .	39
3.6 Conclusion . . . . .	39
<b>IV 2STSC: Shape-based Subsequence Time Series Clustering . . . . .</b>	<b>41</b>
4.1 Related Work . . . . .	42
4.2 Shape-based Subsequence Time Series Clustering (2STSC) . . . . .	47
4.3 Experimental Evaluation . . . . .	48
4.4 Conclusion . . . . .	53
<b>V Incremental Shape-based Averaging . . . . .</b>	<b>54</b>
5.1 Incremental Shape-based Averaging . . . . .	54
5.2 Experimental Evaluation . . . . .	56
5.2.1 First Experiment . . . . .	56
5.2.2 Second Experiment . . . . .	57
5.3 Conclusion . . . . .	60
<b>VI 3STSC: Shape-based Streaming Subsequence Time Series Clustering . . . . .</b>	<b>61</b>
6.1 Related Work . . . . .	62
6.2 Shape-based Streaming Subsequence Time Series Clustering . . . . .	63
6.3 Experimental Evaluation . . . . .	65
6.3.1 First Experiment . . . . .	65
6.3.2 Second Experiment . . . . .	67
6.4 Conclusion . . . . .	68
<b>VII Conclusion . . . . .</b>	<b>70</b>
<b>VIII Publications . . . . .</b>	<b>72</b>
<b>References . . . . .</b>	<b>83</b>
<b>Appendix</b>	



Chapter	Page
<b>Appendix A Datasets</b> . . . . .	<b>85</b>
<b>Appendix B Complete Experimental Results of the First Experiment in Chapter 2</b> . . . . .	<b>89</b>
<b>Appendix C Complete Experimental Results of the Experiment in Chapter 3</b> . . .	<b>105</b>
<b>Appendix D Complete Experimental Results of the Experiment in Chapter 4</b> . . .	<b>109</b>
<b>Appendix E Complete Experimental Results of the First Experiment in Chapter 5</b> . . . . .	<b>128</b>
<b>Appendix F Complete Experimental Results of the Second Experiment on Chapter 5</b> . . . . .	<b>133</b>
<b>Appendix G Complete Experimental Results of the First Experiment in Chapter 6</b> . . . . .	<b>150</b>
<b>Appendix H Complete Experimental Results of the Second Experiment in Chapter 6</b> . . . . .	<b>171</b>
<b>Biography</b> . . . . .	<b>192</b>

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

## List of Tables

Table	Page
2.1 Pseudo code of Subsequence Time Series Clustering (STSC) . . . . .	9
2.2 Agglomerative hierarchical clustering algorithm (AGNES) . . . . .	10
2.3 Pseudo code of single linkage distance function . . . . .	11
2.4 Pseudo code of complete linkage distance function . . . . .	12
2.5 Pseudo code of average linkage distance function . . . . .	12
2.6 Pseudo code of $k$ -means clustering . . . . .	13
2.7 Pseudo code of Amplitude Averaging function . . . . .	15
3.1 Pseudo code of Dynamic Time Warping distance measure . . . . .	30
3.2 Pseudo code of Dynamic Time Warping averaging function . . . . .	31
3.3 Pseudo code of generating a warping path . . . . .	32
3.4 Pseudo code of Shape-based Averaging scheme . . . . .	33
3.5 Pseudo code of Cubic-Spline Dynamic Time Warping (CDTW) averaging function . . .	35
3.6 Pseudo code of Iterative Cubic-Spline Dynamic Time Warping (ICDTW) averaging function . . . . .	36
3.7 SUMDIST of each averaging method . . . . .	38
4.1 Pseudo code of Shape-based Subsequence Time Series Clustering (2STSC) . . . . .	48
5.1 Pseudo code of Incremental Shape-based Averaging . . . . .	55
5.2 Updating stored sequences in Incremental Shape-based Averaging . . . . .	55
5.3 Averaging stored sequences in Incremental Shape-based Averaging . . . . .	56
6.1 Pseudo code of Shape-based Streaming Subsequence Time Series Clustering (3STSC) .	64
6.2 Updating stored sequences in 3STSC . . . . .	65
A.1 Details of the UCR classification/clustering datasets used in Chapters 3 and 5 . . . . .	88

## List of Figures

Figure	Page
1.1 Examples of time series data in real world. . . . .	2
1.2 Multivariate time seires collected from SmartCane system. (Wu et al., 2008). . . . .	3
1.3 Cluster representatives generated from STSC . . . . .	3
1.4 Trivial-matched subsequences of CBF sequence . . . . .	4
2.1 Overview of Subsequence Time Series Clustering (STSC) . . . . .	9
2.2 Example of Euclidean distance calculation. . . . .	14
2.3 Example of Amplitude Averaging calculation. . . . .	14
2.4 Example of $z$ -normalization. . . . .	16
2.5 Examples of Cylinder-Bell-Funnel dataset . . . . .	17
2.6 Some part of Cylinder-Bell-Funnel sequence . . . . .	18
2.7 Cluster representatives generated from STSC . . . . .	18
2.8 Datasets from TSDMA used in the experiments. . . . .	20
2.9 Cluster representatives generated from STSC of Buoy1 when $k = 3$ and $w = 64$ . . . . .	21
2.10 Cluster representatives generated from STSC of CBF when $k = 3$ and $w = 64$ . . . . .	21
2.11 Constructed sine waves generated from STSC of Buoy1 when $k = 3$ and $w = 64$ . . . . .	22
2.12 Constructed sine waves generated from STSC of CBF when $k = 3$ and $w = 64$ . . . . .	22
2.13 KLMMs of STSC using $k$ -means clustering. . . . .	23
2.14 KLMMs of STSC using $k$ -hierarchical clustering. . . . .	24
2.15 Trivial-matched subsequences of CBF sequence . . . . .	25
2.16 Euclidean distance cannot capture similarity between trivial-matched subsequences . . . . .	26
2.17 Amplitude Averaging produces an smoothened averaged result. . . . .	27
3.1 Comparision between two averaged results generated from Amplitude Averaging and Shape-based Averaging. . . . .	28
3.2 Alignment obtained from a DTW distance calculation. . . . .	30
3.3 Result generated from DTW Averaging . . . . .	31
3.4 Comparison between DTW averaging and CDTW averaging functions . . . . .	34
3.5 Averaged results before and after re-sampling in CDTW averaging function. . . . .	35
3.6 Examples of some classes in evaluated datasets. . . . .	37
3.7 Averaged results of CBF . . . . .	38
3.8 Averaged results of ECG . . . . .	38
3.9 Trivial-matched subsequences b) extracted from a) CBF sequence. . . . .	39
3.10 Averaged results generated from Amplitude Averaging. . . . .	40
3.11 Averaged results generated from Shape-based Averaging with CDTW function. . . . .	40

Figure	Page
3.12 Averaged results generated from Shape-based Averaging with ICDTW function. . . . .	40
4.1 Three sets of trivial-matched subsequences. . . . .	42
4.2 a) Euclidean cannot capture the similarity of trivial-matched subsequences, while b) DTW can. . . . .	43
4.3 a) Amplitude Averaging cannot construct meaningful representatives of trivial- matched subsequences, while b) Shape-based Averaging can. . . . .	44
4.4 a) STSC produces a meaningless clustering result, while b) 2STSC produces a meaningful clustering result. . . . .	44
4.5 Overview of 2STSC using DTW distance and Shape-based Averaging. . . . .	48
4.6 Datasets used to evaluate meaningfulness of STSC and 2STSC . . . . .	49
4.7 SMMs of Buoy1 when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied. . . . .	51
4.8 SMMs of CBF when the number of clusters ( $k$ ) is 3 and the length of sliding win- dow ( $w$ ) is varied. . . . .	51
4.9 SMMs of Buoy1 when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied. . . . .	52
4.10 SMMs of CBF when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied. . . . .	52
4.11 Cluster representatives generated from 2STSC of Buoy1 with complete linkage (left) and average linkage (right) when $k = 3$ and $w = 64$ . . . . .	53
4.12 Cluster representatives generated from 2STSC of CBF with complete linkage (left) and average linkage (right) when $k = 3$ and $w = 64$ . . . . .	53
5.1 Examples of some classes in evaluated datasets. . . . .	56
5.2 Computational time of Incremental Shape-based Averaging and Shape-based Av- eraging when a new incoming sequence arrives. . . . .	57
5.3 Difference of SUMDIST and speedup of Buoy1 when the number of stored se- quences to an original dataset is varied. . . . .	58
5.4 Difference of SUMDIST and speedup of CBF when the number of stored sequences to an original dataset is varied. . . . .	59
5.5 Averaged results of some classes of CBF from Incremental Shape-based Averaging. . .	59
5.6 Averaged results of some classes of ECG from Incremental Shape-based Averaging. . .	60
6.1 Overview of Shape-based Streaming Subsequence Time Series Clustering (3STSC). . .	63
6.2 Some datasets from TSDMA used in the experiment. . . . .	66
6.3 Computational time of 3STSC and 2STSC of Buoy1 when a new incoming se- quence arrives. . . . .	66

Figure	Page
6.4 Computational time of 3STSC and 2STSC of CBF when a new incoming sequence arrives. . . . .	66
6.5 Percentage difference of SMM and speedup of 3STSC of Buoy1 when $k = 3$ , $w = 64$ , and number of stored sequences are varied. . . . .	67
6.6 Percentage difference of SMM and speedup of 3STSC of CBF when $k = 3$ , $w = 64$ , and number of stored sequences are varied. . . . .	68
A.1 Datasets from TSDMA used in the experiments of Chapters 2, 4, and 6. . . . .	86
A.2 Examples of some classes of the UCR classification/clustering datasets used in Chapters 3 and 5. . . . .	87
B.1 Cluster representatives generated from STSC using $k$ -means clustering when $k = 3$ and $w = 32$ . . . . .	90
B.2 Cluster representatives generated from STSC using $k$ -means clustering when $k = 3$ and $w = 64$ . . . . .	90
B.3 Cluster representatives generated from STSC using $k$ -means clustering when $k = 3$ and $w = 128$ . . . . .	91
B.4 Cluster representatives generated from STSC using $k$ -means clustering when $k = 5$ and $w = 64$ . . . . .	91
B.5 Cluster representatives generated from STSC using $k$ -means clustering when $k = 7$ and $w = 64$ . . . . .	92
B.6 Constructed sine waves generated from STSC using $k$ -means clustering when $k = 3$ and $w = 32$ . . . . .	92
B.7 Constructed sine waves generated from STSC using $k$ -means clustering when $k = 3$ and $w = 64$ . . . . .	93
B.8 Constructed sine waves generated from STSC using $k$ -means clustering when $k = 3$ and $w = 128$ . . . . .	93
B.9 Constructed sine waves generated from STSC using $k$ -means clustering when $k = 5$ and $w = 64$ . . . . .	94
B.10 Constructed sine waves generated from STSC using $k$ -means clustering when $k = 7$ and $w = 64$ . . . . .	94
B.11 Cluster representatives generated from STSC using $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when $k = 3$ and $w = 32$ . . . . .	95
B.12 Cluster representatives generated from STSC using $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when $k = 3$ and $w = 64$ . . . . .	96

Figure	Page
B.13 Cluster representatives generated from STSC using $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when $k = 3$ and $w = 128$ . . . . .	97
B.14 Cluster representatives generated from STSC using $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when $k = 5$ and $w = 64$ . . . . .	98
B.15 Cluster representatives generated from STSC using $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when $k = 7$ and $w = 64$ . . . . .	99
B.16 Constructed sine waves generated from STSC using $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when $k = 3$ and $w = 32$ . . . . .	100
B.17 Constructed sine waves generated from STSC using $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when $k = 3$ and $w = 64$ . . . . .	101
B.18 Constructed sine waves generated from STSC using $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when $k = 3$ and $w = 128$ . . . . .	102
B.19 Constructed sine waves generated from STSC using $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when $k = 5$ and $w = 64$ . . . . .	103
B.20 Constructed sine waves generated from STSC using $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when $k = 7$ and $w = 64$ . . . . .	104
C.1 Averaged results generated from CDTW function of each dataset . . . . .	106
C.2 Averaged results generated from ICDTW function of each dataset . . . . .	107
C.3 Averaged results generated from NLAAP of each dataset. . . . .	108
D.1 SMMs of AEM2 when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied. . . . .	110
D.2 SMMs of TOR96 when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied. . . . .	110
D.3 SMMs of Buoy1 when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied. . . . .	111
D.4 SMMs of CBF when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied. . . . .	111

Figure	Page
D.5 SMMs of ERP when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied. . . . .	112
D.6 SMMs of Field4 when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied. . . . .	112
D.7 SMMs of Fortune5004 when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied. . . . .	113
D.8 SMMs of MITDBX108 when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied. . . . .	113
D.9 SMMs of AEM2 when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied. . . . .	114
D.10 SMMs of TOR96 when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied. . . . .	114
D.11 SMMs of Buoy1 when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied. . . . .	115
D.12 SMMs of CBF when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied. . . . .	115
D.13 SMMs of ERP when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied. . . . .	116
D.14 SMMs of Field4 when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied. . . . .	116
D.15 SMMs of Fortune5004 when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied. . . . .	117
D.16 SMMs of MITDBX108 when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied. . . . .	117
D.17 Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using CDTW function when $k = 3$ and $w = 32$ . . . . .	118
D.18 Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using CDTW function when $k = 3$ and $w = 64$ . . . . .	119
D.19 Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using CDTW function when $k = 3$ and $w = 128$ . . . . .	120
D.20 Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using ICDTW function when $k = 5$ and $w = 64$ . . . . .	121
D.21 Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using ICDTW function when $k = 7$ and $w = 64$ . . . . .	122

Figure	Page
D.22 Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using ICDTW function when $k = 3$ and $w = 32$ . . . . .	123
D.23 Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using ICDTW function when $k = 3$ and $w = 64$ . . . . .	124
D.24 Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using ICDTW function when $k = 3$ and $w = 128$ . . . . .	125
D.25 Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using ICDTW function when $k = 5$ and $w = 64$ . . . . .	126
D.26 Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using ICDTW function when $k = 7$ and $w = 64$ . . . . .	127
E.1 Computational time of Incremental Shape-based Averaging and Shape-based Averaging with CDTW function when a new incoming sequence arrives. . . . .	129
E.2 Computational time of Incremental Shape-based Averaging and Shape-based Averaging with CDTW function when a new incoming sequence arrives. (cont.) . . . . .	130
E.3 Computational time of Incremental Shape-based Averaging and Shape-based Averaging with ICDTW function when a new incoming sequence arrives. . . . .	131
E.4 Computational time of than Shape-based Averaging around Incremental Shape-based Averaging and Shape-based Averaging with ICDTW function when a new incoming sequence arrives. . . . .	132
F.1 Difference of SUMDIST of Incremental Shape-based Averaging with CDTW when the number of stored sequences to an original dataset is varied. . . . .	134
F.2 Difference of SUMDIST of Incremental Shape-based Averaging with CDTW when the number of stored sequences to an original dataset is varied. (cont.) . . . . .	135
F.3 Difference of SUMDIST of Incremental Shape-based Averaging with ICDTW when the number of stored sequences to an original dataset is varied. . . . .	136
F.4 Difference of SUMDIST of Incremental Shape-based Averaging with ICDTW when the number of stored sequences to an original dataset is varied. (cont.) . . . . .	137
F.5 Speedup of Incremental Shape-based Averaging with CDTW when the number of stored sequences to an original dataset is varied. . . . .	138
F.6 Speedup of Incremental Shape-based Averaging with CDTW when the number of stored sequences to an original dataset is varied. (cont.) . . . . .	139
F.7 Speedup of Incremental Shape-based Averaging with ICDTW when the number of stored sequences to an original dataset is varied. . . . .	140
F.8 Speedup of Incremental Shape-based Averaging with CDTW when the number of stored sequences to an original dataset is varied. (cont.) . . . . .	141



Figure	Page
F.9 Averaged results of some classes from Incremental Shape-based Averaging with CDTW when $\alpha = 1$ . . . . .	142
F.10 Averaged results of some classes from Incremental Shape-based Averaging with CDTW when $\alpha$ is 25% of total number of each class. . . . .	143
F.11 Averaged results of some classes from Incremental Shape-based Averaging with CDTW when $\alpha$ is 50% of total number of each class. . . . .	144
F.12 Averaged results of some classes from Incremental Shape-based Averaging with CDTW when $\alpha$ is 100% of total number of each class. . . . .	145
F.13 Averaged results of some classes from Incremental Shape-based Averaging with ICDTW when $\alpha = 1$ . . . . .	146
F.14 Averaged results of some classes from Incremental Shape-based Averaging with ICDTW when $\alpha$ is 25% of total number of each class. . . . .	147
F.15 Averaged results of some classes from Incremental Shape-based Averaging with ICDTW when $\alpha$ is 50% of total number of each class. . . . .	148
F.16 Averaged results of some classes from Incremental Shape-based Averaging with ICDTW when $\alpha$ is 100% of total number of each class. . . . .	149
G.1 Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where $k = 3$ and $w = 64$ . . . . .	151
G.2 Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where $k = 3$ and $w = 32$ . . . . .	152
G.3 Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where $k = 5$ and $w = 64$ . . . . .	153
G.4 Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where $k = 7$ and $w = 64$ . . . . .	154
G.5 Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where $k = 3$ and $w = 128$ . . . . .	155
G.6 Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where $k = 3$ and $w = 64$ . . . . .	156
G.7 Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where $k = 3$ and $w = 32$ . . . . .	157
G.8 Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where $k = 5$ and $w = 64$ . . . . .	158
G.9 Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where $k = 7$ and $w = 64$ . . . . .	159

Figure	Page
G.10 Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where $k = 3$ and $w = 128$ . . . . .	160
G.11 Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where $k = 3$ and $w = 64$ . . . . .	161
G.12 Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where $k = 3$ and $w = 32$ . . . . .	162
G.13 Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where $k = 5$ and $w = 64$ . . . . .	163
G.14 Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where $k = 7$ and $w = 64$ . . . . .	164
G.15 Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where $k = 3$ and $w = 128$ . . . . .	165
G.16 Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where $k = 3$ and $w = 64$ . . . . .	166
G.17 Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where $k = 3$ and $w = 32$ . . . . .	167
G.18 Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where $k = 5$ and $w = 64$ . . . . .	168
G.19 Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where $k = 7$ and $w = 64$ . . . . .	169
G.20 Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where $k = 3$ and $w = 128$ . . . . .	170
H.1 Percentage difference of SMM and speedup of 3STSC with CDTW function and complete linkage when $k = 3$ , $w = 64$ , and number of stored sequences are varied. . . .	172
H.2 Percentage difference of SMM and speedup of 3STSC with CDTW function and complete linkage when $k = 3$ , $w = 32$ , and number of stored sequences are varied. . . .	173
H.3 Percentage difference of SMM and speedup of 3STSC with CDTW function and complete linkage when $k = 5$ , $w = 64$ , and number of stored sequences are varied. . . .	174
H.4 Percentage difference of SMM and speedup of 3STSC with CDTW function and complete linkage when $k = 7$ , $w = 64$ , and number of stored sequences are varied. . . .	175
H.5 Percentage difference of SMM and speedup of 3STSC with CDTW function and complete linkage when $k = 3$ , $w = 128$ , and number of stored sequences are varied. . . .	176
H.6 Percentage difference of SMM and speedup of 3STSC with CDTW function and average linkage when $k = 3$ , $w = 64$ , and number of stored sequences are varied. . . .	177

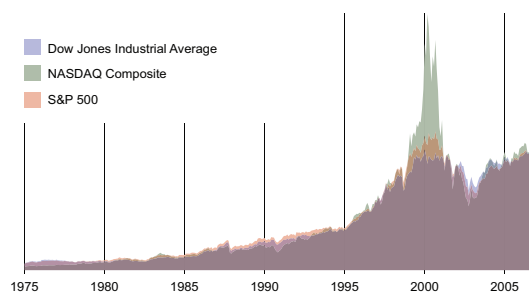
Figure	Page
H.7 Percentage difference of SMM and speedup of 3STSC with CDTW function and average linkage when $k = 3$ , $w = 64$ , and number of stored sequences are varied. . . .	178
H.8 Percentage difference of SMM and speedup of 3STSC with CDTW function and average linkage when $k = 5$ , $w = 64$ , and number of stored sequences are varied. . . .	179
H.9 Percentage difference of SMM and speedup of 3STSC with CDTW function and average linkage when $k = 7$ , $w = 64$ , and number of stored sequences are varied. . . .	180
H.10 Percentage difference of SMM and speedup of 3STSC with CDTW function and average linkage when $k = 3$ , $w = 128$ , and number of stored sequences are varied. . . .	181
H.11 Percentage difference of SMM and speedup of 3STSC with ICDTW function and complete linkage when $k = 3$ , $w = 64$ , and number of stored sequences are varied. . . .	182
H.12 Percentage difference of SMM and speedup of 3STSC with ICDTW function and complete linkage when $k = 3$ , $w = 32$ , and number of stored sequences are varied. . . .	183
H.13 Percentage difference of SMM and speedup of 3STSC with ICDTW function and complete linkage when $k = 5$ , $w = 64$ , and number of stored sequences are varied. . . .	184
H.14 Percentage difference of SMM and speedup of 3STSC with ICDTW function and complete linkage when $k = 7$ , $w = 64$ , and number of stored sequences are varied. . . .	185
H.15 Percentage difference of SMM and speedup of 3STSC with ICDTW function and complete linkage when $k = 3$ , $w = 128$ , and number of stored sequences are varied. . . .	186
H.16 Percentage difference of SMM and speedup of 3STSC with ICDTW function and average linkage when $k = 3$ , $w = 64$ , and number of stored sequences are varied. . . .	187
H.17 Percentage difference of SMM and speedup of 3STSC with ICDTW function and average linkage when $k = 3$ , $w = 32$ , and number of stored sequences are varied. . . .	188
H.18 Percentage difference of SMM and speedup of 3STSC with ICDTW function and average linkage when $k = 3$ , $w = 64$ , and number of stored sequences are varied. . . .	189
H.19 Percentage difference of SMM and speedup of 3STSC with ICDTW function and average linkage when $k = 7$ , $w = 64$ , and number of stored sequences are varied. . . .	190
H.20 Percentage difference of SMM and speedup of 3STSC with ICDTW function and average linkage when $k = 3$ , $w = 128$ , and number of stored sequences are varied. . . .	191

# CHAPTER I

## INTRODUCTION

Time series data mining is an active research area which involves tasks including classification (Ueno et al., 2006; Kasetty et al., 2008; Ratanamahatana and Keogh, 2004; Niennattrakul and Ratanamahatana, 2007c), clustering (Lin et al., 2004b; Yankov and Keogh, 2006; Niennattrakul and Ratanamahatana, 2007b, 2006), anomaly detection (Keogh et al., 2005, 2002; Yankov et al., 2008a; Niennattrakul et al., 2010a), pattern discovery (Chiu et al., 2003; Yankov et al., 2007; Mueen et al., 2009), visualization (Lin et al., 2004a; Kumar et al., 2005), association rules (Sacchi et al., 2007; Wan et al., 2007), and indexing (Keogh et al., 2004; Keogh and Ratanamahatana, 2005; Shieh and Keogh, 2009; Niennattrakul et al., 2010b). Time series is a sequence of real/integer/symbolic values which are sequentially observed, where in some applications, a time series sequence is also considered to be a very high dimensional data object, where the number of dimensions is equal to the length of time series. A characteristic that makes time series differ from other data types is that adjacent dimensions are extremely related; the order of each dimension cannot be swapped. Time series is ubiquitous, where it is easily found in daily life such as stock market, electrocardiogram, and a temperature record, as shown in Figure 1.1. Normally, time series can be collected from scientific measurements such as a star light curve (Protopapas et al., 2005), respiration (Keogh et al., 2005), and winding (B.L.R., 2010). In addition, a 2-D image can be transformed to be time series by sequentially measuring distances from the centroid of an image to the edge (Ye and Keogh, 2011; Yankov et al., 2008b). Therefore, instead of image recognition in 2-D images, time series mining will require much less complexity. A video can also be transformed to a time series sequence by tracking a coordinate of a point of interest. Time series can be multivariate, which at the specific time, many channels from different sources are observed. For example, SmartCane (Wu et al., 2008), a device attached with many types of sensors to help doctors monitor the walk of elderly people (see Figure 1.2), has eight channels of data from two pressure sensors, a three-axis accelerometer, and three single-axis gyros. Data from motion capture (Cai and Ng, 2004) are also considered that each dimension is collected from movement of each sensor.

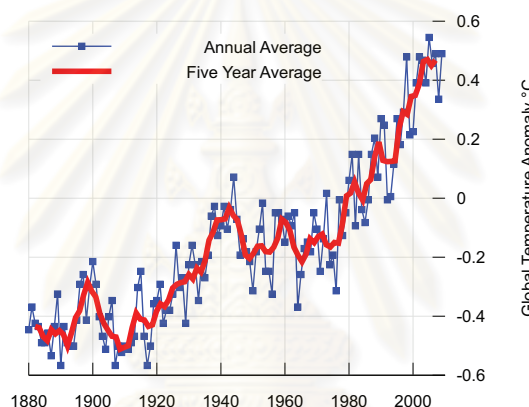
Subsequence clustering for time series data streams is an important data mining task which can return time series patterns in real time. Currently, no streaming subsequence clustering has yet been proposed. As a subsequence clustering result, cluster representatives can then be used in rule discovery (Das et al., 1998; Fu et al., 2001; Harms et al., 2002b,a; Hetland, 2002; Jin et al.,



a) Stock market (Wikipedia.org, 2011b)



b) Electrocardiogram (Wikipedia.org, 2011a)



c) Temperature record (Wikipedia.org, 2011c)

Figure 1.1: Examples of time series data in real world.

2002b,a; Mori and Kuni, 2001; Osaki et al., 2000; Sarker et al., 2003; Uehara and Shimada, 2002; Yairi et al., 2001), indexing (Li et al., 1998; Radhakrishnan et al., 2000), classification (Cotofrei, 2002; Cotofrei and Stoffel, 2002), prediction (Schittenkopf et al., 2000), and anomaly detection (Yairi et al., 2001). However, the current subsequence clustering, Subsequence Time Series Clustering (STSC), has been proved both theoretically and empirically to produce meaningless results, i.e., sine waves regardless of input sequences. Figure 1.3 illustrates cluster representatives from STSC. Therefore, hundreds of works that use STSC as a preprocessing step and a subroutine also produce meaningless results. The causes of meaninglessness are twofold: inappropriate uses of Euclidean distance measure and Amplitude Averaging function. In other words, Euclidean distance and Amplitude Averaging cannot handle trivial-matched subsequences which are a set of contiguous subsequences that are very similar but have shifts in time domain since Euclidean distance and Amplitude Averaging compute dissimilarity and an averaged result in one-to-one manner. Figure 1.4 provides some examples of trivial-matched subsequences.

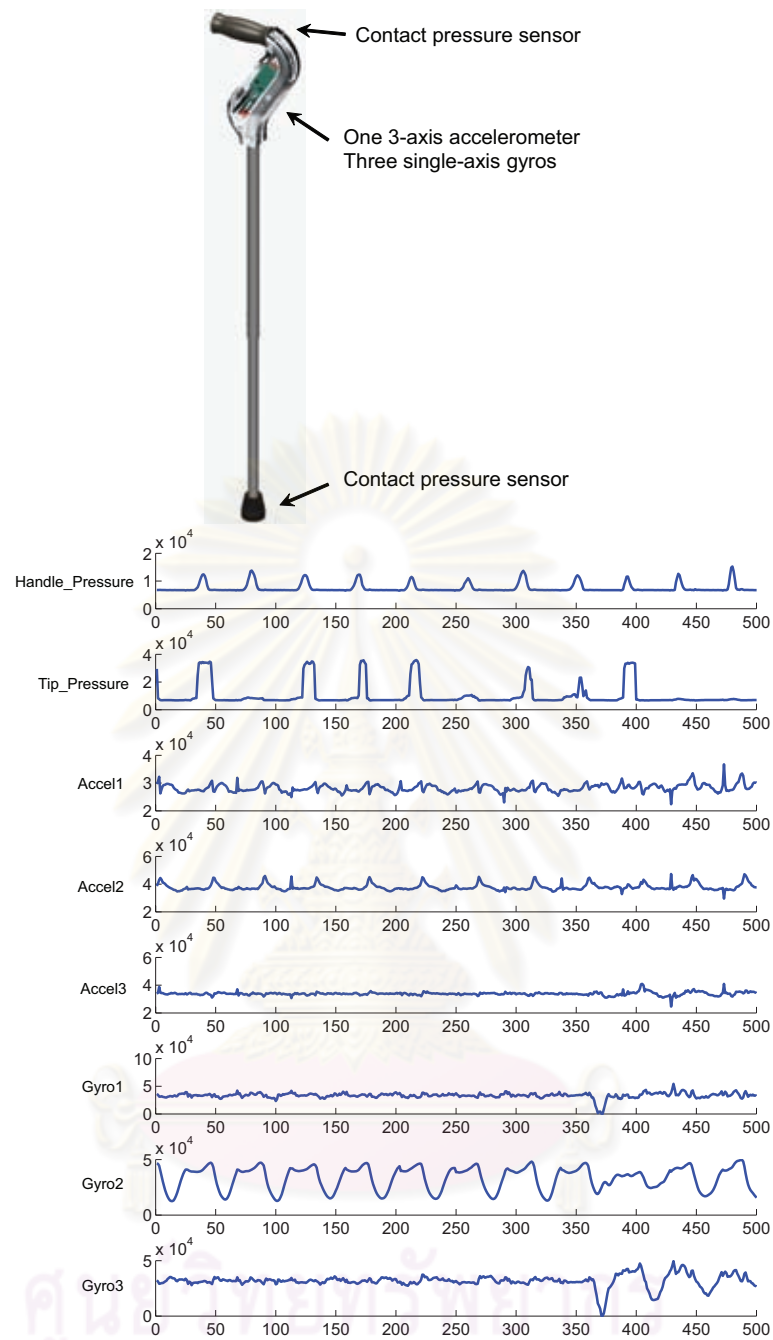


Figure 1.2: Multivariate time series collected from SmartCane system. (Wu et al., 2008)

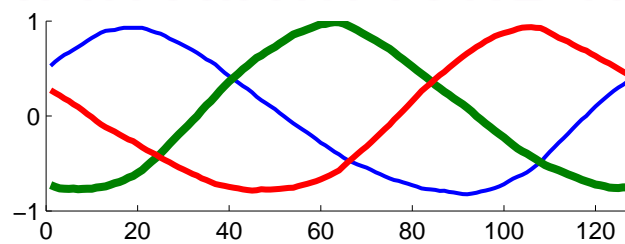


Figure 1.3: Cluster representatives generated from STSC

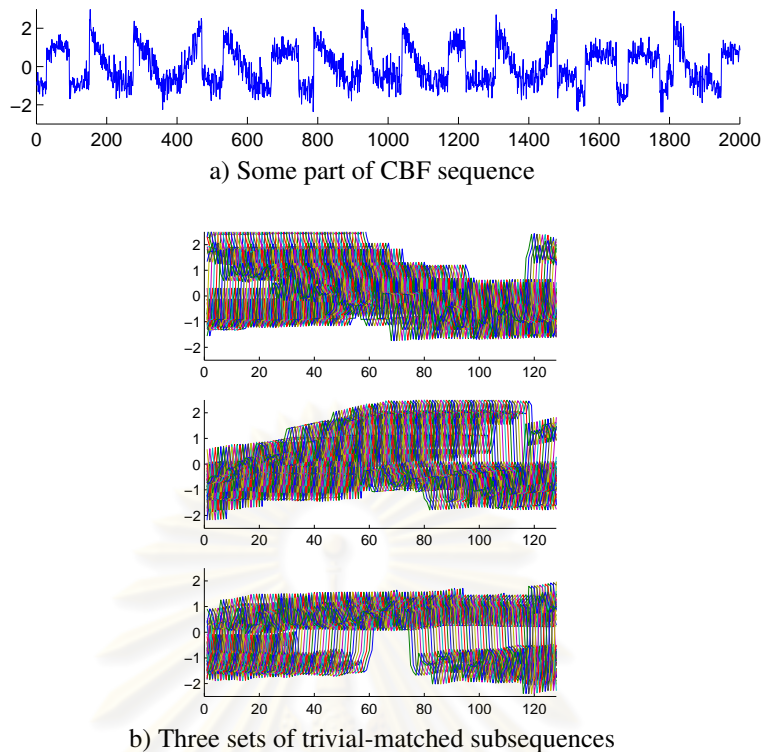


Figure 1.4: Trivial-matched subsequences of CBF sequence

Recently, many researchers (Keogh and Lin, 2005; Denton, 2005; Chen, 2007a; Goldin et al., 2006; Fu et al., 2005; Struzik, 2003; Simon et al., 2006; Kumar et al., 2006; Fujimaki et al., 2008) attempt to overcome this problem by proposing many solutions. However, none of them propose the right solutions to deal with trivial-matched subsequences, i.e., new distance measures requires additional parameters and Amplitude Averaging is still used to create a cluster representative. The distance threshold in Density-based Subsequence Time Series Clustering (DSTSC) (Denton, 2005), the lag value in Lag-based Subsequence Time Series Clustering (LSTSC) (Simon et al., 2006; Chen, 2007b,a), and the slide length in Phrase-Analysis Subsequence Time Series Clustering (PASTSC) (Fujimaki et al., 2008), are additional parameters that users must be specified a priori, depending on characteristics of each dataset, whose values are very sensitive to clustering results. With incorrect values, outputs of clustering results may be meaningless. In addition, these values are used to discard trivial-matched subsequences; therefore, some important trivial-matched subsequences are unexpectedly filtered out. For the meaningfulness measurement, all previous works used Keogh-Lin Meaningfulness Measurement (KLMM) (Keogh et al., 2003) to measure clustering output. However, it will be demonstrated in this work that KLMM is an invalid measurement since it cannot capture similarity of sine waves with different phases and frequencies.

In this work, a novel subsequence clustering for data streams, Shape-based Streaming Sub-

sequence Time Series Clustering (3STSC), is proposed to return a meaningful clustering result in real time. Since all existing subsequence clustering algorithms produce meaningless result, to make subsequence clustering for data streams meaningful, subsequence clustering that produces meaningful results must first be introduced. In this work, a novel subsequence clustering for Shape-based Subsequence Time Series Clustering (2STSC), is firstly proposed. To produce meaningful clustering results, 2STSC utilizes Dynamic Time Warping (DTW) distance and Shape-based Averaging as a distance measure and an averaging function to replace Euclidean distance and Amplitude Averaging, respectively. DTW distance aligns subsequences before distance calculation; therefore, two trivial-matched subsequences are recognized as similar, and Shape-based Averaging aligns subsequences before averaging; therefore, a characteristic-preserved averaging result are returned from two trivial-matched subsequences. 2STSC is evaluated in terms of meaningfulness and this 2STSC is then extended to handle streaming cases in 3STSC.

The remaining of this dissertation is organized as follows. The meaninglessness of Subsequence Time Series Clustering (STSC) with the causes are analyzed and identified in Chapter 2. Shape-based Averaging is first introduced in Chapter 3. The solution to make a clustering result meaningful by Shape-based Subsequence Time Series Clustering (2STSC) is described and evaluated in Chapter 4. Incremental Shape-based Averaging is then proposed to extend Shaped-based Averaging to support streaming applications in Chapter 5. Chapter 6 provides a streaming subsequence clustering algorithm, Shape-based Streaming Subsequence Time Series Clustering (3STSC), which is extended from 2STSC to support streaming applications. And finally, this dissertation is concluded in Chapter 7.

### **1.1 Objective of the Thesis**

The objective of this thesis is to design a novel subsequence clustering algorithm which produces meaningful clustering results for time series data streams.

### **1.2 Scopes of the Thesis**

The scopes of this thesis are as follows:

- This thesis focuses on subsequence clustering for time series data streams, where the stream is univariate and a new data point arrives at a constant rate.
- The datasets from the Time Series Data Mining Archive (TSDMA) are used as benchmarks to evaluate subsequence clustering and streaming clustering, and the datasets from Time Series Clustering/Classification Page are used as benchmarks to evaluate shape-based aver-



aging and incremental shape-based averaging.

- Performance measurements used to evaluate the meaningfulness of the subsequence clustering algorithm is the Shape-based Meaningfulness Measurement (SMM), and the streaming subsequence clustering is evaluated by an actual time improved from the subsequence clustering.

### 1.3 Contributions of the Thesis

The contributions of this thesis are as follows:

- A new meaningfulness measurement is introduced.
- A novel subsequence clustering and a novel streaming subsequence clustering are proposed.
- A novel shape-based averaging and a novel incremental shape-based averaging are introduced.

### 1.4 Research Methodology

- Study background knowledge about time series data mining.
- Survey on potential and related topics including clustering, classification, anomaly detection, indexing, motif discovery, and subsequence matching.
- Review literatures on subsequence clustering algorithm.
- Identify causes of meaninglessness of the current subsequence clustering algorithm.
- Design the shape-based averaging algorithm as a major subroutine of subsequence clustering algorithm to solve the meaninglessness, and evaluate the algorithms with the benchmark datasets.
- Design the shape-based subsequence clustering algorithm that utilizes shape-based averaging algorithm to return a meaningful clustering result, and evaluate the algorithms with the benchmark datasets.
- Design the incremental shape-based averaging algorithm extended from shape-based averaging algorithm to support a streaming application, and evaluate the algorithms with the benchmark datasets.
- Design the shape-based streaming subsequence clustering algorithm extended from shape-based subsequence clustering algorithm to support a streaming application, and evaluate the algorithms with the benchmark datasets.

- Compose the thesis.



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

## CHAPTER II

# MEANINGLESSNESS OF SUBSEQUENCE TIME SERIES CLUSTERING

Subsequence Time Series Clustering (STSC) has been proven both empirically (Peker, 2005; Chen, 2007a; Goldin et al., 2006; Denton, 2005; Keogh et al., 2003; Fujimaki et al., 2008; Kontaki et al., 2008; Chen, 2007b; Simon et al., 2006) and theoretically (Idé, 2006a,b) that its output is meaningless. Keogh and Lin (Keogh and Lin, 2005) first flagged this issue by observation that STSC always produced a set of sine waves as cluster representatives instead of expected patterns from a time series sequence. In addition, they also proposed a meaningfulness measurement, so-called Keogh-Lin Meaningfulness Measurement (KLMM). Specifically, KLMM defines that cluster representatives should be similar if the representatives are from the same input sequence, and cluster representatives should be dissimilar if the representatives are from different input sequences. However, this thesis argues that KLMM is an invalid measurement for two reasons. First, although cluster representatives from different input sequences are sine waves, these sine waves may have different phases and frequencies. Second, KLMM only measures clustering results without considering how similar input sequences are; similarity between two input sequences are not defined for KLMM. For example, clustering results from two similar sequences must be very similar, but they are considered meaningless in the view of KLMM, even a clustering algorithm does produce a meaningful result. In this chapter, the meaningfulness of clustering results of STSC will be demonstrated, and KLMM will be shown that it is an invalid meaningfulness measurement.

### 2.1 Background

In this section, background knowledge of Subsequence Time Series Clustering (STSC),  $k$ -hierarchical clustering,  $k$ -means clustering, Euclidean distance, and Amplitude Averaging is provided to give better understanding of STSC's the meaningfulness.

#### 2.1.1 Subsequence Time Series Clustering (STSC)

Subsequence Time Series Clustering (STSC) has been proposed to discover patterns or to group subsequences as a part of a subroutine or a preprocessing step of various mining tasks such as rule discovery (Das et al., 1998; Fu et al., 2001; Harms et al., 2002b,a; Hetland, 2002; Jin et al.,

2002b,a; Mori and Kuni, 2001; Osaki et al., 2000; Sarker et al., 2003; Uehara and Shimada, 2002; Yairi et al., 2001), indexing (Li et al., 1998; Radhakrishnan et al., 2000), classification (Cotofrei, 2002; Cotofrei and Stoffel, 2002), prediction (Schittenkopf et al., 2000), and anomaly detection (Yairi et al., 2001). Given a time series sequences  $S = \langle s_1, s_2, \dots, s_n \rangle$  of length  $n$ , STSC first extracts a set  $\mathbb{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_i, \dots, \mathcal{S}_{n-w+1}\}$  of subsequences using a fixed-length sliding window, where a subsequence  $\mathcal{S}_i = \langle s_i, s_{i+1}, \dots, s_{i+w-1} \rangle$ ,  $1 \leq i \leq n-w+1$ , and  $w$  is the sliding window length. Then every subsequence is normalized by  $z$ -normalization (see Section 2.1.6), and subsequences are clustered by  $k$ -hierarchical clustering or  $k$ -means clustering algorithms with Euclidean distance and Amplitude Averaging as a distance measure and an averaging function. In addition, Euclidean distance is used to calculate similarity between two subsequences and Amplitude Averaging function is used to construct a cluster representative for each cluster. STSC finally returns a set of clusters returned from  $k$ -hierarchical clustering or  $k$ -means clustering. Formally, STSC receives a long time series  $S$  with two parameters, i.e., the number of clusters ( $k$ ) and the length of a sliding window ( $w$ ), and returns a set  $\mathbb{C} = \{C_1, C_2, \dots, C_i, \dots, C_k\}$  of clusters, where each cluster  $C_i = (\mathbb{M}, R)$  contains cluster members  $\mathbb{M} = \{\mathcal{S}_i \mid \mathcal{S}_i \in \mathbb{S}\}$  and a cluster representative  $R = \langle r_1, r_2, \dots, r_w \rangle$ . Pseudo code of STSC is provided in Table 2.1 and Figure 2.1 visualizes an overview of STSC.

Table 2.1: Pseudo code of Subsequence Time Series Clustering (STSC)

FUNCTION $[\mathbb{C}] = \text{SUBSEQUENCETIME SERIES CLUSTERING } [S, k, w]$
1. $\mathbb{S} = \text{EXTRACTSUBSEQUENCES}(S, w)$
2. $\mathbb{S}_{Norm} = \text{NORMALIZESUBSEQUENCES}(\mathbb{S})$
3. $\mathbb{C} = \text{CLUSTERING}(\mathbb{S}_{Norm}, k)$ // with Euclidean distance and Amplitude Averaging
4. Return $\mathbb{C}$

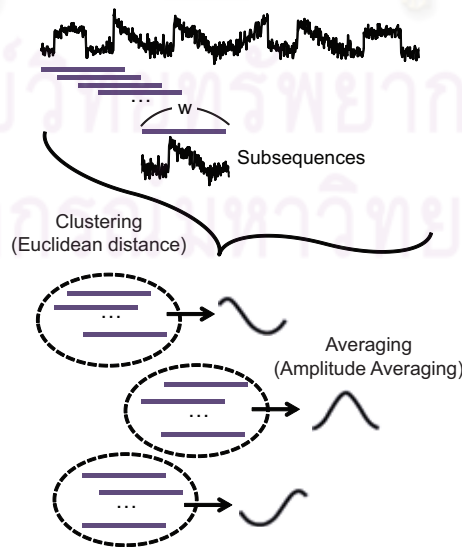


Figure 2.1: Overview of Subsequence Time Series Clustering (STSC)

### 2.1.2 $K$ -Hierarchical Clustering

$K$ -hierarchical clustering used in STSC is an agglomerative clustering algorithm. AGNES (AGglomerative NESTing) is a well-known hierarchical clustering algorithms that can visualize relationships among data sequences in a hierarchical structure or a tree-based structure on distance calculations. Although many variations of hierarchical clustering algorithms have been introduced such as BIRCH (Zhang et al., 1996) (Balanced Iterative Reducing and Clustering Using Hierarchies), ROCK (Guha et al., 2000) (A Hierarchical Clustering Algorithm for Categorical Attributes), and Chameleon (Karypis et al., 1999) (A Hierarchical Clustering Algorithm Using Dynamic Modeling), AGNES is commonly used due to implementation simplicity.

Specifically, AGNES has been proposed to group data using bottom-up strategy. The method iteratively merges two atomic clusters into a larger cluster until one single cluster containing every data sequences is achieved. For each iteration, two clusters which have minimum inter-cluster distance are merged. However, grouping a dataset into one single cluster for agglomerative clustering is impractical; therefore, the number of clusters ( $k$ ) is required. Concretely, pseudo codes of the agglomerative clustering algorithm which receives a set  $\mathbb{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_i, \dots, \mathcal{S}_n\}$  of time series sequences as an input and returns a set  $\mathbb{C} = \{C_1, C_2, \dots, C_i, \dots, C_k\}$  of  $k$  clusters as an output, where each  $C_i = (\mathbb{M}, R)$  contains a set  $\mathbb{M} = \{\mathcal{S}_i \mid \mathcal{S}_i \in \mathbb{S}\}$  of time series sequences and a cluster representative  $R$ , are shown in Table 2.2.

Table 2.2: Agglomerative hierarchical clustering algorithm (AGNES)

FUNCTION [C] = AGGLOMERATIVECLUSTERING [S, k]	
1.	Initialize a set $\mathbb{C}$ of clusters which contains one sequence from $\mathbb{S}$
2.	While (the size of $\mathbb{C} > k$ )
3.	$dist_{best} = \text{INFINITY}$
4.	For each pair of $C_i$ and $C_j$ in $\mathbb{C}$
5.	$dist = \text{INTERCLUSTERDISTANCE}(C_i, C_j)$
6.	if ( $dist < dist_{best}$ )
7.	$dist_{best} = dist$
8.	$pair_{best} = [C_i, C_j]$
9.	Endif
10.	Endfor
11.	$[C_i, C_j] = pair_{best}$
12.	$C_k = \text{MERGE}(C_i, C_j)$
13.	Remove $C_i$ and $C_j$ from $\mathbb{C}$
14.	Add $C_k$ to $\mathbb{C}$
15.	Endwhile
16.	For each cluster $C$ in $\mathbb{C}$
17.	$C.R = \text{AVERAGE}(C.M)$
18.	Endfor
19.	Return $\mathbb{C}$

While many similarity functions between two clusters (called inter-cluster distances) have been proposed, three functions are typically used, i.e., single linkage, complete linkage, and average linkage inter-cluster distance functions. Single linkage function returns a minimum distance among all possible pairs between two clusters, while complete linkage function returns a maximum distance among all possible pairs between two clusters. On the other hand, average linkage function finds a mean value of all distances. Pseudo codes of single, complete, and average linkage distance functions are provided in Table 2.3, 2.4, and 2.5, respectively, and these inter-cluster distances are formalized as follows.

$$D_{single}(C_i, C_j) = \min_{\mathcal{S} \in \mathbb{M}_i, \mathcal{S}' \in \mathbb{M}_j} Distance(\mathcal{S}, \mathcal{S}') \quad (2.1)$$

$$D_{complete}(C_i, C_j) = \max_{\mathcal{S} \in \mathbb{M}_i, \mathcal{S}' \in \mathbb{M}_j} Distance(\mathcal{S}, \mathcal{S}') \quad (2.2)$$

$$D_{average}(C_i, C_j) = \frac{1}{|\mathbb{M}_i| |\mathbb{M}_j|} \sum_{c \in C_i} \sum_{c' \in C_j} Distance(\mathcal{S}, \mathcal{S}') \quad (2.3)$$

where  $D_{single}$ ,  $D_{complete}$ , and  $D_{average}$  are single, complete, and average linkage distance functions, respectively,  $C_i$  and  $C_j$  are any clusters,  $\mathbb{M}_i$  and  $\mathbb{M}_j$  are corresponding cluster members of  $C_i$  and  $C_j$ , respectively, and  $\mathcal{S}$  and  $\mathcal{S}'$  are sequences in  $\mathbb{M}_i$  and  $\mathbb{M}_j$ , respectively.  $Distance(\mathcal{S}, \mathcal{S}')$  is a distance function that returns a distance between two sequences  $\mathcal{S}$  and  $\mathcal{S}'$ .

Table 2.3: Pseudo code of single linkage distance function

FUNCTION [ $dist_{best}$ ] = SINGLELINKAGE [ $C_i, C_j$ ]	
1.	$\mathbb{M}_i$ is a set of cluster member of $C_i$
2.	$\mathbb{M}_j$ is a set of cluster member of $C_j$
3.	$dist_{best} = \text{INFINITY}$
4.	For each sequence $\mathcal{S}$ in $\mathbb{M}_i$
5.	For each sequence $\mathcal{S}'$ in $\mathbb{M}_j$
6.	$dist = \text{DISTANCE}(\mathcal{S}, \mathcal{S}')$
7.	if ( $dist < dist_{best}$ )
8.	$dist_{best} = dist$
9.	Endif
10.	Endfor
11.	Endfor
12.	Return $dist_{best}$

For Subsequence Time Series Clustering (STSC), Euclidean distance and Amplitude Averaging is used as a distance function and an averaging function.

Table 2.4: Pseudo code of complete linkage distance function

---

FUNCTION [ $dist_{best}$ ] = COMPLETELINKAGE [ $C_i, C_j$ ]

---

1.  $\mathbb{M}_i$  is a set of cluster member of  $C_i$
2.  $\mathbb{M}_j$  is a set of cluster member of  $C_j$
3.  $dist_{best} = \text{INFINITY}$
4. For each sequence  $\mathcal{S}$  in  $\mathbb{M}_i$
5.     For each sequence  $\mathcal{S}'$  in  $\mathbb{M}_j$
6.          $dist = \text{DISTANCE}(\mathcal{S}, \mathcal{S}')$
7.         if ( $dist > dist_{best}$ )
8.              $dist_{best} = dist$
9.         Endif
10.     Endfor
11. Endfor
12. Return  $dist_{best}$

---

Table 2.5: Pseudo code of average linkage distance function

---

FUNCTION [ $dist_{avg}$ ] = AVERAGELINKAGE [ $C_i, C_j$ ]

---

1.  $\mathbb{M}_i$  is a set of cluster member of  $C_i$
2.  $\mathbb{M}_j$  is a set of cluster member of  $C_j$
3.  $dist_{avg} = 0$
4. For each sequence  $\mathcal{S}$  in  $\mathbb{M}_i$
5.     For each sequence  $\mathcal{S}'$  in  $\mathbb{M}_j$
6.          $dist_{avg} = dist_{avg} + \text{DISTANCE}(\mathcal{S}, \mathcal{S}')$
7.     Endfor
8. Endfor
9.  $dist_{avg} = dist_{avg} / (|\mathbb{M}_i| |\mathbb{M}_j|)$
10. Return  $dist_{avg}$

---

### 2.1.3 K-Means Clustering

$K$ -means clustering algorithm (Lloyd, 1982; MacQueen, 1967) is a partitioning clustering that finds a group of clusters by iteratively refining members in each cluster to have the maximum objective value that minimizes summation of distances between a cluster representative and cluster members for every cluster. Beside  $k$ -means clustering, many partitioning clustering algorithms are proposed including  $k$ -medoids clustering (Kaufman and Rousseeuw, 2005) and CLARANS (Kaufman and Rousseeuw, 2005). Both  $k$ -medoids and CLARAN use a median of cluster members instead of a mean. However, a median cannot reflect all characteristics of all data sequences of a cluster because a median is selected from one of existing data sequences, while a mean is a sequence constructed by averaging all data sequences within a cluster. Therefore,  $k$ -means clustering is much more preferable than  $k$ -medoids and CLARAN.

Initially,  $k$ -means clustering first selects  $k$  centers by randomizing existing data sequences from a set  $\mathbb{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_i, \dots, \mathcal{S}_n\}$  of sequences, where  $\mathcal{S}_i = \langle s_1, s_2, \dots, s_w \rangle$  is a time series sequence of length  $w$ , and then remaining sequences are assigned to the closest cluster center,

where  $k$  is a user-defined number of clusters. After that, a new cluster center is calculated by averaging all cluster members within each cluster. The algorithm repeats assigning data sequences to the closest center and recalculating for cluster centers until the clustering result remains unchanged. When the algorithm terminates, a set  $\mathbb{C} = \{C_1, C_2, \dots, C_i, \dots, C_k\}$  of clusters, where each cluster  $C_i = (\mathbb{M}, R)$  contains a set  $\mathbb{M} = \{\mathcal{S}_i \mid \mathcal{S}_i \in \mathbb{S}\}$  of cluster members and a cluster representative  $R = \langle r_1, r_2, \dots, r_w \rangle$  is returned. To be more concrete, pseudo code of  $k$ -means clustering is provided in Table 2.6.

Table 2.6: Pseudo code of  $k$ -means clustering

FUNCTION $[\mathbb{C}] = \text{KMEANSCLUSTERING} [\mathbb{S}, k]$	
1.	Initialize a set $\mathbb{C}$ of $k$ cluster centers with existing sequence in $\mathbb{S}$
2.	Do
3.	For each sequence $\mathcal{S}$ in $\mathbb{S}$
4.	$dist_{best} = \text{INFINITY}$
5.	For each cluster $C$ in $\mathbb{C}$
6.	$R = \text{Cluster representative of } C$
7.	$dist = \text{DISTANCE}(\mathcal{S}, R)$
8.	If ( $dist < dist_{best}$ )
9.	$dist_{best} = dist$
10.	$C_{best} = C$
11.	Endif
12.	Endfor
13.	Assign $\mathcal{S}$ to $C_{best}$
14.	Endfor
15.	For each cluster $C$ in $\mathbb{C}$
16.	$C.R = \text{AVERAGE}(C.\mathbb{M})$
17.	Endfor
18.	While (all cluster members in $\mathbb{C}$ change)
19.	Return $\mathbb{C}$

Subsequence Time Series Clustering (STSC) with  $k$ -means clustering uses Euclidean distance and Amplitude Averaging as a distance measure and an averaging function, respectively.

#### 2.1.4 Euclidean Distance

Euclidean distance (Keogh and Ratanamahatana, 2005) is a well-known similarity measure used in many domains including time series data. The distance is calculated in one-to-one manner shown in Figure 2.2, where the distance is a summation of difference between two data points in the same dimension. Euclidean distance between two time series sequences  $A$  and  $B$  is calculated by the following equation.

$$\text{Euclidean}(A, B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$



where  $A = \langle a_1, a_2, \dots, a_i, \dots, a_n \rangle$  and  $B = \langle b_1, b_2, \dots, b_i, \dots, b_n \rangle$  are two time series sequences of length  $n$ .

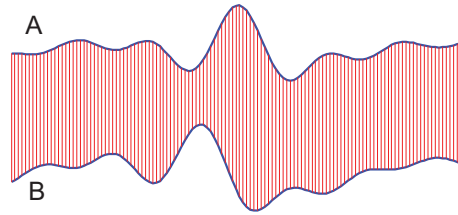
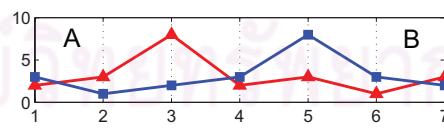


Figure 2.2: Example of Euclidean distance calculation.

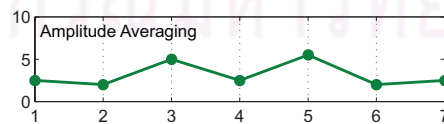
For Subsequence Time Series Clustering (STSC), Euclidean distance is used as a distance measure in  $k$ -means clustering and  $k$ -hierarchical clustering algorithms. However, at the end of this chapter (Section 2.4), Euclidean distance will be shown that it is a cause that makes a clustering result of STSC meaningless.

### 2.1.5 Amplitude Averaging

Amplitude Averaging function is a method to construct a mean of a set of time series sequences, where a value of each dimension of a mean is derived from averaging all values of the same dimension for all sequences. A mean  $Z = \langle z_1, z_2, \dots, z_i, \dots, z_n \rangle$  from Amplitude Averaging of two time series sequences  $A = \langle a_1, a_2, \dots, a_i, \dots, a_n \rangle$  and  $B = \langle b_1, b_2, \dots, b_i, \dots, b_n \rangle$  of length  $n$  is calculated by  $z_i = \frac{a_i + b_i}{2}$ . The example is shown in Figure 2.3; a mean sequence is generated from two sequences  $A = \langle 2, 3, 8, 2, 1, 3 \rangle$  and  $B = \langle 3, 1, 2, 8, 3, 2 \rangle$  by Amplitude Averaging function.



a) Original sequences  $A$  and  $B$



b) Averaged result generated from Amplitude Averaging

Figure 2.3: Example of Amplitude Averaging calculation.

However, if two sequences  $A = \langle a_1, a_2, \dots, a_i, \dots, a_n \rangle$  and  $B = \langle b_1, b_2, \dots, b_i, \dots, b_n \rangle$  have different weights,  $\omega_A$  and  $\omega_B$ , respectively, a mean sequence  $Z = \langle z_1, z_2, \dots, z_i, \dots, z_n \rangle$  can be computed by  $z_i = \frac{\omega_A \cdot a_i + \omega_B \cdot b_i}{\omega_A + \omega_B}$ . And for averaging a set  $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_j, \dots, \mathcal{S}_m\}$  of

sequences, a mean sequence  $Z = \langle z_1, z_2, \dots, z_i, \dots, z_n \rangle$  can be computed once by  $z_i = \frac{\sum_{S \in \mathcal{S}} s_i}{|\mathcal{S}|}$ , and the pseudo code is provided in Table 2.7.

Table 2.7: Pseudo code of Amplitude Averaging function

FUNCTION [Z] = AMPLITUDEAVERAGING [S]	
1.	Initialize the sequence $Z$ to all zeros
2.	For each sequence $S$ in $\mathcal{S}$
3.	For each data point $s_i$ in $\mathcal{S}$
4.	$z_i = z_i + s_i$
5.	Endfor
6.	Endfor
7.	For each data point $z_i$ in $Z$
8.	$z_i = z_i /  \mathcal{S} $
9.	Endfor
10.	Return $Z$

For Subsequence Time Series Clustering (STSC), Amplitude Averaging function is used as an averaging function to construct a cluster representative; however, in this section, Amplitude Averaging will be shown that it is one of the causes that makes the output of STSC meaningless.

### 2.1.6 Z-Normalization

Normalization is a function to rescale a sequence to a specific range. In data mining, many normalization techniques (Han and Kamber, 2000) have been proposed such as min-max normalization, sigmoid normalization, and  $z$ -normalization. For time series data,  $z$ -normalization is typically used to remove an offset and imbalanced distribution. In addition, the sequence is normalized to obtain a mean and a standard deviation of zero and one, respectively. Given a sequence  $A = \langle a_1, a_2, \dots, a_i, \dots, a_n \rangle$  of length  $n$ , a new sequence  $Z = \langle z_1, z_2, \dots, z_i, \dots, z_n \rangle$  is normalized according to the following equations.

$$z_i = \frac{a_i - \mu_A}{\sigma_A} \quad (2.4)$$

$$\mu_A = \frac{\sum_{i=1}^n a_i}{n} \quad (2.5)$$

$$\sigma_A = \sqrt{\frac{\sum_{i=1}^n (a_i - \mu_A)^2}{n}} \quad (2.6)$$

where  $\mu_A$  and  $\sigma_A$  are a mean and a standard deviation of the sequence  $A$ , respectively.

Example is shown in Figure 2.4, where the original sequence is normalized to have its mean and standard deviation of zero and one, respectively.

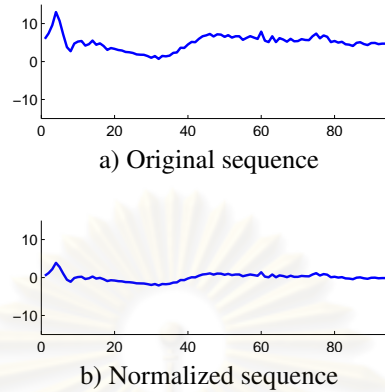


Figure 2.4: Example of  $z$ -normalization.

For Subsequence Time Series Clustering (STSC), a set of sequences extracted from a long time series sequences needs to be normalized before clustering with  $k$ -means clustering and  $k$ -hierarchical clustering algorithms. If normalization is not applied, subsequence clustering will produce undesired results since similarity between subsequences must be independent to mean and standard deviation of subsequences.

## 2.2 Related Work

Keogh and Lin have published a paper describing that an output of Subsequence Time Series Clustering (STSC) is a set of sine waves that is considered meaningless (Keogh and Lin, 2005). This leads to many arguments in data mining community since STSC has been implemented as a subroutine and a preprocessing step of hundreds of mining applications such as rule discovery (Das et al., 1998; Fu et al., 2001; Harms et al., 2002b,a; Hetland, 2002; Jin et al., 2002b,a; Mori and Kuni, 2001; Osaki et al., 2000; Sarker et al., 2003; Uehara and Shimada, 2002; Yairi et al., 2001), indexing (Li et al., 1998; Radhakrishnan et al., 2000), classification (Cotofrei, 2002; Cotofrei and Stoffel, 2002), prediction (Schittenkopf et al., 2000), and anomaly detection (Yairi et al., 2001). Since Keogh and Lin proved that STSC is meaningless, all the works and that successors utilized STSC are also considered invalid. Generally, STSC extracts subsequences from a long time series as an input and returns a set of clusters as an output. Keogh and Lin found that although an input changes, an output remains the same; in other words, STSC always produces the similar sine waves as cluster representatives regardless of a data input of the clustering algorithm.

Keogh and Lin claim that STSC is meaningless by the following experiment. Thirty each of three patterns, i.e., Cylinder, Bell, and Funnel (Saito, 1994), of length 128, shown in Figure 2.5, generated from the following equations are concatenated to create a long sequence in Figure 2.6.

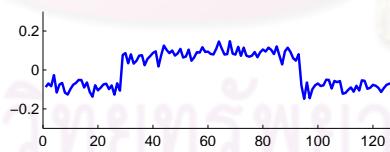
$$c(t) = (6 + \eta) \cdot \chi[a, b](t) + \epsilon(t) \quad (2.7)$$

$$b(t) = (6 + \eta) \cdot \chi[a, b](t) \cdot (t - a) / (b - a) + \epsilon(t) \quad (2.8)$$

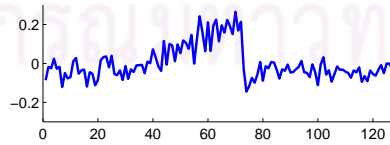
$$f(t) = (6 + \eta) \cdot \chi[a, b](t) \cdot (b - t) / (b - a) + \epsilon(t) \quad (2.9)$$

$$\chi[a, b] = \begin{cases} 0 & t < a \\ 1 & a \leq t \leq b \\ 0 & t > b \end{cases} \quad (2.10)$$

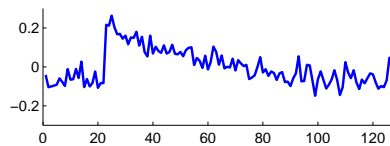
where  $\eta$  and  $\epsilon(t)$  are drawn from a standard normal distribution  $N(0, 1)$ ,  $a$  is an integer drawn uniformly from  $[16, 32]$ ,  $b - a$  is an integer drawn uniformly from  $[32, 96]$ , and  $t$  is varied from 1 to 128.



a) Cylinder



b) Bell



c) Funnel

Figure 2.5: Examples of Cylinder-Bell-Funnel dataset

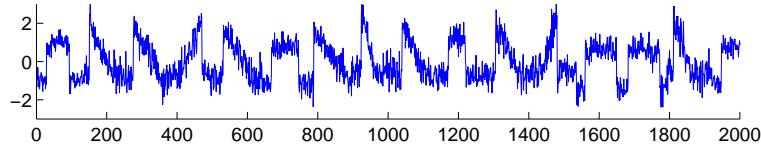


Figure 2.6: Some part of Cylinder-Bell-Funnel sequence

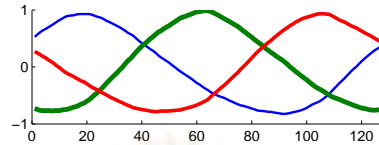


Figure 2.7: Cluster representatives generated from STSC

When this sequence is clustered by STSC, sine-wave-like cluster representatives (see Figure 2.7) are returned, while original patterns are expected to be a result. Keogh and Lin also propose a meaningfulness measurement, so-called Keogh-Lin Meaningfulness Measurement (KLMM), defining that the subsequence clustering is meaningful when the clustering algorithm returns similar cluster representatives from the same input sequence and dissimilar cluster representatives from different input sequences. Suppose  $\mathbb{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n\}$  and  $\mathbb{Y} = \{\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_n\}$  are two sets of clustering results from  $n$  different runs of two different datasets, where  $\mathcal{X} = \{\hat{X}_1, \hat{X}_2, \dots, \hat{X}_k\}$  and  $\mathcal{Y} = \{\hat{Y}_1, \hat{Y}_2, \dots, \hat{Y}_k\}$  are two sets of cluster representatives, respectively. The meaningfulness of KLMM can be calculated from the following equations.

$$WithinDistance(\mathbb{X}) = \frac{\sum_{i=1}^k \sum_{j=1}^k ClusterDistance(\mathcal{X}_i, \mathcal{X}_j)}{k^2} \quad (2.11)$$

$$BetweenDistance(\mathbb{X}, \mathbb{Y}) = \frac{\sum_{i=1}^k \sum_{j=1}^k ClusterDistance(\mathcal{X}_i, \mathcal{Y}_j)}{k^2} \quad (2.12)$$

$$KLMM(\mathbb{X}, \mathbb{Y}) = \frac{WithinDistance(\mathbb{X})}{BetweenDistance(\mathbb{X}, \mathbb{Y})} \quad (2.13)$$

where  $WithinDistance(\mathbb{X})$  is a distance between sets of cluster representatives from the same input sequence,  $BetweenDistance(\mathbb{X}, \mathbb{Y})$  is a distance between sets of cluster representatives from different input sequences, and  $ClusterDistance(\mathcal{A}, \mathcal{B})$  can be calculated from the summation of minimum distances between two sets of cluster representatives. The result is meaningful when KLMM returns the value close to zero since  $WithinDistance(\mathbb{X})$  is small and  $BetweenDistance(\mathbb{X}, \mathbb{Y})$  is very large; otherwise, the result is meaningless.

$ClusterDistance(\mathcal{A}, \mathcal{B})$  can be formalized as the following equation.

$$ClusterDistance(\mathcal{A}, \mathcal{B}) = \sum_{i=1}^k \min [EuclideanDistance(A_i, B_j)], 1 \leq j \leq k \quad (2.14)$$

where  $\mathcal{A} = \{A_1, A_2, \dots, A_i, \dots, A_k\}$  and  $\mathcal{B} = \{B_1, B_2, \dots, B_j, \dots, B_k\}$  are two sets of cluster representatives.

However, KLMM is an invalid meaningfulness measurement for two reasons. The first reason is that with the same number of clusters and the same length of sliding window, cluster representatives of two different input sequences may be sine waves with different phases and frequencies. STSC always produces sine waves regardless of an input sequence; therefore, if cluster representatives are sine waves, the clustering result would mistakenly be considered as meaningless. However, Euclidean distance utilized by KLMM cannot capture similarity between two sine waves with different phases and frequencies; therefore, KLMM considers clustering results are meaningful although results are all sine waves.

Secondly, KLMM assumes that two clustering results are meaningful if they are different. For any meaningful subsequence clustering algorithm, if two input sequences are similar, the clustering results are expected to be similar as well, and if two input sequences are different, the clustering results are expected to be different, but KLMM will always flag any two similar clustering results as meaningless regardless of similarity between two input sequences. Although a meaningful subsequence clustering algorithm exists, KLMM cannot tell how meaningful they are.

Many successor papers in finding a meaningful subsequence clustering also unawares use KLMM as a meaningfulness measurement to evaluate their algorithms; therefore, their experiments become invalid. For theoretical study, Ide (Idé, 2006b) proved that STSC always returns sine waves regardless of an input sequence. In this thesis, a new meaningfulness measure will be introduced in Chapter 4 to be used as a meaningfulness measurement for Shape-based Subsequence Time Series Clustering (2STSC).

### 2.3 Experiments

Two following experiments will demonstrate that STSC produces meaningless clustering results and that KLMM is an invalid meaningfulness measurement. Datasets used in these exper-

iments are eight time series of length 2000 from the Time Series Data mining Archive (TSDMA) (Keogh and Folias, 2011) shown in Figure A.1. Figure 2.8 shows Buoy1 and CBF used in the experiments.

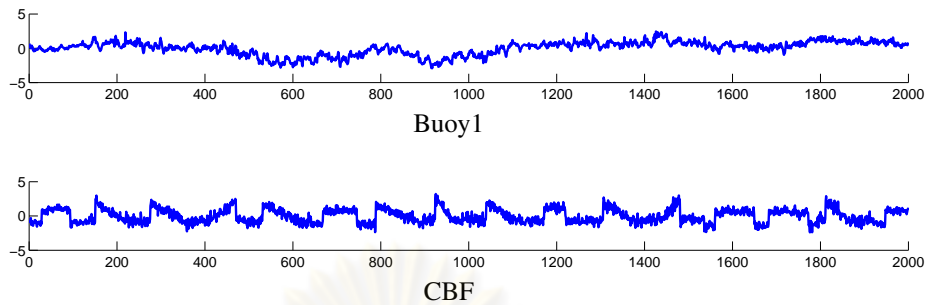


Figure 2.8: Datasets from TSDMA used in the experiments.

### 2.3.1 First Experiment

The first experiment demonstrates that STSC produces clustering results as sine waves regardless of an input sequence. The number of clusters ( $k$ ) and the length of sliding window ( $w$ ) vary. In addition, to show that cluster representatives are sine waves, perfect sine waves are constructed and compared to these cluster representatives. Generally, a sine wave can be formalized as a following equation (Hazewinkel, 2001).

$$y(x) = A \cdot \sin(\omega x + \varphi) + \mu \quad (2.15)$$

where  $A$  is the amplitude,  $\omega = 2\pi f$  is the angular frequency (in radian per second),  $f$  is the ordinary frequency (in hertz),  $\varphi$  is phase, and  $\mu$  is an offset of the sine wave.

Given a set  $\mathbb{R} = \{R_1, R_2, \dots, R_k\}$  of  $k$  cluster representatives, a new set  $\mathbb{R}' = \{R'_1, R'_2, \dots, R'_k\}$  of  $k$  cluster representatives is constructed by searching for those parameters by a non-linear equation solver (Balda, 1999) implemented with Levenberg-Marquardt algorithm (Fletcher, 1971) to minimize Root Mean Square Error (RMSE).

Figures 2.9 and 2.10 show cluster representatives generated from STSC of two datasets, i.e., Buoy1 and CBF, using  $k$ -means clustering and  $k$ -hierarchical clustering (with two variations of inter-cluster distance functions) when  $k = 3$  and  $w = 64$ . Note that single linkage distance function is not used as an inter-distance function in this experiment because  $k$ -hierarchical clustering with single linkage function cannot gracefully handle trivial-matched subsequences, where some subsequences will never in any groups if these subsequences have the largest nearest neighbor distance compared with other subsequences. In other words, single linkage group subsequences

based on the smallest nearest neighbor distance. Therefore, in this study, only two inter-cluster distance functions are utilized, i.e., complete linkage and average linkage functions. The constructed sine waves from cluster representatives generated from STSC of two datasets, i.e., Buoy1 and CBF, using  $k$ -means clustering and  $k$ -hierarchical clustering are shown in Figures 2.11 and 2.12, respectively, when  $k = 3$  and  $w = 64$ , where thick lines are constructed sine waves, and thin lines are original cluster representatives. The complete experiment results of eight datasets are provided in Appendix B, where the number of clusters ( $k$ ) and the length of sliding window ( $w$ ) are varied to be (3, 32), (3, 64), (5, 64), (7, 64), and (3, 128), respectively.

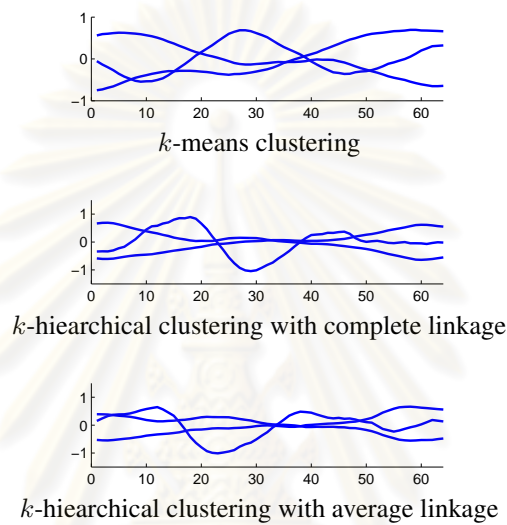


Figure 2.9: Cluster representatives generated from STSC of Buoy1 when  $k = 3$  and  $w = 64$ .

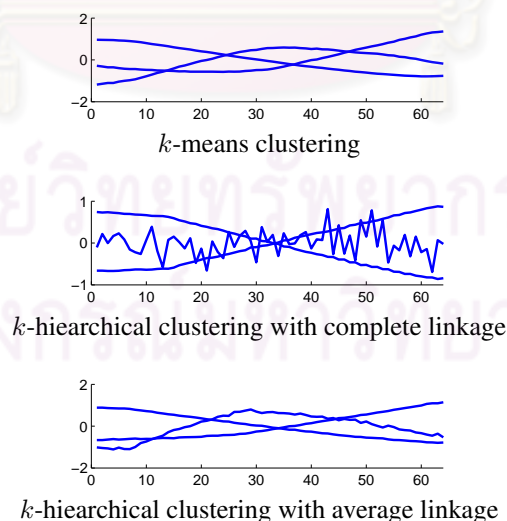


Figure 2.10: Cluster representatives generated from STSC of CBF when  $k = 3$  and  $w = 64$ .



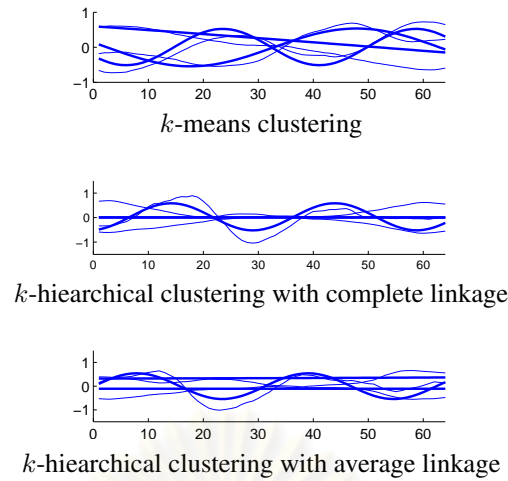


Figure 2.11: Constructed sine waves generated from STSC of Buoy1 when  $k = 3$  and  $w = 64$ .

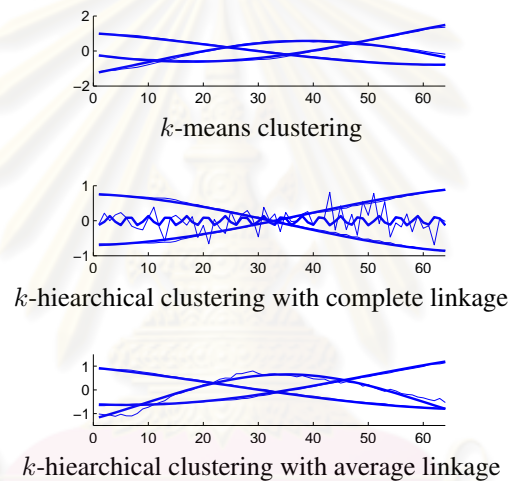


Figure 2.12: Constructed sine waves generated from STSC of CBF when  $k = 3$  and  $w = 64$ .

### 2.3.2 Second Experiment

The second experiment demonstrates that KLMM is an invalid meaningfulness measurement. From the first experiment, clustering results of STSC are meaningless because STSC produces sine waves as cluster representatives. However, KLMM does not capture that the result is a set of sine waves, but KLMM calculates the difference between two cluster representatives using Euclidean distance. Since STSC has been proven both empirically and theoretically that it produces sine waves regardless of inputs (Idé, 2006b; Keogh and Lin, 2005), KLMM should return high values (more than one) for pairs of datasets. The following results show that KLMM is an invalid measurement since KLMM does not return high values; even though the cluster representatives are all sine waves. Figure 2.13 and Figure 2.14 show KLMM of STSC using  $k$ -means clustering and KLMM of STSC using  $k$ -hierarchical clustering by varying the number of clusters

( $k$ ) and the length of sliding window ( $w$ ). From the figures, all pair comparisons of eight datasets are evaluated. The value of KLMM is represented in gray shade, where black color represents a high value of KLMM, while white color representing a low value of KLMM. From the experiments, some values are completely white, and some are gray, but not all black; however, the values are expected to be all black since STSC have been proven that it produces meaningless results.

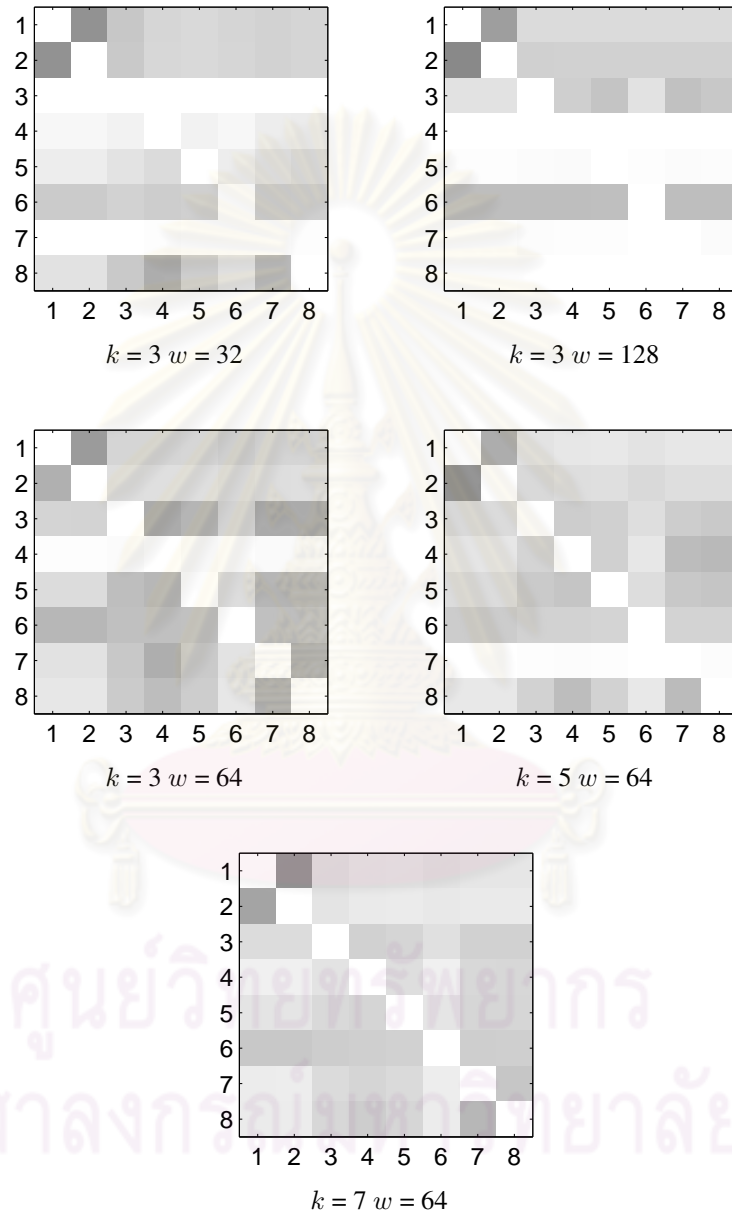


Figure 2.13: KLMMs of STSC using  $k$ -means clustering.

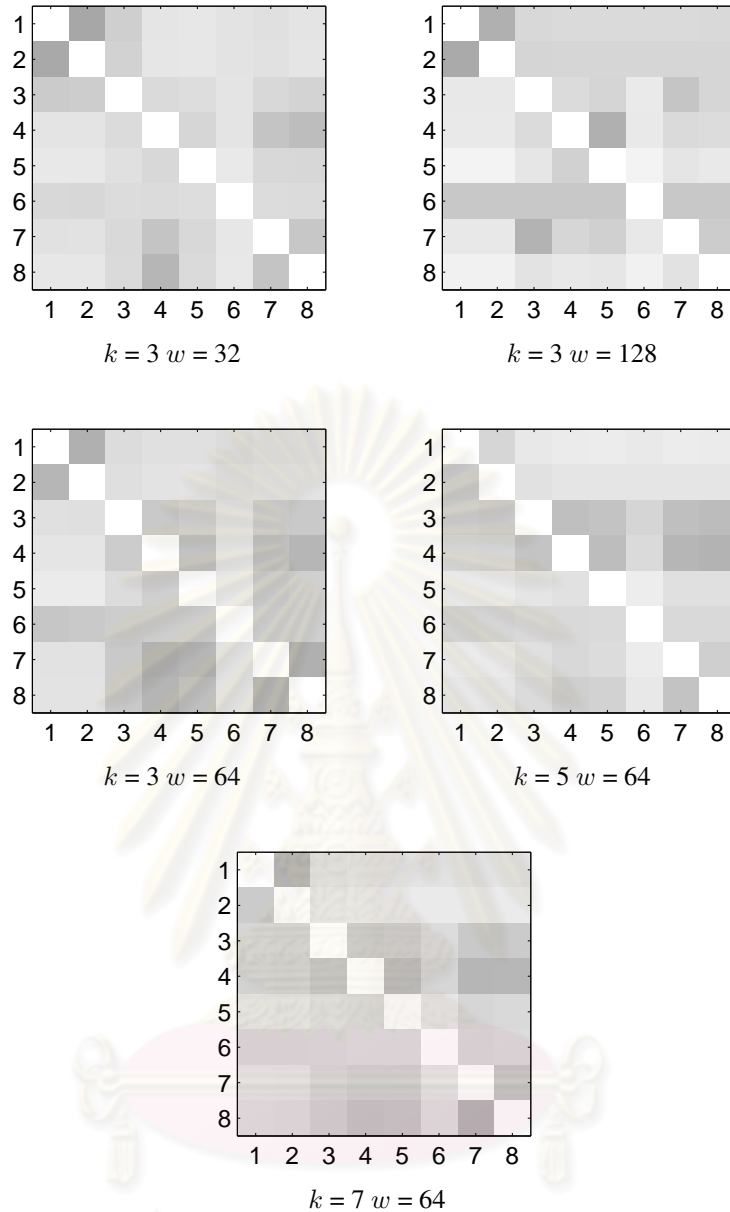


Figure 2.14: KLMMs of STSC using  $k$ -hierarchical clustering.

## 2.4 Causes of Meaninglessness

The causes of meaninglessness are inappropriate approaches to handle trivial-matched subsequences. Trivial-matched subsequences are a set of adjacent subsequences in a time series sequence, where between two adjacent subsequences, only two data points are different. Formally, given a time series sequence  $S = \langle s_1, s_2, \dots, s_n \rangle$  of length  $n$ , a set  $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_{n-w+1}\}$  of subsequences extracted from a sequence  $S$  with a fixed-length sliding window of length  $w$ , a set of trivial-matched subsequences are  $\mathbb{T} = \{\mathcal{S}_i, \mathcal{S}_{i+1}, \dots\}$ , where  $1 \leq i \leq n - w + 1$ . Trivial-matched subsequences of CBF sequence are illustrated in Figure 2.15. In addition, inappropriate

uses of a distance measure and an averaging function to handle trivial-matched subsequences lead to an undesired clustering output. Specifically, STSC utilizes Euclidean distance and Amplitude Averaging function as a distance measure and an averaging function, respectively.

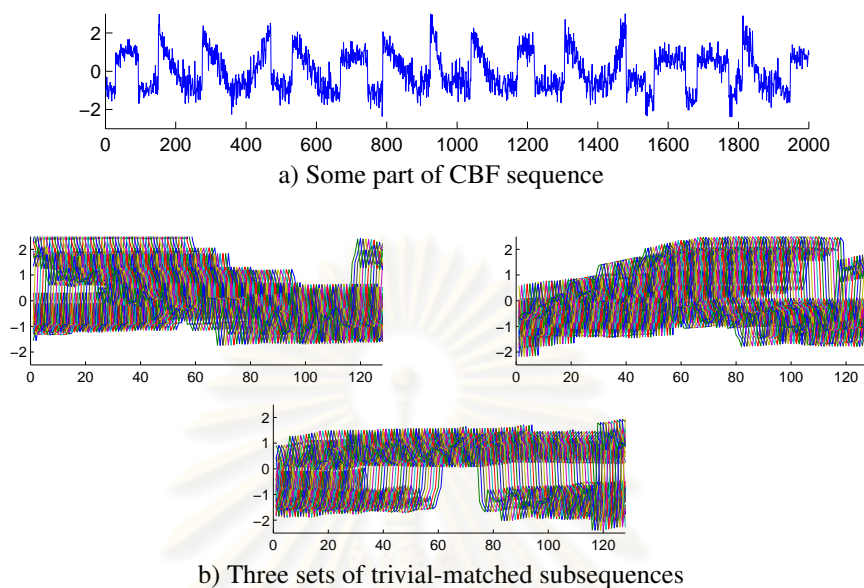


Figure 2.15: Trivial-matched subsequences of CBF sequence

In Euclidean space, two adjacent subsequences may be considered as significantly different although only two data points are different, and the remaining points are the same. To be more illustrative, Euclidean distance cannot group different sets of trivial-matched subsequences shown as a dendrogram in Figure 2.16.

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

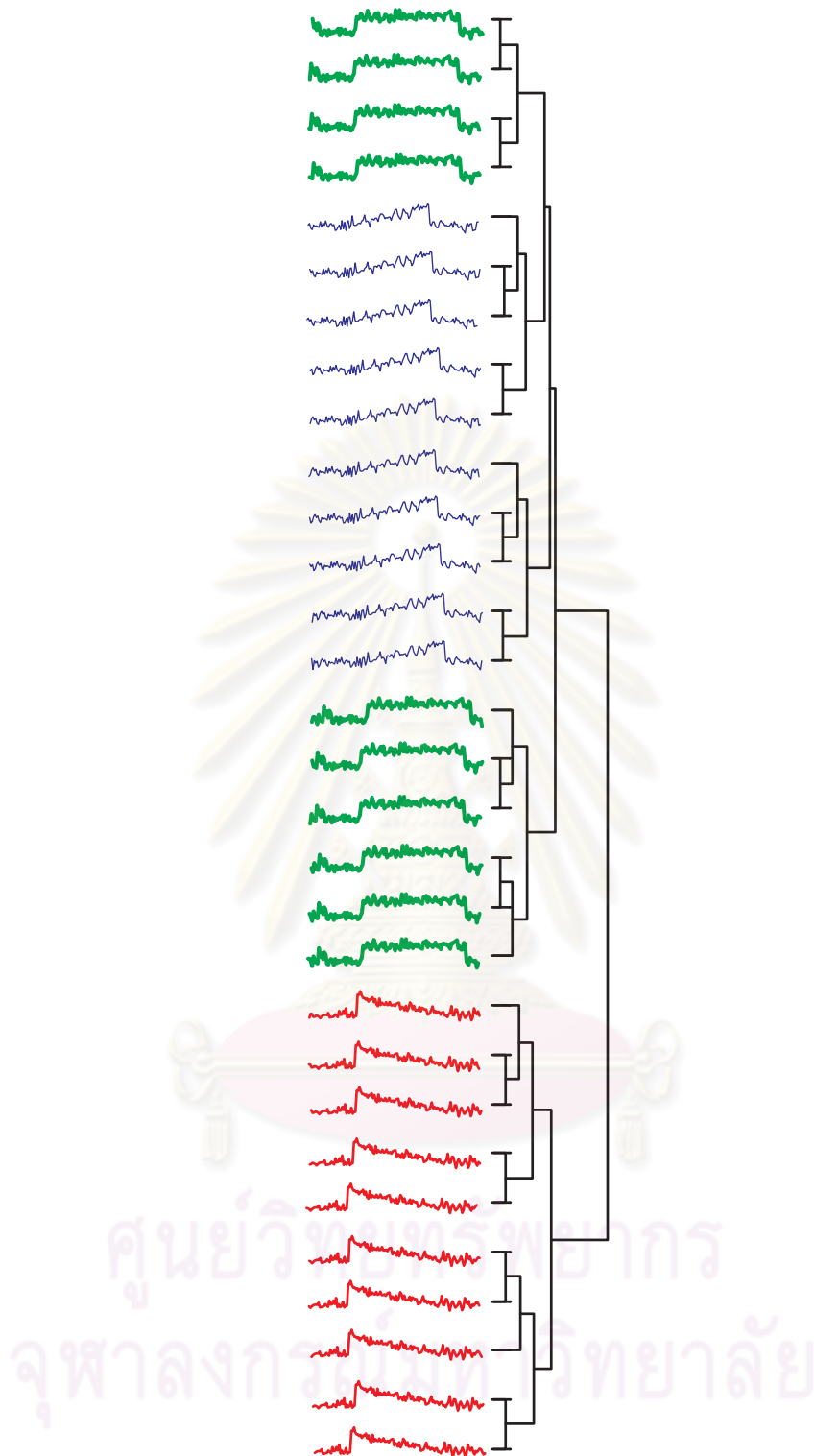


Figure 2.16: Euclidean distance cannot capture similarity between trivial-matched subsequences

To construct a cluster representative, STSC averages all subsequences within a cluster using Amplitude Averaging function, where Amplitude Averaging generates an averaged result by computing a mean of each dimension directly. In addition, Amplitude Averaging is inappropriate to be used as an averaging function of STSC since Amplitude Averaging does not align shifted data

points of adjacent subsequences. Therefore, in the end, each dimension of the result is averaged from unrelated dimensions. This leads to undesired smoothed cluster representatives. Three averaged results of trivial-matched subsequences from CBF sequence generated by Amplitude Averaging function are shown in Figure 2.17. The averaged result will be smoother and more convergent to sine waves; therefore, trivial-matched subsequence clustering can be meaningful when appropriate distance measure and average function are used instead of Euclidean distance and Amplitude Averaging function.

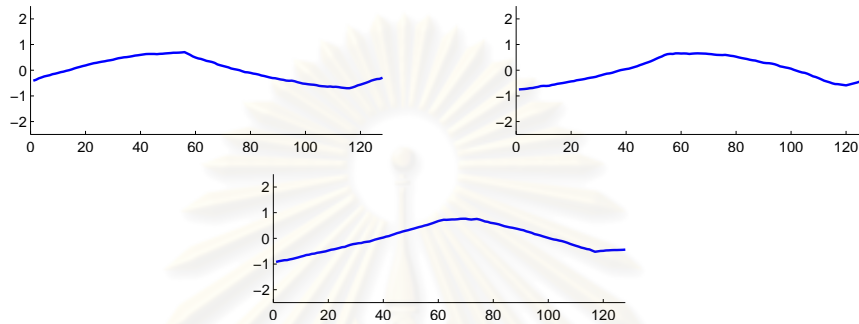


Figure 2.17: Amplitude Averaging produces a smoothed averaged result.

## 2.5 Conclusion

Subsequence Time Series Clustering (STSC) with both  $k$ -means and  $k$ -hierarchical clustering algorithms produces sine waves as cluster representatives regardless of an input sequence. To measure meaningfulness, Keogh and Lin have proposed a meaningfulness measurement, called KLMM, which is shown to be invalid because it returns that the result is meaningful even though cluster representatives are sine waves. The causes of meaningfulness are identified as twofold, i.e., an inappropriate distance measure and an inappropriate averaging function, where STSC utilizes Euclidean distance and Amplitude Averaging function as a distance measure and an averaging function. Therefore, the use of appropriate a distance measure and an averaging function can return a meaningful result.

## CHAPTER III

### SHAPE-BASED AVERAGING

Since the causes of having cluster representatives, the outputs generated from a Subsequence Time Series Clustering (STSC) with both  $k$ -means clustering and  $k$ -hierarchical clustering, becoming all sine waves are inappropriate uses of Euclidean distance and Amplitude Averaging as a distance measure and an averaging function, respectively, in this chapter, Shape-based Averaging is proposed to use instead of Amplitude Averaging in STSC to correctly generate a cluster representative from trivial-matched subsequences. Unlike other typical data types, time series data need Shape-based Averaging instead of Amplitude Averaging since correlations among adjacent dimensions exist (Niennattrakul and Ratanamahatana, 2007a,b). Additionally, Amplitude Averaging produces an undesired mean, where this leads to an inaccurate cluster representative. Figure 3.1 shows the results from averaging of two time series sequences  $A$  and  $B$  using Amplitude Averaging and Shape-based Averaging, respectively. The sequence generated from the Amplitude Averaging shows an undesired averaged result that contains two events, where both original sequences  $A$  and  $B$  consist of only one event. The sequence generated from Shape-based averaging preserves characteristics of these two data sequences that only one event exists.

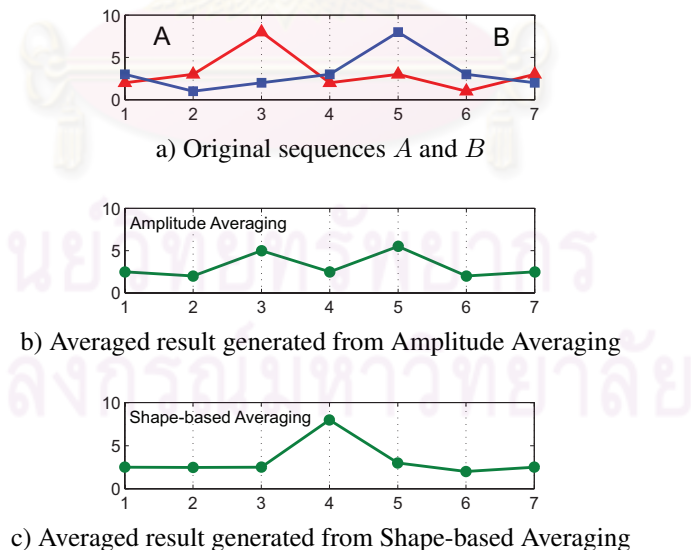


Figure 3.1: Comparison between two averaged results generated from Amplitude Averaging and Shape-based Averaging.

However, constructing an accurate shape-based mean is still controversial because data sequences are averaged in Dynamic Time Warping (DTW) distance space not in the Euclidean space. Unfortunately, no optimal solution has been proposed. Over a decade ago, Gupta et al.

proposed a heuristic solution called NLAFF (Gupta et al., 1996), while only a handful number of work has been adapted to time series data mining domain (Ratanamahatana and Keogh, 2005a; Salvador and Chan, 2007). Particularly, NLAFF does not produce good averaged results since an averaged result is always longer than the original sequence and has large errors. In this thesis, a new averaging scheme with two averaging functions, Cubic-Spline Dynamic Time Warping (CDTW) averaging and Iterative Cubic-Spline Dynamic Time Warping (ICDTW) averaging, is introduced. With the proposed construction algorithm, a very well-formed mean is generated. Averaged results generated from NLAFF, CDTW, and ICDTW are compared and evaluated in terms of SUMDIST, a summation of distances between the averaged result and all original sequences.

### 3.1 Background

This section provides essential background knowledge, i.e., Dynamic Time Warping (DTW) distance and Dynamic Time Warping (DTW) averaging function, to understand proposed methods in this chapter.

#### 3.1.1 Dynamic Time Warping (DTW) Distance

DTW distance (Berndt and Clifford, 1994; Ratanamahatana and Keogh, 2005b) is a well-known shape-based similarity measure that uses a dynamic programming technique to find an optimal warping path between two time series sequences. To calculate the distance, it first creates a distance matrix, where each element in the matrix is a cumulative distance of the minimum value of three surrounding neighbors. Given two time series sequences  $A = \langle a_1, a_2, \dots, a_i, \dots, a_n \rangle$  and  $B = \langle b_1, b_2, \dots, b_j, \dots, b_m \rangle$ , an  $n$ -by- $m$  matrix is first created, and then each  $(i, j)$  element  $\gamma_{i,j}$  of the matrix is defined as:

$$\gamma_{i,j} = |a_i - b_j|^p + \min \{ \gamma_{i-1,j-1}, \gamma_{i-1,j}, \gamma_{i,j-1} \} \quad (3.1)$$

where  $\gamma_{i,j}$  is the summation of  $|a_i - b_j|^p$  and the minimum cumulative distance of three elements surrounding the  $(i, j)$  element, and  $p$  is the dimension of  $L_p$ -norms. When all elements in the matrix are filled, DTW distance is determined from the last element  $\gamma_{n,m}$  of the matrix. For time series domain,  $p = 2$ , equipping to Euclidean distance, is typically used. Since DTW distance is important background knowledge for this thesis, a pseudo code is provided in Table 3.1 and an illustrative example of DTW distance calculation is shown in Figure 3.2.



Table 3.1: Pseudo code of Dynamic Time Warping distance measure

FUNCTION $[dist] = \text{DTW-DISTANCE} [A, B]$	
1.	Let $n$ be the length of time series $A$
2.	Let $m$ be the length of time series $B$
3.	Let $p$ be the dimension of $L_p$ -norms
4.	Initialize $D = \text{ARRAY}[n][m]$
5.	For ( $i = 1$ to $n$ )
6.	For ( $j = 1$ to $m$ )
7.	If ( $i = 1$ and $j \neq 1$ )
8.	$min = D_{i,j-1}$
9.	Else if ( $i \neq 1$ and $j = 1$ )
10.	$min = D_{i-1,j}$
11.	Else
12.	$min = \text{MIN}(D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1})$
13.	End if
14.	$D_{i,j} = min +  a_i - b_j ^p$
15.	End for
16.	End for
17.	Return $dist = \sqrt[p]{D_{n,m}}$

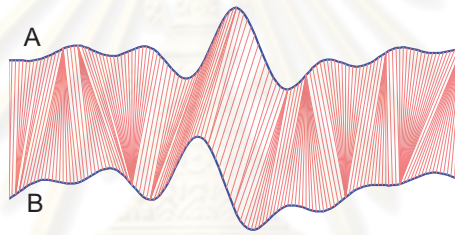


Figure 3.2: Alignment obtained from a DTW distance calculation.

### 3.1.2 Dynamic Time Warping (DTW) Averaging

DTW averaging was first introduced by Gupta et al. (Gupta et al., 1996) to find an averaged result between two time series sequences. Unlike DTW distance, DTW averaging uses another matrix to store an index of the minimum distance among adjacent elements. The path matrix is created to store an index of the adjacent element that has minimum cumulative distance, and a path is traced back from the last element to the first element. An averaged result is then calculated along the path. Suppose the path  $W = \langle w_1, w_2, \dots, w_k, \dots, w_N \rangle$  of length  $N$ , where  $w_k$  is  $k^{\text{th}}$  coordinate  $(i_k, j_k)$  in the optimal path of sequences  $A = \langle a_1, a_2, \dots, a_i, \dots, a_n \rangle$  and  $B = \langle b_1, b_2, \dots, b_j, \dots, b_m \rangle$ , where  $i_k$  and  $j_k$  are indices of data points in sequences  $A$  and  $B$ , respectively. Therefore, a new sequence  $Z = \langle z_1, z_2, \dots, z_k, \dots, z_N \rangle$  is derived from elements  $z_k = \frac{a_{i_k} \cdot \omega_A + b_{j_k} \cdot \omega_B}{\omega_A + \omega_B}$ , where  $\omega_A$  and  $\omega_B$  are the weights of sequences  $A$  and  $B$ , respectively. We also provide a concrete pseudo code of DTW averaging in Table 3.2. For example in Figure 3.3, two sequences  $A = \langle 2, 3, 8, 2, 3, 1, 3 \rangle$  and  $B = \langle 3, 1, 2, 3, 8, 3, 2 \rangle$  are averaged by DTW averaging algorithm to produce an averaged result  $Z = \langle 2.5, 1.5, 2, 3, 8, 2.5, 3, 1.5, 2.5 \rangle$ .

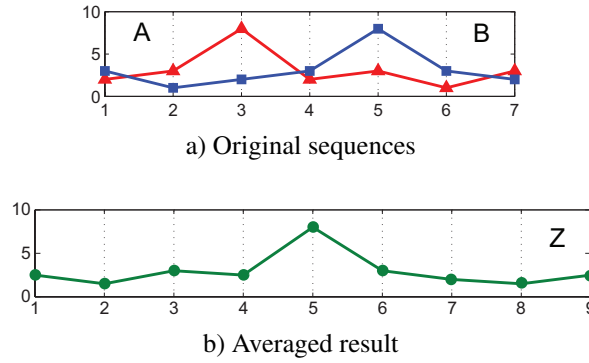


Figure 3.3: Result generated from DTW Averaging

Table 3.2: Pseudo code of Dynamic Time Warping averaging function

FUNCTION $[W] = \text{DTW-AVERAGING} [A, B, \omega_A, \omega_B]$
1. $W = \text{WARPINGPATH}(A, B)$
2. Let $N$ be a length of the path $W$
3. Let $Z$ be a time series sequence of length $N$
4. For ( $k = 1$ to $N$ )
5. $[i, j] = w_k$
6. $z_k = \frac{a_i \cdot \omega_A + b_j \cdot \omega_B}{\omega_A + \omega_B}$
7. Add $z_k$ to $Z$
8. End for
9. Return $Z$

It is important to note that DTW averaging function is an operation which has only commutative property with no associative property (Niennattrakul and Ratanamahatana, 2007a). In the other words, if there are three sequences  $A$ ,  $B$ , and  $C$ , a result of averaging  $A$  and  $B$ , then  $C$  is not necessarily equal to a result of averaging  $B$  and  $C$ , then  $A$ . A sequence ordering can largely affect the averaged result. In addition, an averaging sequence will always be longer or equal to the original sequences. If a large dataset is to be averaged, averaging sequences will be very long which will definitely decrease a system performance. Therefore, in this chapter, two new shape-based averaging functions to resolve this problem and a new averaging scheme to efficiently order averaging sequences are proposed.

### 3.2 Related Work

Over a decade ago, Gupta et al. proposed a heuristic shape-averaging scheme called NLAFF (Gupta et al., 1996), which was first introduced in signal processing community, and later has been utilized in data mining tasks (Ratanamahatana and Keogh, 2005a; Salvador and Chan, 2007). Specifically, NLAFF uses DTW averaging to produce a mean between a pair of time series sequences. NLAFF consists of two averaging schemes, i.e., NLAFF<sub>1</sub> and NLAFF<sub>2</sub>. NLAFF<sub>1</sub> averages sequences in hierarchical manner. Suppose there are eight sequences, i.e.,  $A_1$

Table 3.3: Pseudo code of generating a warping path

FUNCTION [W] = WARPINGPATH [A, B]	
1.	Initialize distance matrix $DM$ and path matrix $PM$
2.	For each $a_i$ in $A$ and $b_j$ in $B$
3.	$DM[i, j] =  a_i - b_j ^p$
4.	If ( $i = 1$ and $j \neq 1$ )
5.	$DM[i, j] += DM[i, j - 1]$
6.	$PM[i, j] = 1$
7.	Else if ( $i \neq 1$ and $j = 1$ )
8.	$DM[i, j] += DM[i - 1, j]$
9.	$PM[i, j] = 2$
10.	Else if ( $i \neq 1$ and $j \neq 1$ )
11.	$dist = \text{MIN}(DM[i, j - 1], DM[i - 1, j], DM[i - 1, j - 1])$
12.	If ( $dist = DM[i, j - 1]$ )
13.	$PM[i, j] = 1$
14.	Else if ( $dist = DM[i - 1, j]$ )
15.	$PM[i, j] = 2$
16.	Else
17.	$PM[i, j] = 3$
18.	End if
19.	$DM[i, j] += dist$
20.	Else
21.	$PM[i, j] = 3$
22.	Endif
23.	Endfor
24.	Let $n$ be a length of the sequence $X$
25.	Let $m$ be a length of the sequence $Y$
26.	While ( $n \neq 0$ and $m \neq 0$ )
27.	$w_k = [n, m]$
28.	If ( $PM[n, m] = 1$ )
29.	$m = m - 1$
30.	Else if ( $PM[n, m] = 2$ )
31.	$n = n - 1$
32.	Else
33.	$m = m - 1; n = n - 1$
34.	End if
35.	End while
36.	$W = \text{Reverse order of } W$
37.	Return $W$

to  $A_8$ .  $A_1$  and  $A_2$  are averaged to produce  $A_{1,2}$ , and  $A_3$  and  $A_4$  are averaged to produce  $A_{3,4}$ , and so on. Then, in the next level,  $A_{1,2}$  and  $A_{3,4}$  are averaged to produce  $A_{(1,2),(3,4)}$ , and so on. Limitation of  $NLAAF_1$  is that it requires that the number of sequences must be a power of two. Unlike  $NLAAF_1$ ,  $NLAAF_2$  averages sequences in sequential manner.  $A_1$  and  $A_2$  are first averaged to produce  $A_{1,2}$ , and then  $A_{1,2}$  and  $A_3$  are averaged to produce  $A_{(1,2),3}$ , and so on.

Since  $NLAAF_1$  has limitation that it requires the number of sequences to be a power of two, Gupta et al. recommend to use combination of both  $NLAAF_1$  and  $NLAAF_2$ . For example, to average 100 sequences, 4 sequences will be discarded, and the rest of the sequences will be separated into three groups of 32 sequences, each of which will be averaged using  $NLAAF_1$ . Therefore,

three averaged sequences produced from  $NLAAF_1$  will then be averaged using  $NLAAF_2$ . Since DTW averaging function does not have an associative property, different orderings of sequences in both  $NLAAF_1$  and  $NLAAF_2$  will lead to different averaged results. Additionally, an averaged sequence from  $NLAAF$  will be very long since DTW averaging function will always produce a longer or equal sequence to its original sequences. In this chapter, two new DTW averaging functions and an averaging scheme which produce a more accurate averaged result are proposed, and when this result is used in subsequence clustering, it produces more meaningful clustering results.

### 3.3 Shape-based Averaging

To average a set of sequences, an averaging scheme to construct an averaged result is proposed since the shape-based averaging does not have an associative property (Niennattrakul and Ratanamahatana, 2007a). Instead of averaging sequences in a random order as done in  $NLAAF$ , a heuristic solution is introduced to return a good averaged result by averaging a pair of sequences which are the most similar first. After the averaged result is generated, a pair of sequences from the remaining data including the previous averaged result is determined for the next iteration. The scheme keeps going until only one sequence remains. A pseudo code of the averaging scheme is provided in Table 3.4.

Table 3.4: Pseudo code of Shape-based Averaging scheme

FUNCTION [W] = AVERAGINGScheme [S]	
1.	Initialize a weight $\omega = 1$ for each sequence $S$ in $\mathbb{S}$
2.	While (SIZE( $\mathbb{S}$ ) > 1)
3.	$[A, B]$ = Most similar sequences in $\mathbb{S}$
4.	$Z$ = AVERAGINGFUNCTION( $A, B, \omega_A, \omega_B$ )
5.	Remove $A$ and $B$ from $\mathbb{S}$
6.	$\omega_Z = \omega_A + \omega_B$
7.	Add $Z$ to $\mathbb{S}$
8.	End while
9.	Return $Z$

In this chapter, two novel averaging functions, i.e., Cubic-Spline Dynamic Time Warping (CDTW) and Iterative Cubic-Spline Dynamic Time Warping (ICDTW) are introduced. Either one of these two averaging functions can be used as the AVERAGINGFUNCTION in Line 4 of Table 3.4.

#### 3.3.1 Cubic-Spline Dynamic Time Warping (CDTW) Averaging

CDTW averaging function produces a more accurate averaged result by considering both position and amplitude of each data point of a new averaged sequence, while DTW averaging function (Table 3.1) considers only amplitude. In other words, DTW averaging function equally

treats every new data point in a new sequence, while CDTW averaging function additionally determines where a new data point should be placed. Specifically, a position and an amplitude of a data point in the sequence can be observed as  $x$ - and  $y$ - coordinate in time series. Therefore, the sequence generated from CDTW function is more useful since it preserves both position and amplitude from the warping path. Figure 3.4 shows the comparison between averaged results generated from CDTW and DTW averaging functions, where two inputs are  $A = \langle 2, 3, 8, 2, 3, 1, 3 \rangle$  and  $B = \langle 3, 1, 2, 3, 8, 3, 2 \rangle$ .

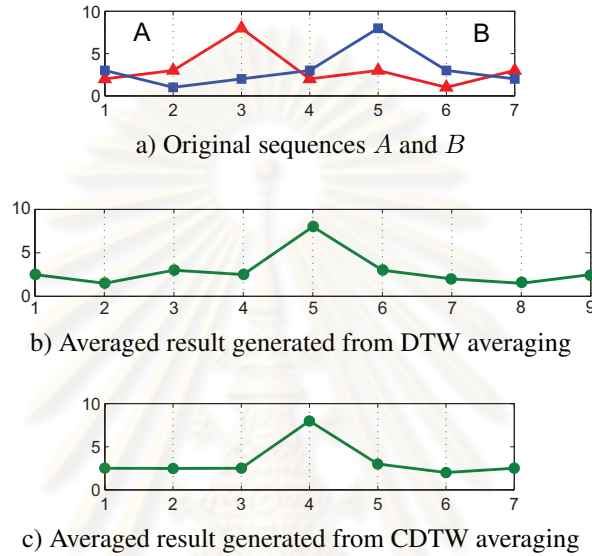


Figure 3.4: Comparison between DTW averaging and CDTW averaging functions

Suppose the path  $W = \langle w_1, w_2, \dots, w_k, \dots, w_N \rangle$ , where  $w_k = (i_k, j_k)$  is  $k^{th}$  coordinate in the optimal path of sequences  $A$  and  $B$ . Therefore, a position  $z'_{k_x}$  of a data point in a new sequence  $Z'$  is determined by  $z'_{k_x} = \frac{\omega_A \cdot i_k + \omega_B \cdot j_k}{\omega_A + \omega_B}$ , and an amplitude  $z'_{k_y}$  of a data point in a new sequence  $Z'$  is determined by  $z'_{k_y} = \frac{\omega_A \cdot a_{i_k} + \omega_B \cdot b_{j_k}}{\omega_A + \omega_B}$ , where  $\omega_A$  and  $\omega_B$  are the weights of sequences  $A$  and  $B$ , respectively.

However, the length of the sequence  $Z'$  is always equal to or longer than two original sequences; therefore, re-sampling is required. In this thesis, CDTW averaging function uses a cubic-spline interpolation (Burden et al., 1997) since it requires no parameter and outperforms other interpolation techniques in re-sampling of natural sequences. Additionally, CDTW function re-samples positions of averaged result to integer values. As illustrated in Figure 3.5, the sequence  $Z'$  of 9 data points is re-sampled to the sequence  $Z$  of 7 data points. A concrete pseudo code of CDTW function is provided in Table 3.5.

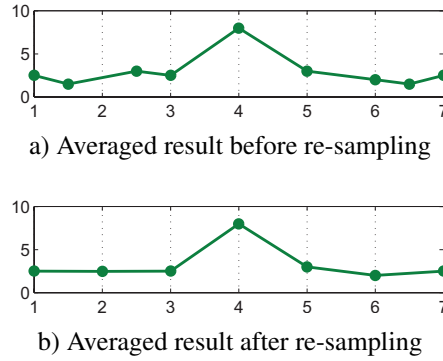


Figure 3.5: Averaged results before and after re-sampling in CDTW averaging function.

Table 3.5: Pseudo code of Cubic-Spline Dynamic Time Warping (CDTW) averaging function

FUNCTION $[Z] = \text{CDTW-AVERAGING}[A, B, \omega_A, \omega_B]$
1. $W = \text{WARPINGPATH}(A, B)$
2. Let $N'$ be the length of the path $W$
3. Let $N$ be the equal length of time series $A$ and $B$
4. Let $Z$ be a time series sequence of size $N$
5. Let $Z'$ be a time series sequence of size $N'$
6. For ( $k = 1$ to $N'$ )
7. $[i, j] = w_k$
8. $x = \frac{i \cdot \omega_A + j \cdot \omega_B}{\omega_A + \omega_B}$
9. $y = \frac{a_i \cdot \omega_A + b_j \cdot \omega_B}{\omega_A + \omega_B}$
10. Add $[x, y]$ to $Z'$
11. End for
12. $Z = \text{CUBICSPLINE}(Z')$
13. Return $Z$

### 3.3.2 Iterative Cubic-Spline Dynamic Time Warping (ICDTW) Averaging

Although CDTW function produces a good averaged result since it considers both position and amplitude, another essential but not necessary condition for averaging is that the averaged result should be in the middle of two original sequences. In other words, DTW distances between the sequences and the result should be equal. Therefore, an iterative approach for CDTW averaging function called Iterative Cubic-Spline Dynamic Time Warping (ICDTW) averaging function is proposed. ICDTW function can truly represent characteristics of a set of subsequences.

It is important to emphasize that the distances between the generated result from CDTW function and two original time series sequences are *not* always equal; therefore, the averaged result needs to be slightly adjusted. Obviously, since all elements in the sequence are real numbers, it is very difficult to obtain the sequence that satisfies this condition; therefore, a heuristic and deterministic solution is proposed, i.e., ICDTW averaging function mentioned above. To average two time series sequences  $A$  and  $B$ , ICDTW function will find new weights  $\beta_A$  and  $\beta_B$  which make the averaged result  $Z$  be the center between the sequences  $A$  and  $B$ . Obviously, finding both

weights  $\beta_A$  and  $\beta_B$  is not very practical since the weights  $\beta_A$  and  $\beta_B$  are real numbers. A binary search is used instead to find only the weight  $\beta_A$ , when the weight  $\beta_B$  is fixed. Specifically, for each iteration, a new weight  $\beta_A$  is checked whether or not the generated averaged result  $Z$  has an equal DTW distances to the sequences  $A$  and  $B$ . If the distances are equal, ICDTW terminates. In other words, only weight  $\beta_A$  is necessary to search, while weight  $\beta_B$  can be fixed as a constant because two sets of weights are equivalent. For example, for  $\{\beta_A, \beta_B\} = \{4, 5\}$ , it can be reduced to  $\{0.8, 1\}$  when the weight  $\beta_B$  is fixed to 1; therefore, searching for  $\beta_A$  is enough to find any pair of weights  $\{\beta_A, \beta_B\}$ . Pseudo code of ICDTW averaging function is provided in Table 3.6. Note that two initial weights of  $A$ ,  $\beta_{A_1}$  and  $\beta_{A_2}$ , are set to be  $10^{-5}$  and  $10^5$ . These numbers can be initialized to any numbers, where  $\beta_{A_1}$  must be much smaller than  $\beta_{A_2}$ , so the algorithm can be converged.

Table 3.6: Pseudo code of Iterative Cubic-Spline Dynamic Time Warping (ICDTW) averaging function

FUNCTION $[Z] = \text{ICDTW-AVERAGING}[A, B, \omega_A, \omega_B]$	
1.	Initialize weights $\beta_{A_1} = 10^{-5}$ , $\beta_{A_2} = 10^5$ , and $\beta_B = 1$
2.	Initialize weight $\beta_{A_3} = \frac{(\beta_{A_1} + \beta_{A_2})}{2}$
3.	$Z = \text{CDTW-AVERAGING}(A, B, \beta_{A_3}, \beta_B)$
4.	$d_{Z,A} = \text{DTWDISTANCE}(Z, A) \cdot \omega_A$
5.	$d_{Z,B} = \text{DTWDISTANCE}(Z, B) \cdot \omega_B$
6.	$\beta_{A_3} = d_{Z,A} < d_{Z,B} ? \beta_{A_1} : \beta_{A_2}$
7.	While $( d_{Z,A} - d_{Z,B}  > 0)$
8.	$\beta_{A_3} = \frac{(\beta_{A_1} + \beta_{A_2})}{2}$
9.	$Z = \text{CDTW-AVERAGING}(A, B, \beta_{A_3}, \beta_B)$
10.	$d_{Z,A} = \text{DTWDISTANCE}(Z, A) \cdot \omega_A$
11.	$d_{Z,B} = \text{DTWDISTANCE}(Z, B) \cdot \omega_B$
12.	If $(d_{Z,A} < d_{Z,B})$
13.	$\beta_{A_2} = \beta_{A_3}$
14.	Else
15.	$\beta_{A_1} = \beta_{A_3}$
16.	End if
17.	End while
18.	Return $Z$

Note that both CDTW and ICDTW averaging functions can be used in subsequence clustering. However, to preserve characteristics of an averaged result, ICDTW function is more preferred. For CDTW function, the averaged result preserves shape-based averaging process which considers both position and amplitude of the warping alignment, while ICDTW averaging returns more accurate characteristics of the averaged result by calibrating the resulted sequence having the same distance between the result and original sequences. Performance of CDTW and ICDTW functions will be demonstrated in the experiment evaluation.

### 3.4 Experimental Evaluation

The following experiment will demonstrate the superiority of the proposed averaging functions over the current existing approaches, where the accuracies of the proposed shape-based averaging method, i.e., a new averaging scheme with two proposed CDTW and ICDTW algorithms, comparing with those of NLAAF, are reported. Our proposed methods are evaluated with 20 datasets from the UCR classification/clustering page (Keogh et al., 2011). Table A.1 shows the number of classes, the length of each time series sequence, and the size of the datasets, and Figure A.2 shows some examples of each dataset. Figure 3.6 shows examples of some classes used in this evaluation.

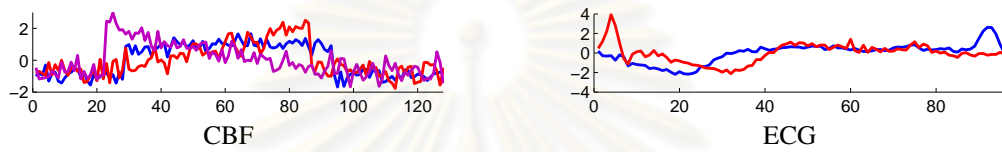


Figure 3.6: Examples of some classes in evaluated datasets.

For each dataset, training data and test data are all combined, and then all sequences are averaged. Note that sequences are averaged within their own classes to achieve maximum utilities. The averaged results are evaluated using SUMDIST function, defined as a summation of all distances between the averaged result and each of the original sequences in the dataset. If a value from SUMDIST is small, it means that this method generates a good averaged result. SUMDIST function is provided as follows.

$$\text{SumDist}(\hat{\mathcal{S}}, \mathbb{S}) = \sum_{i=1}^{|\mathbb{S}|} \text{DTWDistance}(\hat{\mathcal{S}}, \mathcal{S}_i) \quad (3.2)$$

where  $\mathbb{S}$  is a dataset,  $\hat{\mathcal{S}}$  is the averaged result, and  $\mathcal{S}_i$  is each data sequence in the dataset  $\mathbb{S}$ .

Table 3.7 shows the SUMDIST comparison between NLAAF and our proposed methods, CDTW and ICDTW functions, where SUMDIST reported in Table 3.7 is a summation of SUMDISTs of all classes. From the experiment results, it is apparent from the experiment results that CDTW and ICDTW functions achieve lower SUMDIST values since all sequences are averaged using a new averaging schemes, while the scheme of NLAAF averages sequences in random manner, and no resampling method is adopted in NLAAF to scale the averaged sequence to the same length. Averaged results from CDTW, ICDTW, and NLAAF of CBF and ECG are shown in Figures 3.7 and 3.8, respectively, where the results from other datasets are provided in Figures



C.1 to C.3 in Appendix C.

Table 3.7: SUMDIST of each averaging method

Dataset	NLAAF	CDTW	ICDTW
50words	4277.6	<b>2348.2</b>	2360.5
Adiac	353.9	285.9	<b>284.3</b>
Beef	384.7	<b>219.9</b>	222.5
CBF	8730.6	4007.1	<b>3821.0</b>
Coffee	69.5	<b>43.0</b>	43.6
ECG	1160.8	528.4	<b>519.4</b>
Face (all)	18339.0	8748.6	<b>8670.4</b>
Face (four)	945.0	613.4	<b>604.4</b>
Fish	516.9	297.3	<b>284.4</b>
Gun-Point	1375.0	<b>466.0</b>	468.4
Lighting-2	2606.9	1195.5	<b>1183.3</b>
Lighting-7	1142.0	<b>858.4</b>	865.1
Oliveoil	6.8	<b>6.5</b>	<b>6.5</b>
OSULeaf	6309.6	<b>2797.4</b>	2805.9
SwedishLeaf	2510.6	1452.8	<b>1415.1</b>
Synthetic	3472.5	2063.1	<b>2050.5</b>
Trace	469.9	<b>221.0</b>	248.5
TwoPatterns	46392.0	1911.4	<b>1874.2</b>
Wafer	635545.3	53026.4	<b>52723.1</b>
Yoga	117113.2	<b>16924.0</b>	16947.9

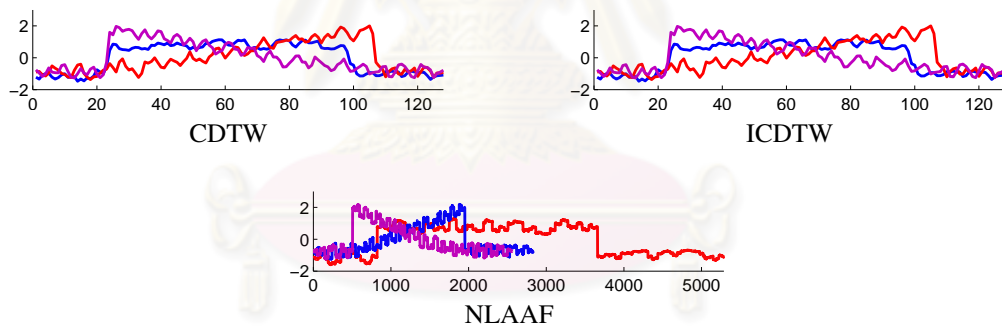


Figure 3.7: Averaged results of CBF

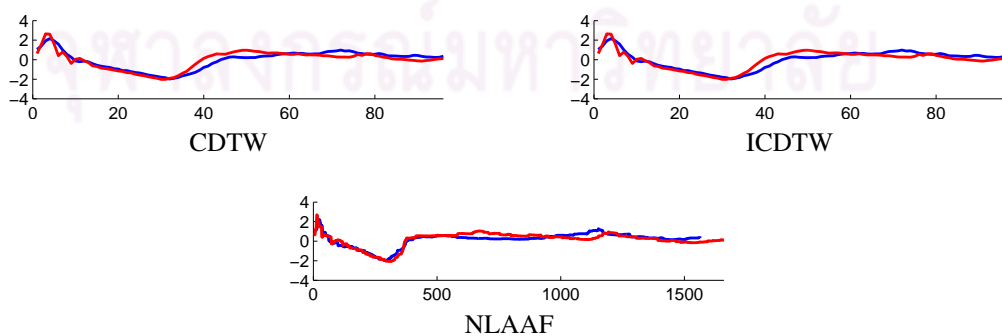


Figure 3.8: Averaged results of ECG

### 3.5 Averaging Trivial-Matched Subsequences

Trivial-matched subsequences are a set of adjacent subsequences whose differences are only a few points. For example, from a CBF dataset, in Figure 3.9, three sets of trivial-matched subsequences are extracted and shown in Figure 3.9. Therefore, Amplitude Averaging function is inappropriate to average these subsequences since Amplitude Averaging function does not align subsequences before averaging. If Amplitude Averaging is used, the averaged result will be smoothed and the output of subsequence clustering will be meaningless. Figure 3.10 shows the averaged results when Amplitude Averaging averages three sets of trivial-matched subsequences. Since CDTW and ICDTW averaging functions align subsequences before averaging, the averaged result preserves all characteristics, as shown in Figure 3.11 and Figure 3.12, respectively. Therefore, CDTW and ICDTW averaging functions are more appropriate to use to construct cluster representatives in subsequence clustering than Amplitude Averaging function. Either CDTW or ICDTW averaging function can be used to generate cluster representatives in subsequence clustering, where according to experiments, ICDTW provides more accurate averaged results.

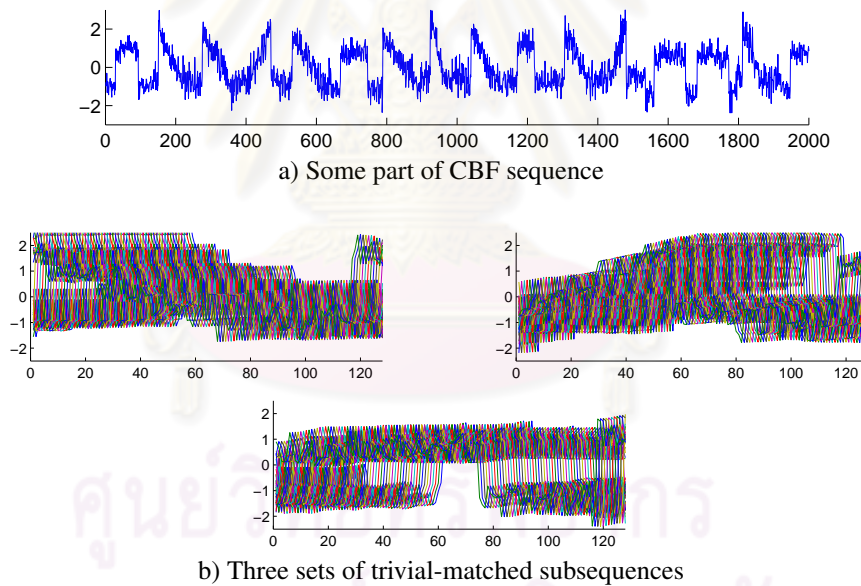


Figure 3.9: Trivial-matched subsequences b) extracted from a) CBF sequence.

### 3.6 Conclusion

This thesis proposes CDTW and ICDTW functions to generate an accurate averaged result. Since time series data have correlation among dimensions, CDTW and ICDTW functions are more appropriate than Amplitude Averaging function. In addition, CDTW and ICDTW functions are shown to outperform NLAFF, and they should be used as an averaging function for subsequence clustering.

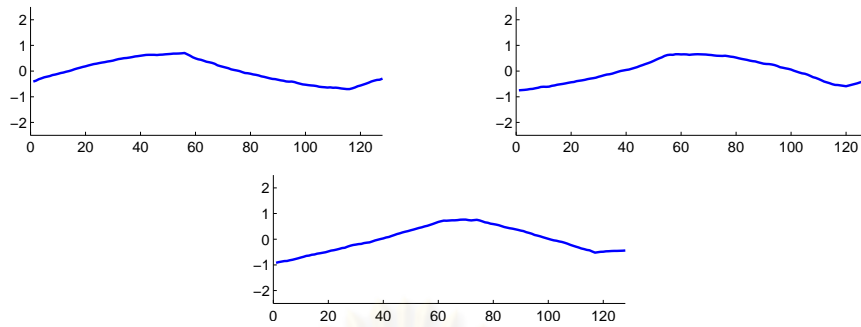


Figure 3.10: Averaged results generated from Amplitude Averaging.

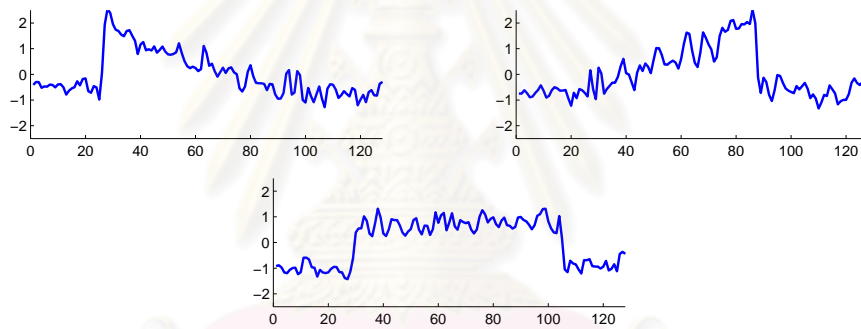


Figure 3.11: Averaged results generated from Shape-based Averaging with CDTW function.

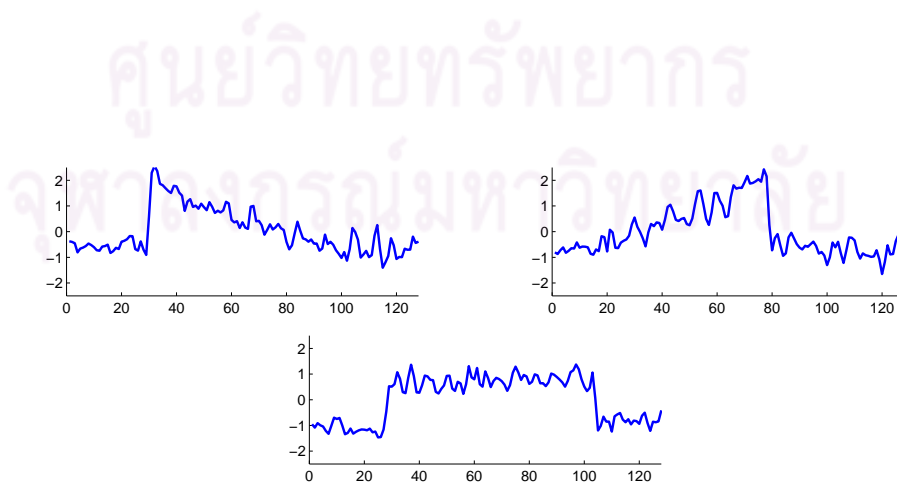


Figure 3.12: Averaged results generated from Shape-based Averaging with ICDTW function.

## CHAPTER IV

# 2STSC: SHAPE-BASED SUBSEQUENCE TIME SERIES CLUSTERING

Since Keogh and Lin proved that the clustering results of Subsequence Time Series Clustering (STSC) are meaningless (Lin et al., 2003; Keogh and Lin, 2005), many other methods, e.g., Density-based Subsequence Time Series Clustering (DSTSC) (Denton, 2005), Lag-based Subsequence Time Series Clustering (LSTSC) (Simon et al., 2006; Chen, 2007a,b), and Phrase-Analysis Subsequence Time Series Clustering (PASTSC) (Fujimaki et al., 2008), have been proposed in order to solve this meaninglessness. These previous works introduce additional parameters to discard or filter out trivial-matched subsequences. For DSTSC, a distance threshold is proposed to eliminate groups of clusters that have distances below this threshold, while LSTSC and PASTSC propose a lag value and a slide length to select only some subsequences from a large set of extracted subsequences. These parameters, however, are very sensitive to a clustering result that if inappropriate parameter values are chosen, the clustering results will still be meaningless. In addition, those works have too strict assumption that the time series sequence must be cyclic, where this assumption scarcely satisfies in real-world data. Obviously, these previous work do not solve at the right point. Firstly, inappropriate parameter values may discard some useful subsequences, and secondly, distance measures used in those clustering algorithms are based on Euclidean distance that cannot capture similarity between two adjacent subsequences of trivial-matched subsequences. Lastly, a cluster representative generated from those clustering algorithms are from typical statistical values such as a mean or a median, where a mean is an averaged result of all cluster members generated from Amplitude Averaging, and a median is selected from an existing data sequence. Although a median can produce a meaningful clustering representative since it is selected from the existing sequence, the median is still not preferred to be used as a cluster representative because the median is usually sensitive to an imbalanced dataset, while the mean, on the other hand, does preserve characteristics of all data objects in the averaging.

In this chapter, Shape-based Subsequence Time Series Clustering (2STSC) is proposed to produce meaningful clustering results. Since trivial-matched subsequences are contiguous subsequences which have shifts in a time domain, an appropriate distance measure and an averaging function, i.e., Dynamic Time Warping (DTW) distance and Shape-based Averaging, are used to find the optimal alignment before distance calculation and averaging. Suppose there are three sets

of trivial-matched subsequences as shown in Figure 4.1, Figure 4.2 demonstrates that Euclidean distance cannot capture the similarity of trivial-matched subsequences by identifying that subsequences from the same set of trivial-matched subsequences are different. Compared to Euclidean distance, DTW distance, on the other hand, can correctly group three sets of trivial-matched subsequences because Euclidean distance calculates a distance in one-to-one manner, while DTW distance finds an optimal alignment before distance calculation. Given the same three sets of subsequences as in Figure 4.3, the Amplitude Averaging produces an averaged result whose shapes are smoothed, while Shape-based Averaging still preserves all characteristics of the sequences, especially the peaks and valleys of the sequences.

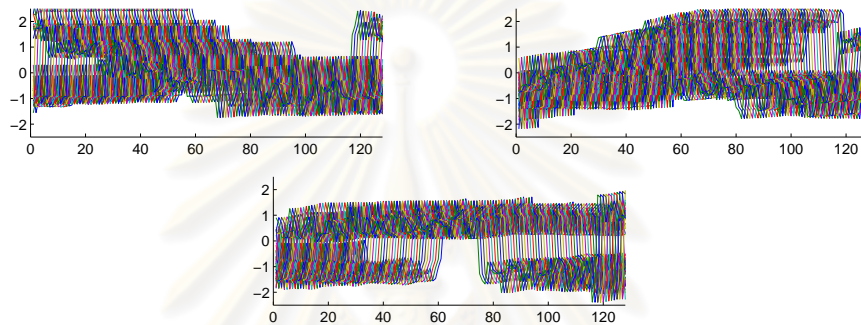


Figure 4.1: Three sets of trivial-matched subsequences.

To be more illustrative, a simple experiment demonstrates that STSC produces meaningless results. The test sequence is generated from concatenation of thirty sequences of three patterns, i.e., Cylinder, Bell, and Funnel, from the CBF dataset. The clustering results of STSC and 2STSC are shown in Figure 4.4, where the number of clusters ( $k$ ) and the length of sliding window ( $w$ ) are set to be 3 and 128, respectively. Clustering results of STSC are all sine waves, while 2STSC returns meaningful patterns. Note that 2STSC does not return three patterns, i.e., Cylinder, Bell, and Funnel, as expected cluster representatives because other patterns including joints between the patterns also do exist in the long time series sequence. With this proposed solution that utilizes DTW distance and Shape-based Averaging as a distance measure and an averaging function, 2STSC will demonstrate that it produces meaningful results in an experimental evaluation section.

#### 4.1 Related Work

In this section, related works are reviewed and described to show that subsequence clustering is challenging and still an open problem. So far, no proposed work has yet efficiently solved the problem. This thesis will be the first work to introduce meaningful subsequence clustering algorithm.

Since Keogh and Lin have reported the shocking finding that the output of STSC was mean-

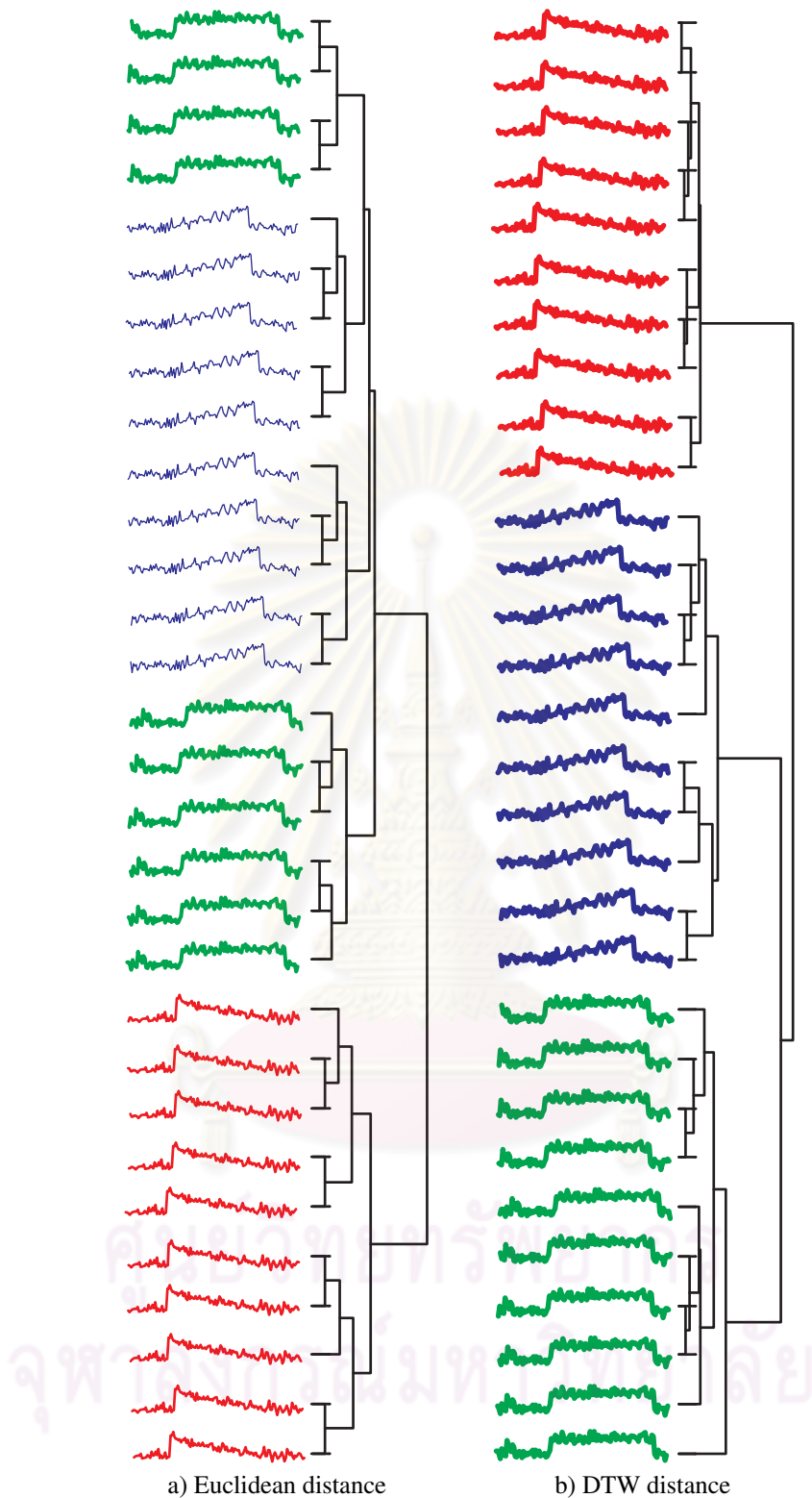


Figure 4.2: a) Euclidean cannot capture the similarity of trivial-matched subsequences, while b) DTW can.

ingless (Lin et al., 2003; Keogh and Lin, 2005), hundreds of works and their successors that use STSC as a subroutine or a preprocessing step are also considered producing meaningless outputs. Keogh and Lin also proposed a tentative solution (Lin et al., 2003; Keogh and Lin, 2005) by using

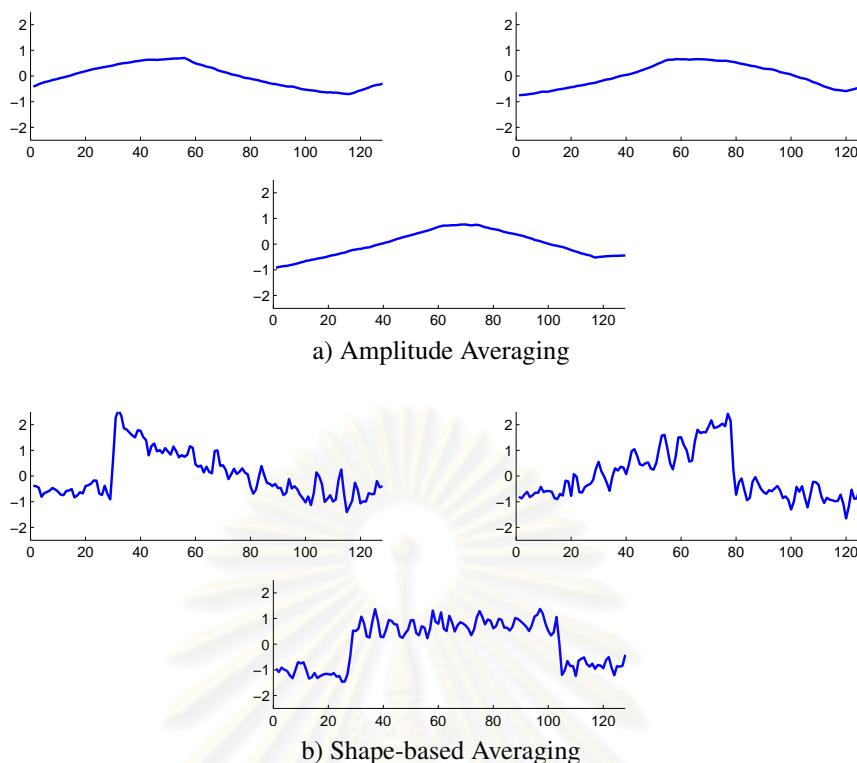


Figure 4.3: a) Amplitude Averaging cannot construct meaningful representatives of trivial-matched subsequences, while b) Shape-based Averaging can.

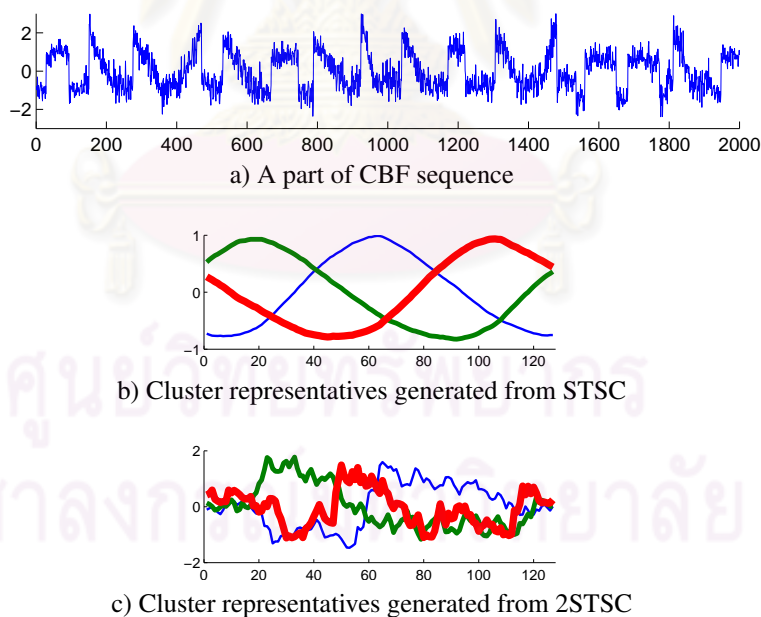


Figure 4.4: a) STSC produces a meaningless clustering result, while b) 2STSC produces a meaningful clustering result.

motif discovery (Mueen et al., 2009) to remove trivial-matched subsequences, and the remaining subsequences are then clustered using  $k$ -hierarchical clustering and  $k$ -means clustering. However, the motif discovery is parameter-laden in that a real-value distance threshold must be specified in advance to define which sequences are motifs or trivial matches, and using any preprocessing

steps to filter out these trivial-matched sequences may lead to an error because some important or desired subsequences are discarded.

Density-based Subsequence Time Series Clustering (DSTSC) (Denton, 2005) has then been proposed by using a kernel function to model trivial-matched subsequences as noises, and a distance threshold has been used to discover the clusters and eliminate noises. Nevertheless, the distance threshold has to be manually defined by users, and its cluster representative is selected from a median of cluster members. However, a median is an undesired cluster representative because a median is affected from imbalance distribution of cluster members that all cluster members should be averaged instead of just selecting one existing sequence. Therefore, a mean is more appropriate than other statistical values, i.e., a mode or a median, since a mean can better reflect characteristics of an interesting data collection by averaging all data sequences.

Lag-based Subsequence Time Series Clustering (LSTSC) (Simon et al., 2006; Chen, 2007b,a) is a subsequence clustering algorithm that re-samples subsequences to a specific lag value using a new distance measure (Chen, 2007b), and a cluster representative is derived from a mean (Chen, 2007b; Simon et al., 2006) or a median (Chen, 2007a). LSTSC requires a lag value by assuming that an input sequence is cyclic. However, a perfect cyclic sequence is scarcely found in real-world data; the output of LSTSC is meaningless if an improper value is chosen (Chen, 2007a). In other words, LSTSC works well when a good lag value is provided by users. To achieve a cluster representative, LSTSC uses a mean or a median of cluster members. Since resampling of subsequences using a lag value cannot be done easily, the cluster representative derived from the mean is still meaningless. In addition, using a median instead of a mean is not a good solution. Although a median selected from an existing sequence is not a sine wave, a median is still not suitable to be a cluster representative due to lack of reflection of data characteristics. Note that some papers (Chen, 2007b) utilize lag-based approach, but subsequences are not normalized before clustering; those papers are, therefore, considered meaningless as well.

Phrase-Analysis Subsequence Time Series Clustering (PASTSC) (Fujimaki et al., 2008) utilizes Discrete Fourier Transform (DFT) to convert a time series sequence to a frequency domain before clustering. For efficient transformation, PASTSC selects the phase which gives the maximum power spectrum as a parameter in DFT. After all subsequences are transformed, those data are clustered using  $k$ -means clustering or  $k$ -hierarchical clustering algorithms, and then a cluster representative for each clustering is identified in the frequency domain. After clustering is finished, a cluster representative is transformed back to the original time domain. PASTSC has an important parameter, i.e., a slide length that is a number of overlapping subsequences allowed. Since the slide length is used to eliminate trivial-matched subsequences, an inappropriate value



still leads to meaninglessness as well. Although the slide length is set so that the output is not meaningless, the cluster representative is not generated from all sequences that some important subsequences are discarded by the slide length.

Perceptually Important Point (PIP) (Fu et al., 2005) has been proposed to reduce the number of dimensions of subsequences before clustering, where PIP captures peaks and valleys of subsequences. Specifically, extracted subsequences are first reduced using PIP, and then redundant subsequences which have the same PIP will be removed, where trivial-matched subsequences normally have similar PIPs. The parameter of this dimensionality reduction is the number of points that is used to represent a subsequence. Additionally, this method is suitable for noisy time series sequences but not smooth sequences since peaks and valleys are hard to be identified in the smooth sequences. However, their paper does not evaluate their clustering results with meaningfulness measurement. Similar to PIPs, many other representation techniques, e.g., Discrete Cosine Transform (DCT) (Kumar et al., 2006) and Discrete Fourier Transform (DFT) (Fujimaki et al., 2008), are also proposed to represent extracted subsequences to be an input of subsequence for clustering algorithms instead of using raw subsequences. However, data representations are not suitable since they require parameters, and precisions of clustering results are lost after these transformations.

These related works (Keogh and Lin, 2005; Denton, 2005; Chen, 2007a; Goldin et al., 2006; Fu et al., 2005; Struzik, 2003; Simon et al., 2006; Kumar et al., 2006; Fujimaki et al., 2008) do not propose the right solutions to deal with trivial-matched subsequences, i.e., new distance measures requires additional parameters and Amplitude Averaging is still used to construct a cluster representative. The distance threshold in DSTSC, the lag value in LSTSC, the slide length in PASTSC, and PIP are additional parameters that users must specify depending on characteristics of each dataset, where these values are sensitive to clustering results. With incorrect values, outputs of clustering results may be meaningless. In addition, these values are used to discard trivial-matched subsequences; therefore, some important trivial-matched subsequences are unexpectedly filtered out. For the meaningfulness measurement, all previous works used Keogh-Lin Meaningfulness Measurement (KLMM) to measure clustering output. As shown in Chapter 2, KLMM turns out to be an invalid measurement since it cannot completely capture similarity of two cluster representatives, when the outputs are all sine waves with different phases and frequencies; the outputs will always be interpreted as meaningless.

In this chapter, the issues of similarity between trivial-matched subsequences and cluster representative construction are solved by using the well-known DTW distance and the proposed Shape-based Averaging instead of Euclidean distance and Amplitude Averaging, respectively.

With DTW distance and Shape-based Averaging, the proposed subsequence clustering, Shape-based Subsequence Time Series Clustering (2STSC) will be the first meaningful subsequence clustering algorithm in terms of Shape-based Meaningfulness Measurement (SMM) demonstrated in an experimental evaluation section.

## 4.2 Shape-based Subsequence Time Series Clustering (2STSC)

Shape-based Subsequence Time Series Clustering (2STSC), a meaningful subsequence clustering algorithm, is proposed in this thesis, where 2STSC utilizes Dynamic Time Warping (DTW) distance and Shape-based Averaging to correctly measure similarity between subsequences and average cluster members for a cluster representative. Shape-based Averaging proposed in this chapter has two variations, i.e., Cubic-Spline Dynamic Time Warping (CDTW) and Iterative Cubic-Spline Dynamic Time Warping (ICDTW) averaging functions. Both CDTW and ICDTW functions use cubic spline interpolation function (Burden et al., 1997) to re-sample  $x$ -axis of an averaged sequence, but ICDTW function is more accurate that an averaged result is guaranteed to be in the middle of two original sequences.

To solve the problem of trivial-matched subsequences, contiguous subsequences with small time shifts, 2STSC integrates DTW distance and Shape-based Averaging in  $k$ -hierarchical clustering. Specifically, like STSC, 2STSC receives a long time series sequence  $S = \langle s_1, s_2, \dots, s_n \rangle$  as an input, and then this sequence is extracted to be a set  $\mathbb{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_i, \dots, \mathcal{S}_{n-w+1}\}$  of subsequences by a sliding window of length  $w$ , where  $\mathcal{S}_i = \langle s_i, s_{i+1}, \dots, s_{i+w-1} \rangle$  and  $1 \leq i \leq n - w + 1$ . This set of subsequences is then normalized under  $z$ -normalization (Han and Kamber, 2000) and clustered with  $k$ -hierarchical clustering algorithm, where DTW distance and Shape-based Averaging are used as a distance measure and an averaging function in the algorithm. Finally, 2STSC returns a set  $\mathbb{C} = \{C_1, C_2, \dots, C_k\}$  of  $k$  clusters, where each cluster  $C = (\mathbb{M}, R)$  contains a set  $\mathbb{M} = \{S_i \mid S_i \in \mathbb{S}\}$  of cluster members and a cluster representative  $R = \langle r_1, r_2, \dots, r_w \rangle$  from  $k$ -hierarchical clustering. Beside an input, 2STSC requires two typical parameters which are the number of clusters ( $k$ ) and the length of sliding window ( $w$ ). Visually, an overview of 2STSC is illustrated in Figure 4.5.

$K$ -hierarchical clustering used in 2STSC are agglomerative clustering which uses bottom-up strategy. Specifically, agglomerative clustering iteratively combines atomic clusters to one large cluster.  $K$ -hierarchical clustering requires an inter-cluster distance function which is used to calculate a distance between two clusters. In this thesis, 2STSC uses two inter-cluster distance functions, i.e., complete linkage and average linkage distance functions, where complete linkage and average linkage functions are maximum and mean distances, respectively, among all subse-

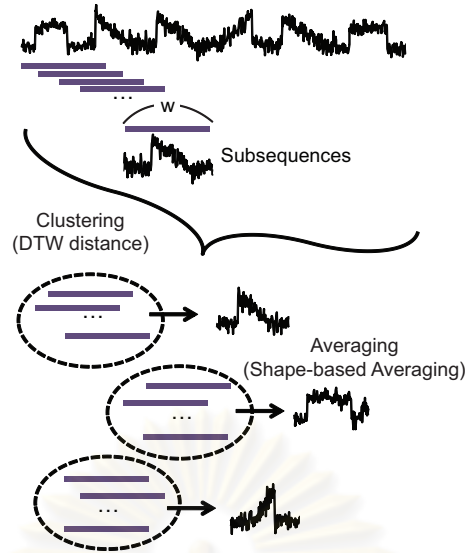


Figure 4.5: Overview of 2STSC using DTW distance and Shape-based Averaging.

quences pairs between two cluster members. More details of agglomerative clustering algorithms are provided in Section 2.1.2. Concretely, 2STSC with the agglomerative algorithm is shown in Table 4.1. Note that 2STSC does not use the single linkage function as an inter-cluster distance function because the single linkage function cannot handle trivial-matched subsequences. Specifically, some subsequences will never be in any group if these subsequences have the largest nearest neighbor distance. Although an average distance of that subsequence is smaller than others, single linkage will only group based on the smaller nearest neighbor distance. Therefore, in this thesis, only two inter-cluster distance functions are utilized, i.e., complete linkage and average linkage distance functions.

Table 4.1: Pseudo code of Shape-based Subsequence Time Series Clustering (2STSC)

FUNCTION [C] = 2STSC [S, k, w]
1. $\mathbb{S} = \text{EXTRACTSUBSEQUENCES}(S, w)$
2. $\mathbb{S}_{Norm} = \text{NORMALIZESUBSEQUENCES}(\mathbb{S})$
3. $\mathbb{C} = \text{CLUSTERING}(\mathbb{S}_{Norm}, k)$ // with DTW distance and Shape-based Averaging
4. Return $\mathbb{C}$

### 4.3 Experimental Evaluation

Shape-based Subsequence Time Series Clustering (2STSC) is evaluated by comparing with STSC in terms of meaningfulness. STSC used in this experiment is implemented on  $k$ -means clustering and  $k$ -hierarchical clustering with Euclidean distance and Amplitude Averaging, while 2STSC is implemented with  $k$ -hierarchical clustering with DTW distance and Shape-based Averaging (CDTW and ICDTW functions). Eight datasets from the Time Series Data Mining Archive (TSDMA) (Keogh and Folias, 2011) used in this experiment are normalized and shown in Ap-

pendix A, where each dataset contains 2000 data points. Two datasets, i.e., Buoy1 and CBF, used to illustrate in this experiment is shown in Figure 4.6.

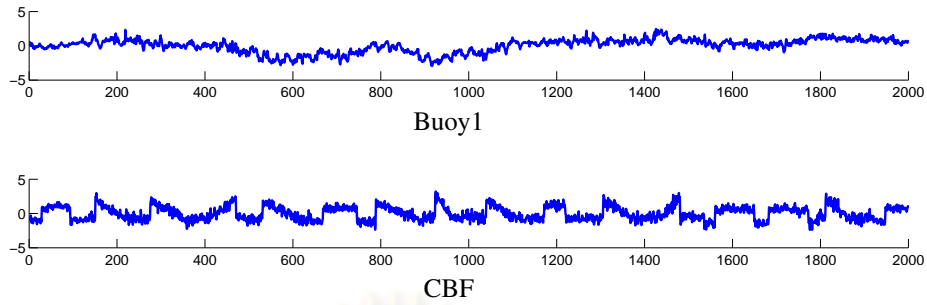


Figure 4.6: Datasets used to evaluate meaningfulness of STSC and 2STSC

The proposed 2STSC is compared with STSC in terms of meaningfulness. However, Keogh-Lin Meaningfulness Measurement (KLMM) (Lin et al., 2003; Keogh and Lin, 2005) is invalid by the following reasons. First, the assumption of KLMM is that clustering results from the same input sequence should be similar; otherwise, the clustering results should be dissimilar. KLMM, therefore, only compares the distances between the distance of clustering results from the same input and the distance of clustering results from the different inputs. However, KLMM does not have any measurement of similarity between two inputs. Given two similar sequences, clustering results from those two inputs are expected to be similar, but KLMM considers that the results are meaningless although the algorithm produces meaningful results. The second reason is that KLMM cannot capture the similarity of sine waves with different phases or frequencies since KLMM utilizes Euclidean distance to calculate distance between two cluster representatives. Therefore, these cluster results are dissimilar in terms of KLMM although the clustering results are sine waves. In Chapter 2, KLMM has been shown that it is considered to be an invalid meaningfulness measurement.

In this experiment, a novel meaningfulness measurement, Shape-based Meaningfulness Measurement (SMM), is introduced to calculate meaningfulness of clustering results. The basic idea of SMM is that clustering results are meaningful if clustering results truly represent subsequences in the time series sequence. In other words, if an input sequence is not a sine wave, cluster representatives should not be sine waves, and if an input sequence is a sine wave, clustering representatives should be sine waves; otherwise, the clustering results are considered meaningless. Unlike KLMM, SMM calculates the meaningfulness between an input sequence and an output clustering result, while KLMM calculates meaningfulness between clustering results from two different datasets. Given an input sequence  $S = \langle s_1, s_2, \dots, s_n \rangle$  and an output set  $\mathbb{C} = \{C_1, C_2, \dots, C_k\}$  of  $k$  clusters, a set  $\mathbb{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_i, \dots, \mathcal{S}_{n-w+1}\}$  of subsequences are extracted from the input sequence  $S$  by a sliding window of length  $w$ , where each cluster

$C = (\mathbb{M}, R)$  contains a set  $\mathbb{M} = \{\mathcal{S}_i \mid \mathcal{S}_i \in \mathbb{S}\}$  of cluster members and a cluster representative  $R = \langle r_1, r_2, \dots, r_w \rangle$ . A set  $\mathbb{R} = \{R_1, R_2, \dots, R_k\}$  of cluster representatives are cluster representatives of all clusters. Specifically, SMM is a summation of minimum distances between each subsequence and cluster representatives. The meaningfulness value can be calculated as the following equation.

$$SMM(S, \mathbb{C}) = \frac{|\mathbb{S}| \cdot w}{\sum_{i=1}^{|\mathbb{S}|} \min(Distance(\mathcal{S}_i, R_j)), \forall R_j \in \mathbb{R}} \quad (4.1)$$

where  $Distance(\mathcal{S}_i, R_j)$  is a DTW distance between two sequences  $\mathcal{S}_i$  and  $R_j$ .

SMM ranges from zero to positive infinity and is a relative value that SMM must be compared between two algorithms at the same set of parameters to identify that with a given dataset, which subsequence clustering algorithm produces more meaningful clustering results.

Two parameters, i.e., the length of sliding window ( $w$ ) and the number of clusters ( $k$ ), are varied to demonstrate the meaningfulness of clustering results of seven variations of subsequence clustering algorithms. Figures 4.7 and 4.8 show SMMs of two datasets when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied to be 32, 64, and 128, and Figures 4.9 and 4.10 show SMMs of two datasets when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied to be 3, 5, and 7. Figures 4.11 and 4.12 show cluster representatives of 2STSC of Buoy1 and CBF, respectively, when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is 64. The results of other parameter settings and datasets are reported in Appendix D.

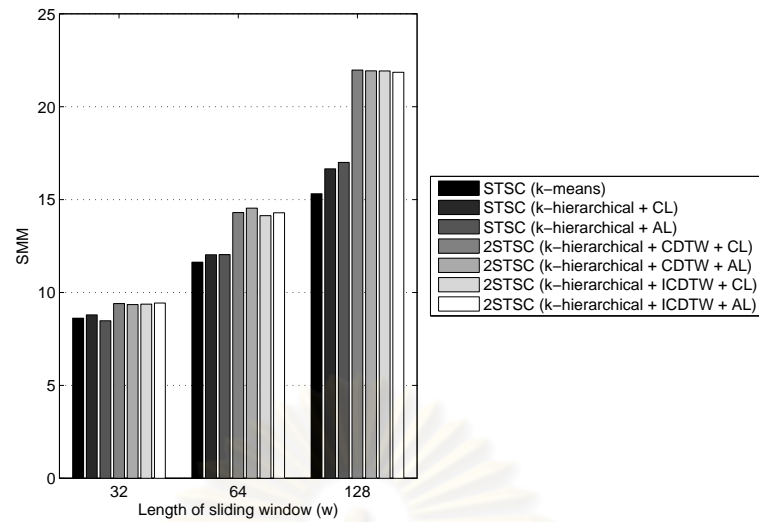


Figure 4.7: SMMs of Buoy1 when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied.

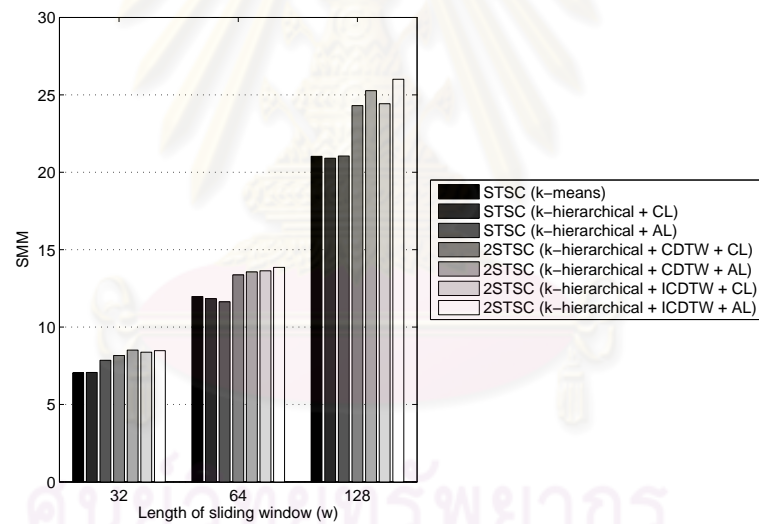


Figure 4.8: SMMs of CBF when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied.

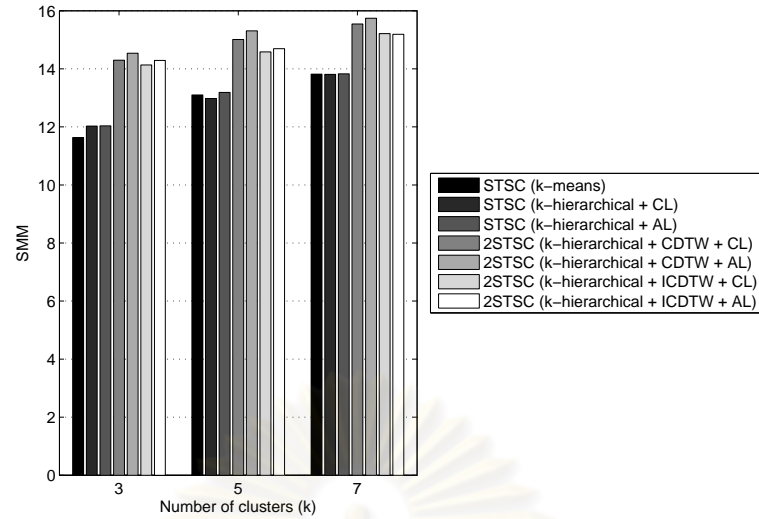


Figure 4.9: SMMs of Buoy1 when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied.

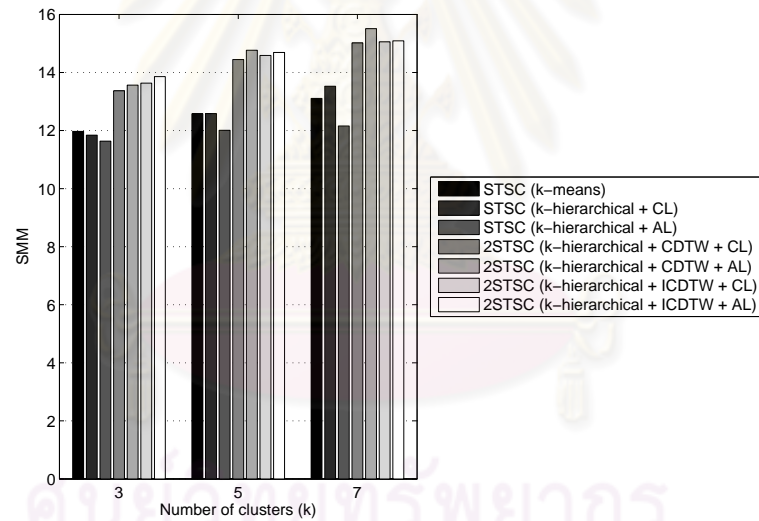


Figure 4.10: SMMs of CBF when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied.

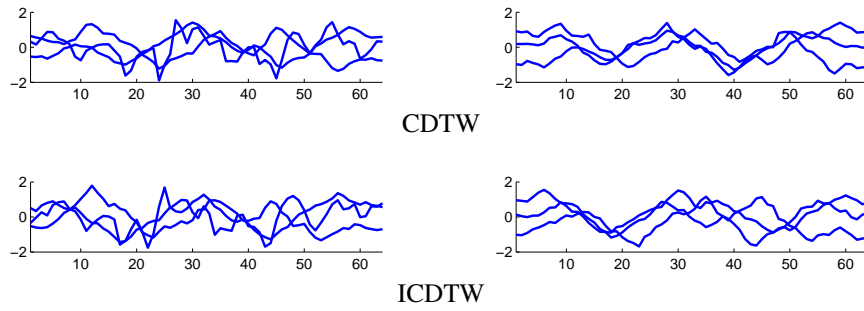


Figure 4.11: Cluster representatives generated from 2STSC of Buoy1 with complete linkage (left) and average linkage (right) when  $k = 3$  and  $w = 64$ .

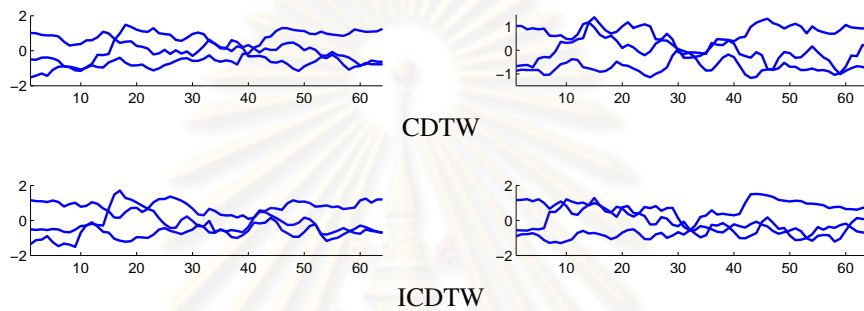


Figure 4.12: Cluster representatives generated from 2STSC of CBF with complete linkage (left) and average linkage (right) when  $k = 3$  and  $w = 64$ .

#### 4.4 Conclusion

DTW distance and Shape-based Averaging are proposed to be used as a distance measure and an averaging function in Shape-based Subsequence Time Series Clustering (2STSC) instead of Euclidean distance and Amplitude Averaging in Subsequence Time Series Clustering (STSC). Instead of discarding trivial-matched subsequences as in many other proposed works, 2STSC uses appropriate distance measure and averaging function that uses DTW distance to capture the similarity between a set of contiguous subsequences and Shape-based Averaging to construct a characteristic-preserved cluster representative. In addition, the cluster results from 2STSC are meaningful in terms of Shape-based Meaningfulness Measurement (SMM) which measures how well a clustering result truly represents characteristics of an input time series sequence. Cluster representatives generated from 2STSC do reflect the characteristics of input sequences, while STSC produces undesired outputs like sine waves. In addition, 2STSC requires no additional parameter like other proposed subsequence clustering algorithms, and 2STSC is extensible to support data streams in Chapter 6.



## CHAPTER V

### INCREMENTAL SHAPE-BASED AVERAGING

From Chapter 3, Shaped-based Averaging is the best solution to construct a representative of a set of subsequences. For streaming applications, a new incoming sequence arrives sequentially in constant time, where an averaged result must be returned for every new incoming sequence. Generally, Shape-based Averaging constructs an averaged result by averaging an entire set of previous sequences. This is obviously impractical for the streaming case, where computational time of constructing an averaged result should not depend on the number of previous sequences which is usually large. Specifically, if there are a lot of previous subsequences, it is not possible to guarantee that a new averaged result will be constructed in time before the next subsequence arrives. Instead of averaging all previous sequences for every new incoming sequence, Iterative Shape-based Averaging creates an averaged result only with a small set of stored sequences. Therefore, time complexity of Incremental Shape-based Averaging depends only on the number of stored subsequences, where the number is much smaller than the number of previous subsequences.

In this chapter, Incremental Shape-based Averaging with two averaging functions, Cubic-Spline Dynamic Time Warping (CDTW) and Iterative Cubic-Spline Dynamic Time Warping (ICDTW) averaging functions, is proposed. The experiments will show that Incremental Shape-based Averaging is much faster than Shape-based Averaging in orders of magnitude, while Incremental Shape-based Averaging maintains low averaging distortion.

#### 5.1 Incremental Shape-based Averaging

Incremental Shape-based Averaging is a method used to incrementally construct averaged result when a set of stored sequences is given with a new incoming sequence. For streaming applications, constructing an averaged result from all previous sequences for every single incoming sequence with limited computational power and storage is simply impractical. Therefore, only some sequences are stored and used to generate an averaged result.

Given a set  $\mathbb{T} = \{T_1, T_2, \dots, T_t\}$  of stored sequences, a set  $\mathbb{W} = \{w_1, w_2, \dots, w_t\}$  of weights of stored sequences, a new incoming sequence  $S$ , and the maximum allowance in the number of stored sequences  $\alpha$ , where  $t$  is a number of stored sequences, Incremental Shape-based Averaging returns an averaged result  $C$ . Initially, sets  $\mathbb{T}$  and  $\mathbb{W}$  are empty, and  $\alpha$  is a user-defined

parameter. When a new incoming sequence  $S$  arrives, the sets  $\mathbb{T}$  and  $\mathbb{W}$  are first updated. If the number of stored sequences  $t$  is less than the maximum allowance  $\alpha$ ,  $S$  is added to  $\mathbb{T}$ , and the weight of  $S$ , which is initially assigned to 1, is added to the set  $\mathbb{W}$ ; otherwise, a stored sequence  $T_i$  which is the most similar to the sequence  $S$  under DTW distance is replaced with the averaged result between the sequences  $T_i$  and  $S$  with weights of  $w_i$  and 1, respectively. Therefore, the sets  $\mathbb{T}$  and  $\mathbb{W}$  are updated with the sequence  $S$ , as shown in Table 5.2. After the sets  $\mathbb{T}$  and  $\mathbb{W}$  are updated, Incremental Shape-based Averaging constructs an averaged result from the copies of  $\mathbb{T}$  and  $\mathbb{W}$ , i.e.,  $\mathbb{T}_{temp}$  and  $\mathbb{W}_{temp}$ , by iteratively averaging the most similar pair of sequences within  $\mathbb{T}_{temp}$  until only one sequence is left. Its pseudo code is shown in Table 5.3. Note that when the maximum allowance  $\alpha$  is positive infinity, to update an averaged result, all previously stored sequences are calculated; therefore, Shape-based Averaging is a special case of Incremental Shape-based Averaging when the maximum allowance  $\alpha = \infty$ . Pseudo code of Incremental Shape-based Averaging is provided in Table 5.1.

Table 5.1: Pseudo code of Incremental Shape-based Averaging

FUNCTION $[C] = \text{INCREMENTALSHAPE-BASEDAVERAGING} [\mathbb{T}, \mathbb{W}, S, \alpha]$
1. $[\mathbb{T}, \mathbb{W}] = \text{UPDATESTOREDSEQUENCES}(\mathbb{T}, \mathbb{W}, S)$
2. $C = \text{AVERAGESTOREDSEQUENCES}(\mathbb{T}, \mathbb{W})$
3. Return $C$

Table 5.2: Updating stored sequences in Incremental Shape-based Averaging

FUNCTION $[\mathbb{T}, \mathbb{W}] = \text{UPDATESTOREDSEQUENCES} [\mathbb{T}, \mathbb{W}, S, \alpha]$
1. Let $t$ be a number of stored sequences in $\mathbb{T}$
2. If $(t < \alpha)$
3.     Add $S$ in $\mathbb{T}$
4.     Add $w = 1$ in $\mathbb{W}$
5. Else
6. $dist_{Best} = \text{INFINITY}$
7.     For each stored sequence $T_i$ in $\mathbb{T}$ and $w_i$ in $\mathbb{W}$
8. $dist = \text{DTW-DISTANCE}(T_i, S)$
9.         If $(dist < dist_{Best})$
10. $dist_{Best} = dist$
11. $T_{Best} = T_i$
12. $w_{Best} = w_i$
13.         End if
14.     End for
15. $S_{avg} = \text{AVERAGINGFUNCTION}(T_{Best}, S, w_{Best}, 1)$
16.     Replace $T_{Best}$ with $S_{avg}$
17.     Replace $w_{Best}$ with $w_{Best} + 1$
18.     End If
19. Return $[\mathbb{T}, \mathbb{W}]$

Table 5.3: Averaging stored sequences in Incremental Shape-based Averaging

FUNCTION $[T_k] = \text{AVERAGESTOREDSEQUENCES} [\mathbb{T}, \mathbb{W}]$	
1.	Let $\mathbb{T}_{temp}$ be a copy of $\mathbb{T}$
2.	Let $\mathbb{W}_{temp}$ be a copy of $\mathbb{W}$
3.	While ( $\text{SIZE}(\mathbb{T}_{temp}) > 1$ )
4.	$[T_i, T_j] = \text{Most similar pair of sequences in } \mathbb{T}_{temp}$
5.	$T_k = \text{AVERAGINGFUNCTION}(T_i, T_j, w_i, w_j)$
6.	Remove $T_i$ and $T_j$ from $\mathbb{T}_{temp}$
7.	Remove $w_i$ and $w_j$ from $\mathbb{W}_{temp}$
8.	$w_k = w_i + w_j$
9.	Add $T_k$ to $\mathbb{T}_{temp}$
10.	Add $w_k$ to $\mathbb{W}_{temp}$
11.	End while
12.	Return $T_k$

## 5.2 Experimental Evaluation

Iterative Shape-based Averaging constructs a new averaged result from only the stored sequences and a new incoming sequence instead of constructing from all previous sequences. Two experiments are designed to demonstrate that Incremental Shape-based Averaging is suitable for streaming applications. The first experiment shows that Incremental Shape-based Averaging is much faster than Shape-based Averaging in orders of magnitude, and the second experiment demonstrates that Incremental Shape-based Averaging, with available storage and computational power, achieves comparable accuracy to Shape-based Averaging with very small distortion, while Incremental Shape-based Averaging is still faster than Shape-based Averaging. Twenty datasets used in this experiment are from the Time Series Clustering/Classification datasets (Keogh et al., 2011). The details of each dataset are provided in Table A.1 in Appendix A, and the examples of each datasets are shown in Figure A.2. In this experimental evaluation, two datasets, i.e., CBF and ECG, are mainly used as shown in Figure 5.1.

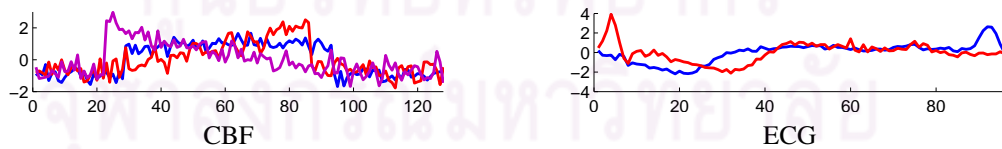


Figure 5.1: Examples of some classes in evaluated datasets.

### 5.2.1 First Experiment

The first experiment shows the significant speedup of Incremental Shape-based Averaging over Shape-based Averaging, where the maximum allowance  $\alpha$  is set to one. For every new incoming sequence, Incremental Shape-based Averaging calculates an averaged result from the

stored sequence, and then this averaged result is used for the next incoming sequence. For Shape-based Averaging, an averaged result is created from all previous sequences for every new incoming sequence which is impractical from streaming data. Figure 5.3 shows time consumption of Incremental Shape-based Averaging compared with Shape-based Averaging using two averaging functions, i.e., CDTW and ICDTW, respectively. From the result, Incremental Shape-based Averaging requires only constant time to update an averaged result, while computational time of Shape-based Averaging grows exponentially. In addition, Incremental Shape-based Averaging is nearly  $10^7$  times faster. Additional results of this experiment are provided in Appendix E.

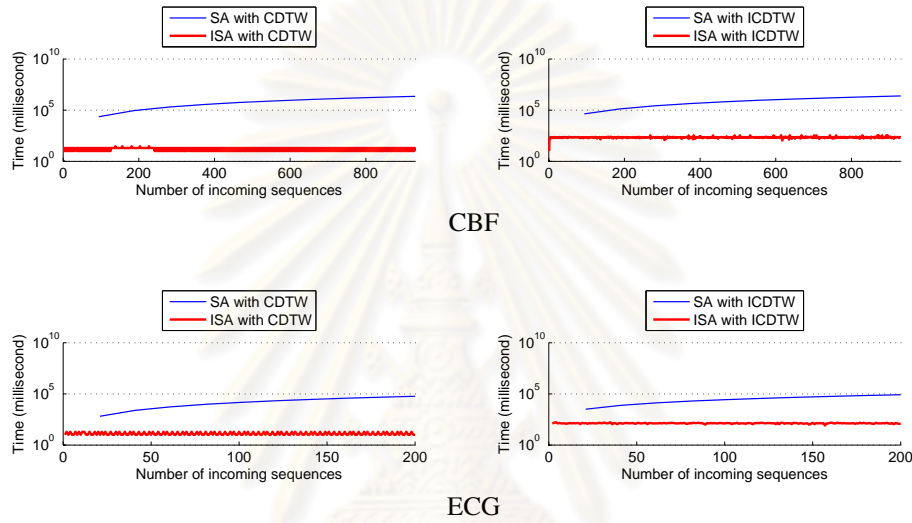


Figure 5.2: Computational time of Incremental Shape-based Averaging and Shape-based Averaging when a new incoming sequence arrives.

### 5.2.2 Second Experiment

The second experiment shows SUMDIST distance when Incremental Shape-based Averaging is used instead of Shape-based Averaging. Since Shape-based Averaging has no associative property, the updated averaged result from Incremental Shape-based Averaging is not equal to that from averaging all sequences using Shape-based Averaging, where SUMDIST distance can be calculated by the following equation.

$$\text{SumDist}(\hat{\mathcal{S}}, \mathbb{S}) = \sum_{i=1}^{|\mathbb{S}|} \text{DTWDistance}(\hat{\mathcal{S}}, \mathcal{S}_i) \quad (5.1)$$

where  $\mathbb{S}$  is a dataset,  $\hat{\mathcal{S}}$  is the averaged result, and  $\mathcal{S}_i$  is each data sequence in the dataset  $\mathbb{S}$ .

In this experiment, with available computational power and storage, Incremental Shape-

based Averaging can achieve SUMDIST close to Shape-based Averaging, where Shape-based Averaging is a special case of Incremental Shape-based Averaging when the maximum allowance number  $\alpha$  is set to a positive infinity. Each class in a dataset is separately evaluated, and SUMDIST of each dataset is reported by summarizing SUMDIST of every class. Difference of SUMDISTs and speedup of Buoy1 and CBF when  $k = 3$ ,  $w = 64$ , and the maximum allowance number  $\alpha$  are varied in percentage to the size of dataset are shown in Figure 5.3 and 5.4, respectively. Figures 5.5 and 5.6 show averaged results generated from Incremental Shape-based Averaging with CDTW and ICDTW, respectively. From the experiment results, Incremental Shape-based Averaging can return averaged results much faster than Shape-based Averaging with only small distortions. Speedup and difference of SUMDIST measured in this experiment is calculated from the time used to update and average sequence of static dataset when the maximum allowance number is varied.

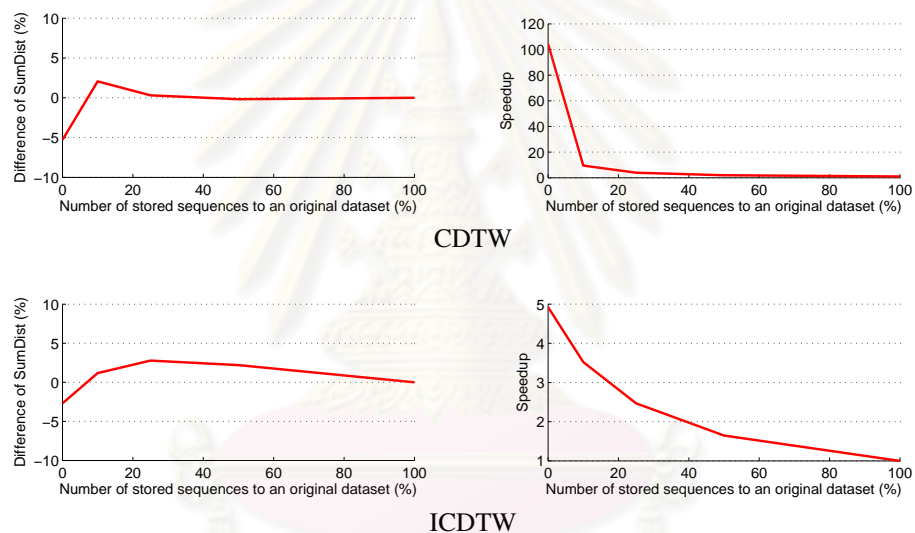


Figure 5.3: Difference of SUMDIST and speedup of Buoy1 when the number of stored sequences to an original dataset is varied.

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

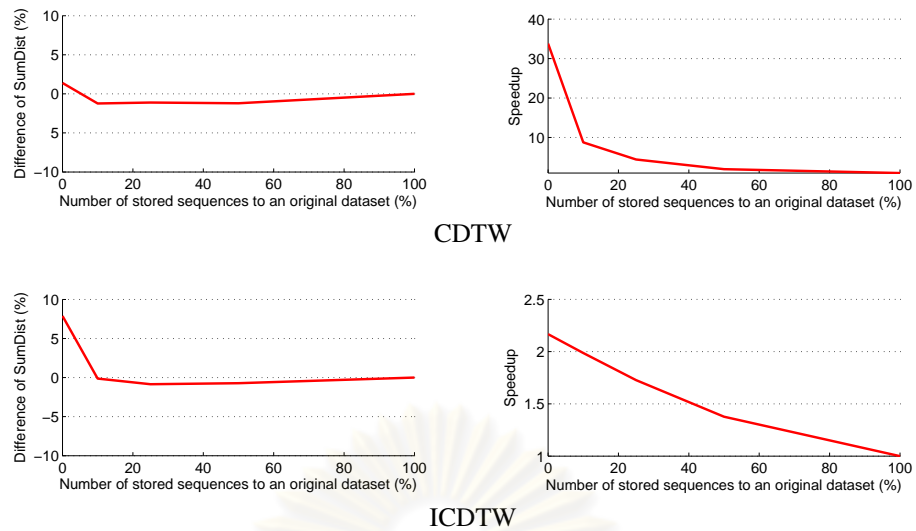


Figure 5.4: Difference of SUMDIST and speedup of CBF when the number of stored sequences to an original dataset is varied.

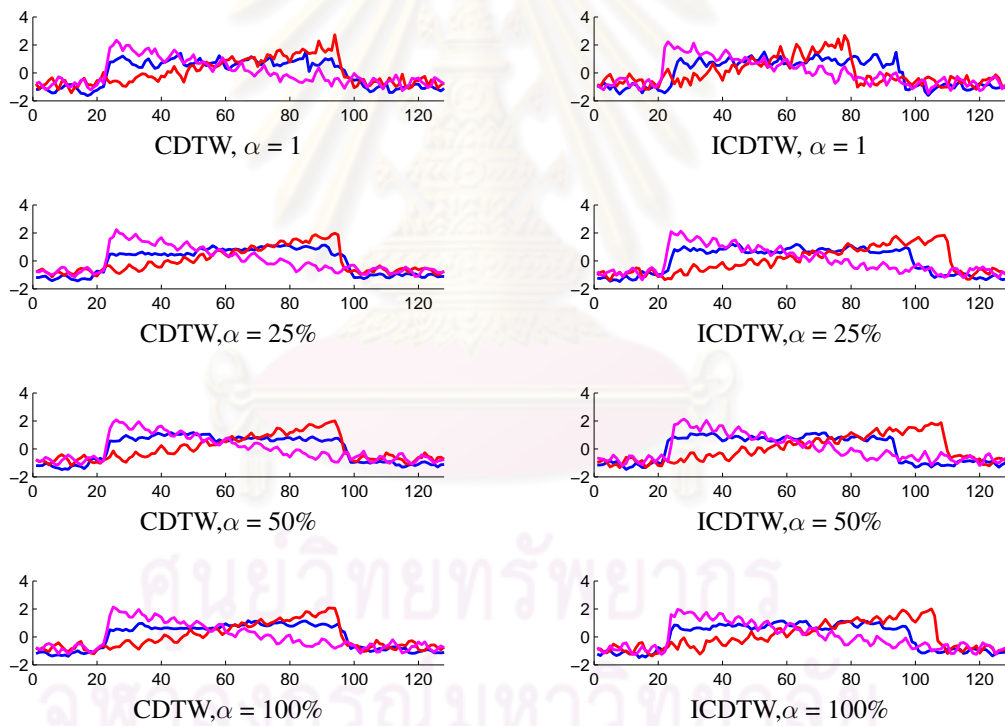


Figure 5.5: Averaged results of some classes of CBF from Incremental Shape-based Averaging.

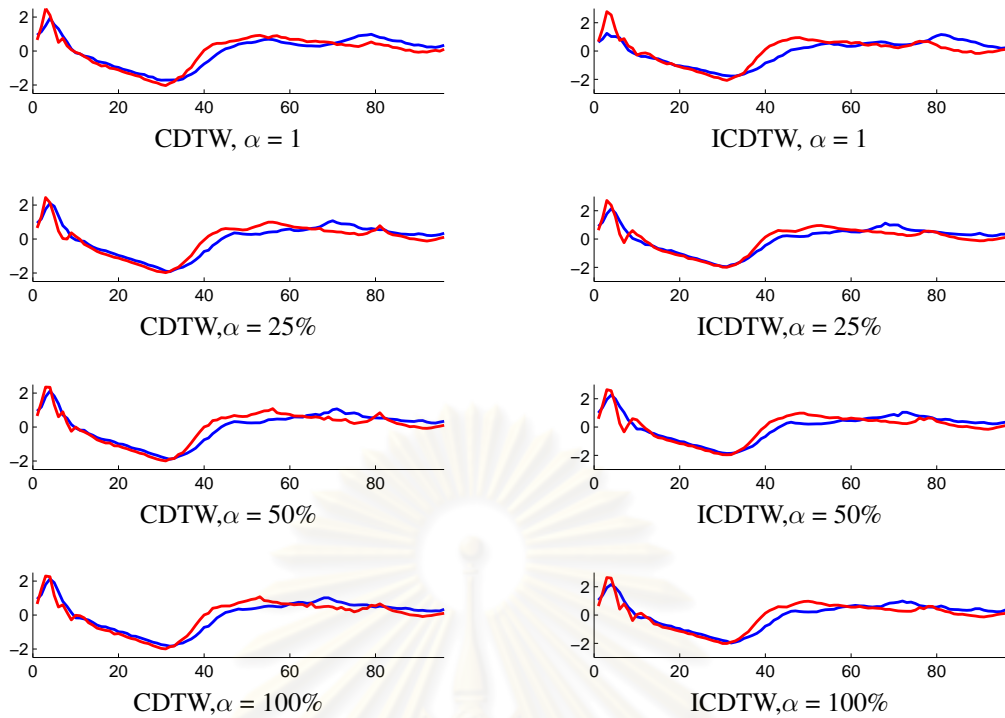


Figure 5.6: Averaged results of some classes of ECG from Incremental Shape-based Averaging.

### 5.3 Conclusion

Incremental Shape-based Averaging with Cubic-Spline Dynamic Time Warping (CDTW) and Iterative Cubic-Spline Dynamic Time Warping (ICDTW) averaging functions are fast and accurate. To update the averaged result, the stored sequence with its weight is updated to generate a new sequence in constant time. Therefore, instead of constructing an averaged result from all previous data sequences for each and every incoming sequence, Incremental Shape-based Averaging updates only once which reduces computational time in orders of magnitude. In addition, Incremental Shape-based Averaging is proposed to be able to store more than one sequence to increase accuracy if more computational power or storage is available. Moreover, Incremental Shape-based Averaging can be widely extended to construct a shape-based averaged result in streaming applications, whose idea of sequence updates in Shape-based Streaming Subsequence Time Series Clustering (3STSC) is explained in Chapter 6.

## CHAPTER VI

### 3STSC: SHAPE-BASED STREAMING SUBSEQUENCE TIME SERIES CLUSTERING

In time series domain, streaming clustering algorithms are divided into two categories, i.e., streaming whole clustering (Rodrigues et al., 2006, 2008) and streaming subsequence clustering. For the streaming whole clustering, the new whole sequence is used to update the clustering result or cluster representatives, while for the streaming subsequence clustering, after the new data point is concatenated, a subsequence is extracted from a fixed-length sliding window, subsequence is normalized, and then the cluster representatives are updated from this subsequence. The naïve algorithm of the streaming problem is that the output of the algorithm is calculated from all previous input subsequences for every new incoming sequence. In this chapter, the streaming subsequence clustering is focused.

As shown in Chapter 2, Keogh and Lin have proved that outputs from Subsequence Time Series Clustering (STSC) are meaningless; therefore, currently, no meaningful naïve algorithm for streaming clustering algorithm exists. In Chapter 4, 2STSC is proposed to return a meaningful clustering result, where Dynamic Time Warping (DTW) distance and Shape-based Averaging function are used as a distance measure and an averaging function instead of Euclidean distance and Amplitude Averaging function as in STSC, respectively. In this chapter, 2STSC is considered as a naïve algorithm of a streaming application. Since 2STSC calculates a clustering result from all previous subsequences, it is impractical since the computational time depends on the number of previous subsequences which increases over time.

In this chapter, Streaming Shape-based Subsequence Time Series Clustering (3STSC) is proposed to efficiently update the clustering result in constant time to the number of previous subsequences. Instead of calculating the clustering result from all previous subsequences as in 2STSC, 3STSC calculates the clustering result from the small number of stored subsequences. The algorithm of updating stored subsequences in 3STSC is the same as that of the Incremental Shape-based Averaging , where the number of stored subsequences is maintained not to exceed the maximum allowance number of stored subsequences. 3STSC then groups these stored subsequences into clusters using  $k$ -hierarchical clustering with Dynamic Time Warping (DTW) distance and Shape-based Averaging function as a distance measure and an averaging function. In other words, 3STSC returns a clustering result from a small set of stored subsequence which is much



faster than 2STSC which returns a clustering result from all previous subsequences.

In experimental evaluation, 3STSC shows superiority over 2STSC in terms of computational time, and the clustering result of 3STSC is also compared in terms of Shape-based Meaningfulness Measurement (SMM) when the parameters, i.e., the number of clusters ( $k$ ), the length of sliding window ( $w$ ), and the maximum allowance number of stored subsequences ( $\alpha$ ), are varied.

## 6.1 Related Work

Clustering time series data streams is divided into two categories, i.e., streaming whole clustering and streaming subsequence clustering. Streaming whole clustering is an incremental clustering, where a whole time series sequence arrives constantly. No sliding windows are involved in the algorithm. A new arriving whole sequence is used to update a clustering structure such as a tree of hierarchical clustering. Rodrigues et al. have proposed Online Divisive Agglomerative Clustering (ODAC) (Rodrigues et al., 2008) for time series data streams which implements splitting and merging operations for updating a tree-like hierarchy of clusters that do not depend on the number of data objects in the data stream. For streaming subsequence clustering, a set of clusters is returned for every incoming data point. However, no existing algorithm has been proposed yet. Although many subsequence clustering algorithms are proposed such as Density-based Subsequence Clustering (DSTSC) (Denton, 2005), Lag-based Subsequence Time Series Clustering (LSTSC) (Simon et al., 2006; Chen, 2007b,a), and Phrase-Analysis Subsequence Time Series Clustering (PASTSC) (Fujimaki et al., 2008), no extension of streaming applications has been introduced. In addition, as mentioned in Chapter 4, these subsequence clustering algorithms still do not produce meaningful clustering results.

Many problems on time series data streams such as subsequence matching, motif discovery, and stream monitoring have been increasingly the topics of interest. For subsequence matching (Sakurai et al., 2005; Niennattrakul and Ratanamahatana, 2009; Niennattrakul et al., 2009), a template query is given and a set of nearest subsequences is returned. Motif discovery for data streams (Mueen and Keogh, 2010) is a method to maintain the best-matched subsequence pair in a given time series sequence. Stream monitoring (Kontaki et al., 2008; Dai et al., 2006) is a method to find correlations among data streams.

In this study, streaming subsequence clustering is considered the first streaming subsequence clustering algorithm that produces meaningful clustering results.

## 6.2 Shape-based Streaming Subsequence Time Series Clustering

Shape-based Streaming Subsequence Time Series Clustering (3STSC) is an incremental subsequence clustering algorithm that returns a set of cluster representatives for every new data point arrival. Specifically, 3STSC first concatenates a new data point with the previous time series sequence. A new subsequence is then extracted with a fixed-length sliding window, and then the subsequence is normalized by  $z$ -normalization. Since the maximum allowance number of stored subsequences needs to be maintained, the set of stored subsequences is updated by a new sequence not to exceed the maximum allowance number. After the set of stored subsequences is updated, 3STSC then finds a clustering result using  $k$ -hierarchical clustering on these stored subsequences with Dynamic Time Warping (DTW) distance and Shape-based Averaging function as a distance measure and an averaging function, respectively. Additionally, the updating algorithm of the stored subsequences is similar to the Incremental Shape-based Averaging function. Note that the maximum allowance number of the stored subsequences is a user-defined parameter depending on the availability of computational power and storage. The overview of 3STSC is provided in Figure 6.1.

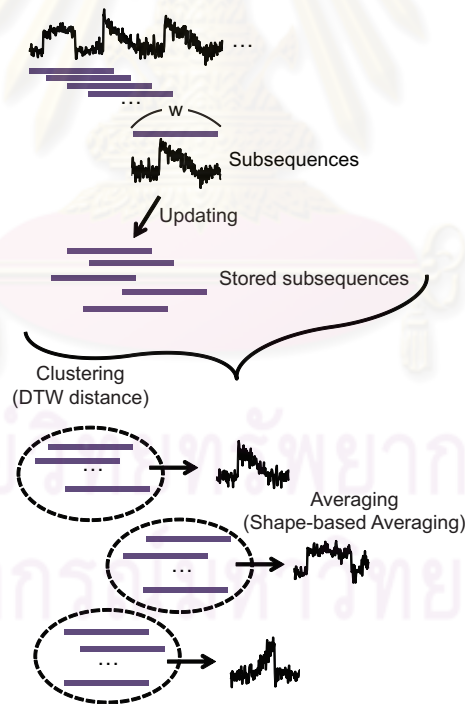


Figure 6.1: Overview of Shape-based Streaming Subsequence Time Series Clustering (3STSC).

Given a new data point  $s_t$ , the number of clusters  $k$ , the length of sliding window  $w$ , and the maximum allowance  $\alpha$  of stored subsequences, 3STSC returns a set  $\mathbb{C} = \{C_1, C_2, \dots, C_k\}$  of clusters. 3STSC first concatenates  $s_t$  to a streaming time series  $S = \langle s_1, s_2, \dots, s_{t-1} \rangle$ , and then a new subsequence  $\mathcal{S} = \langle s_{t-w+1}, \dots, s_{t-1}, s_t \rangle$  is extracted with the fixed-length sliding window

of length  $n$ . In addition, this new subsequence  $\mathcal{S}_{norm}$  is normalized by  $z$ -normalization. 3STSC updates a set  $\mathbb{T} = \{T_1, T_2, \dots, T_\alpha\}$  of stored subsequences using an updating algorithm from Incremental Shape-based Averaging. After the set  $\mathbb{T}$  is updated, subsequences in the set  $\mathbb{T}$  are clustered and return a set  $\mathbb{C} = \{C_1, C_2, \dots, C_k\}$  of clusters using  $k$ -hierarchical clustering with Dynamic Time Warping (DTW) distance and Shape-based Averaging is returned. Each cluster  $C$  contains a set  $\mathbb{M} = \{T_i \mid T_i \in \mathbb{T}\}$  of stored subsequences and a cluster representative  $R$ . The pseudo code of 3STSC is provided in Table 6.1.

Table 6.1: Pseudo code of Shape-based Streaming Subsequence Time Series Clustering (3STSC)

FUNCTION $[\mathbb{C}] = 3\text{STSC} [\mathbb{T}, \mathbb{W}, s_t, k, w, \alpha]$	
1.	Update a streaming time series $S$ by adding a new arriving data point $s_t$
2.	$\mathcal{S} = \text{EXTRACTLASTESTSUBSEQUENCE}(S, w)$
3.	$\mathcal{S}_{norm} = \text{ZNORMALIZE}(\mathcal{S})$
4.	$\mathbb{T} = \text{UPDATESTOREDSUBSEQUENCE}(\mathbb{T}, \mathbb{W}, \mathcal{S}_{norm}, \alpha)$
5.	$\mathbb{C} = \text{KHIERARCHICALCLUSTERING}(\mathbb{T}, k)$
6.	Return $\mathbb{C}$

$K$ -hierarchical clustering used in 3STSC can be used with either complete linkage or average linkage function as an inter-cluster distance function which calculates the distance between two clusters defined as the following equations.

$$D_{complete}(C_i, C_j) = \max_{S \in \mathbb{M}_i, S' \in \mathbb{M}_j} \text{Distance}(S, S') \quad (6.1)$$

$$D_{average}(C_i, C_j) = \frac{1}{|\mathbb{M}_i| |\mathbb{M}_j|} \sum_{c \in C_i} \sum_{c' \in C_j} \text{Distance}(S, S') \quad (6.2)$$

where  $D_{complete}$  and  $D_{average}$  are complete and average linkage functions, respectively,  $C_i$  and  $C_j$  are any clusters,  $\mathbb{M}_i$  and  $\mathbb{M}_j$  are cluster members of  $C_i$  and  $C_j$ , respectively, and  $S$  and  $S'$  are sequences in  $\mathbb{M}_i$  and  $\mathbb{M}_j$ , respectively.  $\text{Distance}(S, S')$  returns a DTW distance between two sequences  $S$  and  $S'$ .

To update stored subsequences, 3STSC utilizes the updating algorithm that is similar to Incremental Shape-based Averaging, where the number of stored subsequences is maintained not to exceed the maximum allowance number ( $\alpha$ ). Specifically, the smallest possible number of the maximum allowance number ( $\alpha$ ) is equal to the number of clusters ( $k$ ). When a new subsequence  $\mathcal{S}_{norm}$  arrives, the nearest stored subsequence  $T_{Best}$  to the new subsequence  $\mathcal{S}_{norm}$  is averaged and the nearest stored subsequence  $T_{Best}$  is replaced with the averaged result, where the weight of the averaged result is increased by one. Pseudo code of the updating algorithm is provided in

Table 6.2.

Table 6.2: Updating stored sequences in 3STSC

---

FUNCTION  $[\mathbb{T}, \mathbb{W}] = \text{UPDATESTOREDSEQUENCES} [\mathbb{T}, \mathbb{W}, \mathcal{S}_{norm}, \alpha]$

---

1. Let  $n$  be a number of stored sequences in  $\mathbb{T}$
2. If  $(n < \alpha)$
3.     Add  $\mathcal{S}_{norm}$  in  $\mathbb{T}$
4.     Add  $w = 1$  in  $\mathbb{W}$
5. Else
6.      $dist_{Best} = \text{INFINITY}$
7.     For each stored sequence  $T_i$  in  $\mathbb{T}$
8.          $dist = \text{DTW-DISTANCE}(T_i, S)$
9.         If  $(dist < dist_{Best})$
10.              $dist_{Best} = dist$
11.              $T_{Best} = T_i$
12.              $w_{Best} = w_i$
13.         End if
14.     End for
15.      $\mathcal{S}_{avg} = \text{AVERAGINGFUNCTION}(T_{Best}, \mathcal{S}_{norm}, w_{Best}, 1)$
16.     Replace  $T_{Best}$  with  $\mathcal{S}_{avg}$
17.     Replace  $w_{Best}$  with  $w_{Best} + 1$
18. End If
19. Return  $[\mathbb{T}, \mathbb{W}]$

---

Note that 2STSC is a special case of 3STSC when the maximum allowance number of stored subsequences ( $\alpha$ ) is set to positive infinity.

### 6.3 Experimental Evaluation

Shape-based Streaming Subsequence Time Series Clustering (3STSC) is proposed to find a set of cluster representatives incrementally. 3STSC is evaluated in two experiments. The first experiment shows speedup of 3STSC over 2STSC, where 3STSC updates a cluster representative for every new incoming sequence in constant time, but 2STSC recalculates a set of cluster representatives in every new incoming sequence. Since the result of 2STSC and 3STSC are not the same due to the incremental algorithm of 3STSC, the second experiment demonstrates the difference of clustering results between 2STSC and 3STSC. The last experiment shows that if computational power and memory storage are available, the clustering result of 3STSC will be close to that of 2STSC. Eight datasets used in these experiments are from the Time Series Data Mining Archives (TSDMA) (Keogh and Folias, 2011) shown in A.1 in Appendix A, where each dataset contains 2000 data points. Two examples of each dataset are provided in Figure 6.2.

#### 6.3.1 First Experiment

The first experiment shows that 3STSC can return a set of clusters much faster than the naïve algorithm using 2STSC. At every new incoming data point, time to update cluster repre-

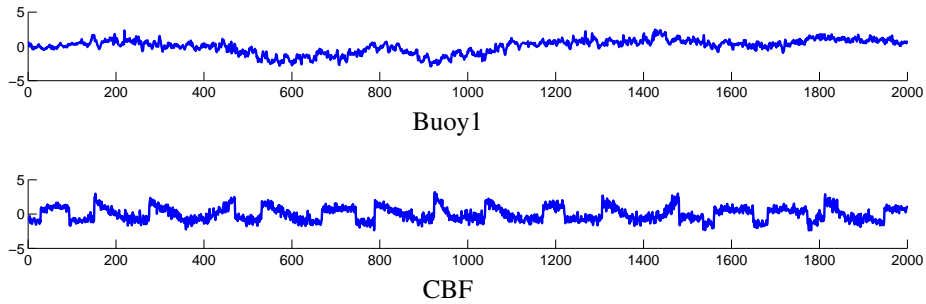


Figure 6.2: Some datasets from TSDMA used in the experiment.

representatives of 3STSC and the naïve algorithm are captured. The number of clusters ( $k$ ) and the sliding window ( $w$ ) are varied, and the maximum allowance number ( $\alpha$ ) is set to be the number of clusters. In this experiment, two inter-cluster distances of  $k$ -hierarchical clustering, i.e., complete linkage and average linkage functions, and two averaging functions, i.e., CDTW and ICDTW, are utilized. Figures 6.3 and 6.4 show the computational time of between 3STSC and the naïve 2STSC algorithm when  $k = 3$  and  $w = 64$ . The complete results are provided in Appendix G.

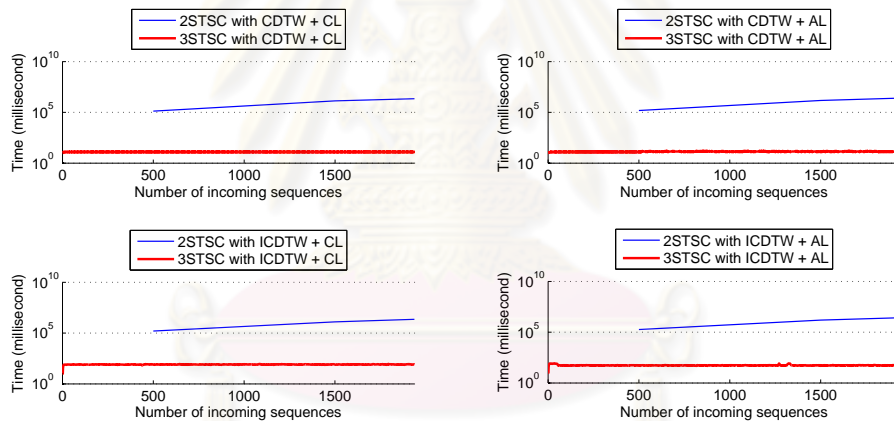


Figure 6.3: Computational time of 3STSC and 2STSC of Buoy1 when a new incoming sequence arrives.

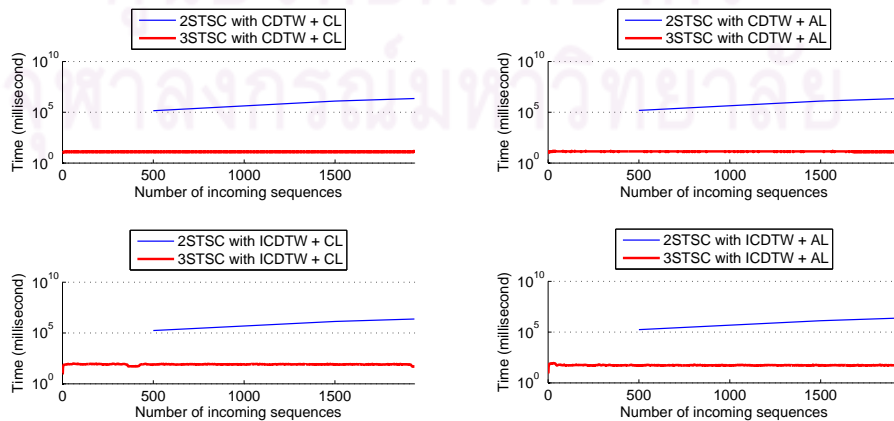


Figure 6.4: Computational time of 3STSC and 2STSC of CBF when a new incoming sequence arrives.

### 6.3.2 Second Experiment

The second experiment shows the quality of clustering results generated from 3STSC, when the maximum allowance number ( $\alpha$ ) is varied; the quality of clustering results increases when there is availability of computational power and storage. However, the quality of clustering results is a tradeoff to clustering time; the number of clusters ( $k$ ) and the sliding window ( $w$ ) are varied, and the maximum allowance number ( $\alpha$ ) are also varied to show speedup and clustering quality. The clustering quality is measured by Shape-based Meaningfulness Measurement (SMM) proposed in Chapter 4, which can be calculated from the following equation.

$$SMM(S, C) = \frac{|\mathcal{S}| \cdot w}{\sum_{i=1}^{|\mathcal{S}|} \min(Distance(S_i, R_j), \forall R_j \in \mathbb{R})} \quad (6.3)$$

where  $Distance(S_i, R_j)$  is a DTW distance between two sequences  $S_i$  and  $R_j$ .

SMM ranges from zero to positive infinity and is a relative value that SMM must be compared between two algorithms at the same set of parameters to identify that with a given dataset which subsequence clustering algorithm produces more meaningful clustering results.

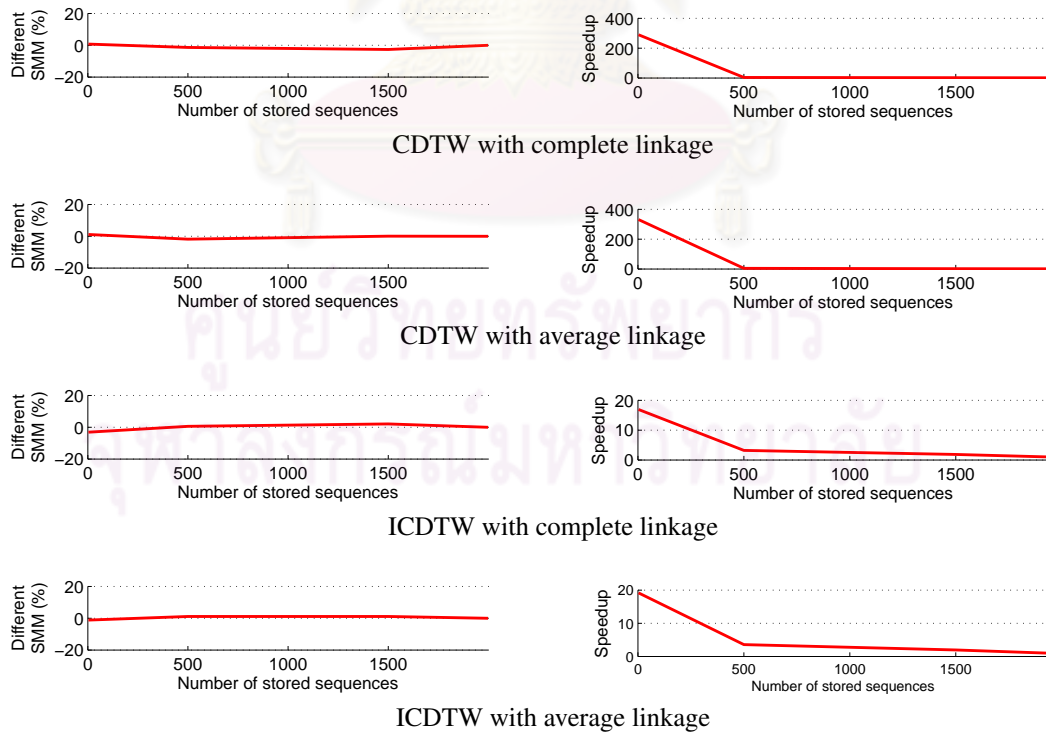


Figure 6.5: Percentage difference of SMM and speedup of 3STSC of Buoy1 when  $k = 3$ ,  $w = 64$ , and number of stored sequences are varied.

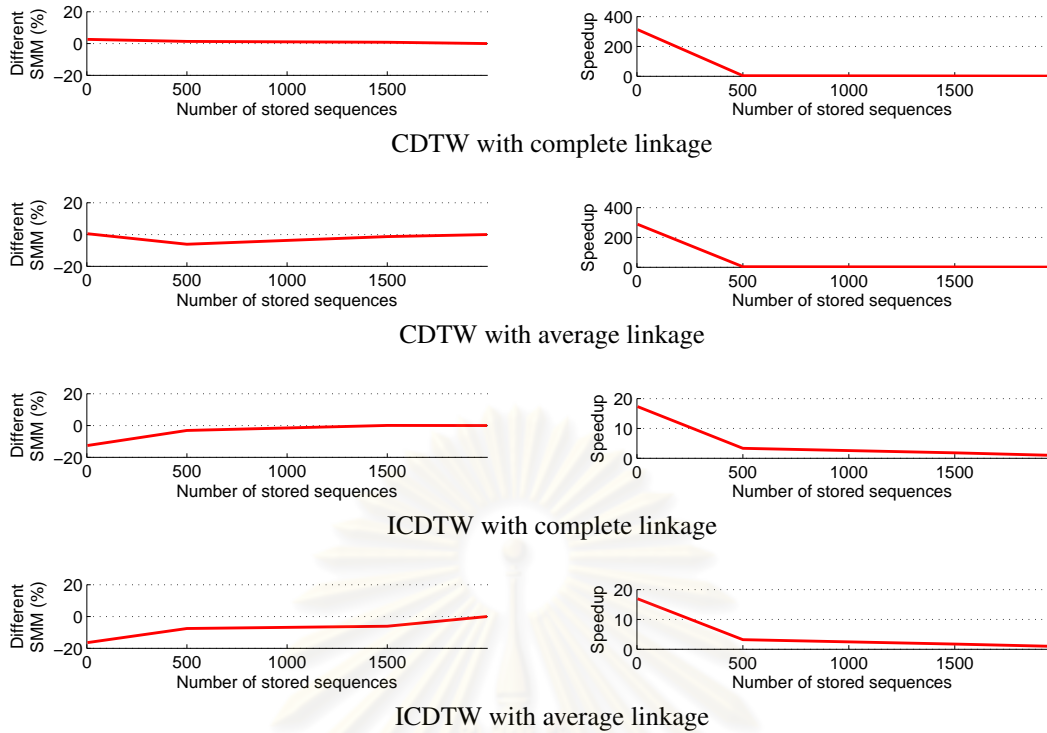


Figure 6.6: Percentage difference of SMM and speedup of 3STSC of CBF when  $k = 3$ ,  $w = 64$ , and number of stored sequences are varied.

In this experiment, two inter-cluster distances of  $k$ -hierarchical clustering, i.e., complete linkage and average linkage functions, and two averaging functions, i.e., CDTW and ICDTW, are utilized. Figures 6.5 and 6.6 show SMM difference and the computational time of 3STSC of Buoy1 and CBF when inter-cluster distance and averaging function are varied. From the experiment results, SMM of both 3STSC and 2STSC are similar, which means 3STSC produces meaningful cluster representatives, while 3STSC can increase calculation speedup by 400 times.

#### 6.4 Conclusion

In this chapter, Streaming Shape-based Subsequence Time Series Clustering (3STSC) is proposed to return a clustering result in real time, where the calculation complexity is constant to the number of previous subsequences. 3STSC is much faster than 2STSC in orders of magnitude, and with availability of computational power and storage, 3STSC returns comparable clustering quality to the naïve algorithm using 2STSC. In addition to 2STSC, 3STSC has the maximum allowance number of stored sequences to calculate a clustering result on this set of stored sequences, where the maximum allowance number is much smaller than the number of previous subsequences. 3STSC utilizes the updating algorithm of stored subsequences from the Incremental Shape-based Averaging, and  $k$ -hierarchical clustering with Dynamic Time Warping (DTW) distance and Shape-based Averaging as a distance measure and an averaging function,

respectively, where two inter-cluster distances, i.e., complete linkage and average linkage, and two averaging functions, i.e., CDTW and ICDTW, are utilized in 3STSC. 3STSC is considered the first streaming subsequence clustering that returns meaningful clustering results.



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



## CHAPTER VII

### CONCLUSION

In this thesis, Shape-based Streaming Subsequence Time Series Clustering (3STSC) is proposed to return clustering results in constant time when a new data point arrives in time series data stream. In addition, 3STSC is extended from the proposed Shape-based Subsequence Time Series Clustering (2STSC) which produces more meaningful clustering results in terms of Shape-based Meaningfulness Measurement (SMM). To make 2STSC produce meaningful results, 2STSC utilizes Dynamic Time Warping (DTW) distance measure and Shape-based Averaging function. An intuitive idea is that DTW distance and Shape-based Averaging can handle a set of trivial-matched subsequences which are contiguous subsequences that have small differences because of time shift. Therefore, DTW distance aligns subsequences to find the optimal warping path between two sequences before distance calculation and Shape-based Averaging aligns subsequences to find an optimal warping path between two sequences before averaging. DTW distance and Shape-based Averaging are superior to the Euclidean distance and Amplitude Averaging used in Subsequence Time Series Clustering (STSC) in that Euclidean distance cannot capture the similarity between two subsequences, and Amplitude Averaging cannot preserve characteristics for producing averaged result. In other words, Euclidean distance in clustering algorithm can lead to incorrect grouping of trivial-matched subsequences, and Amplitude Averaging can lead to undesirable smoothing of trivial-matched subsequences. STSC has been proven as meaningless both theoretically and empirically that STSC will always produce sine waves as cluster representatives regardless of input sequences, where these sine waves are unusable. Therefore, in this thesis, 2STSC is proposed to overcome this problem, and then 3STSC is then proposed to support data streams.

This thesis can be extended to improve the performance further in many data mining tasks. Shape-based Averaging and Incremental Shape-based Averaging can be extended to be used in template matching problem and classification. 2STSC and 3STSC can be used as a preprocessing or a subroutine of many data mining tasks such as association rules, classification, pattern discovery, and visualization.

To improve the algorithms proposed in this thesis, a new methodology of sequence alignment and re-sampling technique can be designed for Shape-based Averaging algorithm, and the averaging scheme can be modified to find the optimal averaging result. Incremental Shape-based

Averaging can be improved by adding decremental algorithm so that the characteristics of an averaged result can be removed by a specific sequence. In addition, 2STSC can be improved by speeding up an algorithm and utilizing other clustering algorithm and removing user-defined parameters that are the number of clusters and the length of sliding window. For 3STSC, other than the number of clusters and the length of sliding window that should be removed, the update algorithm of stored subsequences should be improved to reduce distortion of meaningfulness of clustering results.



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

## CHAPTER VIII

### PUBLICATIONS

1. Vit Niennattrakul, Dararat Srisai, and Chotirat Ann Ratanamahatana. 2010. Shape-based Template Matching for Time Series Data. Knowledge-based Systems. (In press).
2. Sura Rodpongpun, Vit Niennattrakul, and Chotirat Ann Ratanamahatana. 2010. Efficient Subsequence Search on Streaming Data Based on Time Warping Distance. ECTI Transaction CIT 5,1:1-7.
3. Pawan Nunthanid, Vit Niennattrakul, and Chotirat Ann Ratanamahatana. 2011. Discovery of Variable-Length Time Series Motif. In Proceedings of the 8<sup>th</sup> International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON'11), Khon Kean, Thailand.
4. Warissara Meesrikamolkul, Vit Niennattrakul, and Chotirat Ann Ratanamahatana. 2011. Multiple Shape-based Templates Matching for Time Series Data. In Proceedings of the 8<sup>th</sup> International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON'11), Khon Kean, Thailand.
5. Vit Niennattrakul, Pongsakorn Ruengrounghirunya, and Chotirat Ann Ratanamahatana. 2010. Exact Indexing for Massive Time Series Databases under Time Warping Distance. Data Mining and Knowledge Discovery 21,3:509-541.
6. Vit Niennattrakul, Chotirat Ann Ratanamahatana, and Eamonn Keogh. 2010. Data Editing Techniques to Allow the Application of Distance-Based Outlier Detection to Streams. In Proceedings of the 10<sup>th</sup> IEEE International Conference on Data Mining (ICDM'10), pp. 947-952, Sydney, Australia.
7. Doruk Sart, Abdullah Mueen, Walid Najjar, Vit Niennattrakul, and Eamonn Keogh. 2010. Accelerating Dynamic Time Warping Subsequence Search with GPUs and FPGAs. In Proceedings of the 10<sup>th</sup> IEEE International Conference on Data Mining (ICDM'10), pp. 1001-1006, Sydney, Australia.
8. Supasate Choochaisri, Vit Niennattrakul, Saran Jenjaturong, Chalermek Intanagonwiwat, and Chotirat Ann Ratanamahatana. 2010. SENVM: Server Environment Monitoring and Controlling System for a Small Data Center using Wireless Sensor Network. In Proceedings

- of the First International Computer Science and Engineering Conference (ICSEC'10), pp. 23-28, Bangkok, Thailand.
9. Vit Niennattrakul, Chotirat Ann Ratanamahatana. 2009. Shape Averaging under Time Warping. In Proceedings of the 6<sup>th</sup> International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON'09), pp. 626-629, Pattaya, Thailand.
  10. Vit Niennattrakul, Dachawut Wanichsan, and Chotirat Ann Ratanamahatana. 2009. Accurate Subsequence Matching on Data Stream under Time Warping Distance. In Proceedings of the 6<sup>th</sup> International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON'09), pp. 752-755, Pattaya, Thailand.
  11. Vit Niennattrakul, Dachawut Wanichsan, and Chotirat Ann Ratanamahatana. 2009. Accurate Subsequence Matching on Data Stream under Time Warping Distance. In Proceedings of Workshops of the 13<sup>th</sup> Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'09), pp. 752-755, Pattaya, Thailand.
  12. Pongsakorn Ruengrounghirunya, Vit Niennattrakul, and Chotirat Ann Ratanamahatana. 2009. Speeding up Similarity Search on Large Time Series Dataset Under Time Warping Distance. In Proceedings of the 13<sup>th</sup> Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'09), pp. 981-988, Bangkok, Thailand.
  13. Vit Niennattrakul and Chotirat Ann Ratanamahatana. 2009. Meaningful Subsequence Matching on Data Stream under Time Warping Distance. In Proceedings of the 13<sup>th</sup> Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'09), pp. 1013-1020, Bangkok, Thailand.
  14. Pongsakorn Ruengrounghirunya, Vit Niennattrakul, and Chotirat Ann Ratanamahatana. 2008. Efficient Similarity Search under Fast Index Structure for Time Series Data, The 12<sup>th</sup> National Computer Science and Engineering Conference (NCSEC'08), pp. 278-285, Pattaya, Thailand. (Best paper award)
  15. Chotirat Ann Ratanamahatana, Eamonn Keogh, and Vit Niennattrakul. 2008. Making Image Retrieval and Classification More Accurate Using Time Series and Learned Constraints. Artificial Intelligence for Maximizing Content Based Image Retrieval. Idea Group Publishing.
  16. Vit Niennattrakul, Chotirat Ann Ratanamahatana. 2007. Inaccuracies of Shape Averaging

- Method Using Dynamic Time Warping for Time Series Data. In Proceedings of the 7<sup>th</sup> International Conference on Computational Science (ICCS'07), pp. 513-520, Beijing, China.
17. Vit Niennattrakul and Chotirat Ann Ratanamahatana. 2007. On Clustering Multimedia Time Series Data Using K-Means and Dynamic Time Warping. In Proceedings of the International Conference on Multimedia and Ubiquitous Engineering (MUE'07), pp. 733-738, Seoul, South Korea.
  18. Vit Niennattrakul and Chotirat Ann Ratanamahatana. 2007. Making Hand Geometry Verification System More Accurate Using Time Series Representation with R-K Band Learning. In Proceedings of the 11<sup>th</sup> National Computer Science and Engineering Conference (NC-SEC'07), pp. 616-623, Bangkok, Thailand.
  19. Vit Niennattrakul, Dachawut Wanichsan, and Chotirat Ann Ratanamahatana. 2007. Hand Geometry Verification Using Time Series Representation. In Proceedings of the 11<sup>th</sup> International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES'07), pp. 824-831, Vietri sul Mare, Italy.
  20. Vit Niennattrakul and Chotirat Ann Ratanamahatana. 2007. Learning DTW Global Constraint for Time Series Classification. In Workshop and Challenge on Time Series Classification (WCTSC) in conjunction with the 13<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'07), San Jose, California, USA.
  21. Vit Niennattrakul and Chotirat Ann Ratanamahatana. 2006. Clustering Multimedia Data Using Time Series. In Proceedings of the International Conference on Hybrid Information Technology (ICHIT'06), pp. 372-379, Cheju Island, South Korea.

## References

- Balda, M. An estimation of the residual life of blades. In Proceedings of Colloquium Dynamics of Machines (DM'99), pp. 11–18, Prague, Czech Republic, 1999.
- Berndt, D. J. and Clifford, J. Using dynamic time warping to find patterns in time series. In Proceedings of the 1994 AAAI Workshop on Knowledge Discovery in Databases (KDD Workshop'94), pp. 359–370, Seattle, WA, USA, 1994.
- B.L.R., D. M. DaISy: Database for the Identification of Systems. [Online]. Available from : <http://www.esat.kuleuven.ac.be/sista/daisy/> , 2010.
- Burden, R. L., Faires, J. D., and Reynolds, A. G. Numerical Analysis. Brooks Cole, 1997.
- Cai, Y. and Ng, R. T. Indexing spatio-temporal trajectories with chebyshev polynomials. In Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'04), pp. 599–610, Paris, France, 2004.
- Chen, J. Useful clustering outcomes from meaningful time series clustering. In Proceedings of the 6<sup>th</sup> Australasian Data Mining Conference on Data Mining and Analytics 2007 (AusDM'07), pp. 101–109, Gold Coast, Queensland, Australia, December 3-4 2007a.
- Chen, J. R. Making clustering in delay-vector space meaningful. Knowledge Information System, 11,3:369–385, 2007b.
- Chiu, B., Keogh, E. J., and Lonardi, S. Probabilistic discovery of time series motifs. In Proceedings of the 9<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03), pp. 493–498, Washington, DC, USA, August 24 - 27 2003.
- Cotofrei, P. Statistical temporal rules. In Proceedings of the 15<sup>th</sup> Conference on Computational Statistics (COMPSTAT'02), pp. 24–28, Berlin, Germany, 2002.
- Cotofrei, P. and Stoffel, K. Classification rules + time = temporal rules. In Proceedings of the International Conference on Computational Science (ICCS'02), pp. 572–581, Amsterdam, The Netherlands, April 21-24 2002.
- Dai, B.-R., Huang, J.-W., Yeh, M.-Y., and Chen, M.-S. Adaptive clustering for multiple evolving streams. IEEE Transactions on Knowledge and Data Engineering, 18,9:1166–1180, 2006.

- Das, G., Lin, K.-I., Mannila, H., Renganathan, G., and Smyth, P. Rule discovery from time series. In Proceedings of the 4<sup>th</sup> International Conference on Knowledge Discovery and Data Mining (KDD'98), pp. 16–22, New York City, NY, USA, August 27-31 1998.
- Denton, A. Kernel-density-based clustering of time series subsequences using a continuous random-walk noise model. In Proceedings of the 5<sup>th</sup> IEEE International Conference on Data Mining (ICDM'05), pp. 122–129, Houston, TX, USA, 2005.
- Fletcher, R. A modified marquardt subroutine for nonlinear least squares. Technical report, AERE R6799, Atomic Energy Research Establishment, 1971.
- Fu, T.-C., Chung, F.-L., Ng, V., and Luk, R. Pattern discovery from stock time series using self-organizing maps. In Workshop Notes of KDD'01 Workshop on Temporal Data Mining, pp. 27–37, 2001.
- Fu, T.-C., Chung, F.-L., Luk, R. W. P., and Ng, C.man . Preventing meaningless stock time series pattern discovery by changing perceptually important point detection. In Proceedings of Second International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'05), pp. 1171–1174, Changsha, China, August 27-29 2005.
- Fujimaki, R., Hirose, S., and Nakata, T. Theoretical analysis of subsequence time-series clustering from a frequency-analysis viewpoint. In Proceedings of the SIAM International Conference on Data Mining (SDM'08), pp. 506–517, Atlanta, GA, USA, April 24-26 2008.
- Goldin, D. Q., Mardales, R., and Nagy, G. In search of meaning for time series subsequence clustering: matching algorithms based on a new distance measure. In Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management (CIKM'06), pp. 347–356, Arlington, VA, USA, November 6-11 2006.
- Guha, S., Rastogi, R., and Shim, K. ROCK: A robust clustering algorithm for categorical attributes. Information Systems, 25,5:345–366, 2000.
- Gupta, L., Molfese, D. L., and Tammana, R. Nonlinear alignment and averaging for estimating the evoked potential. IEEE Transactions on Biomedical Engineering, 43,4:348–356, April 1996.
- Han, J. and Kamber, M. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.
- Harms, S. K., Deogun, J. S., and Tadesse, T. Discovering sequential association rules with constraints and time lags in multiple sequences. In Proceedings of the 13<sup>th</sup> International Symposium Foundations of Intelligent Systems (ISMIS'02), pp. 432–441, Lyon, France, June 27-29 2002a.

- Harms, S. K., Li, D., Deogun, J. S., and Tadesse, T. Data mining in a geospatial decision support system for drought risk management. In Proceedings of the 3<sup>rd</sup> National Conference on Digital Government Research (DG.O'02), pp. 9 – 16, Los Angeles, CA, USA, May 21 - 23 2002b.
- Hazewinkel, M., editor. Encyclopaedia of Mathematics, chapter Sinusoid. Springer, 2001.
- Hetland, M. L. Temporal rule discovery using genetic programming and specialized hardware. In Proceedings of the 4<sup>th</sup> International Conference on Recent Advances in Soft Computing (RASC'02), pp. 12–13, Nottingham, United Kingdom, 2002.
- Idé, T. Translational symmetry in subsequence time-series clustering. In Proceedings of Conference and Workshops on New Frontiers in Artificial Intelligence (JSAI'06), pp. 5–18, Tokyo, Japan, June 5-9 2006a.
- Idé, T. Why does subsequence time-series clustering produce sine waves? In Proceedings of 10<sup>th</sup> European Conference on Principles and Practice of Knowledge Discovery Knowledge Discovery in Databases (PKDD'06), pp. 211–222, Berlin, Germany, September 18-22 2006b.
- Jin, X., Lu, Y., and Shi, C. Distribution discovery: Local analysis of temporal rules. In Proceedings of the 6<sup>th</sup> Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD'02), pp. 469–480, Taipei, Taiwan, May 6-8 2002a.
- Jin, X., Wang, L., Lu, Y., and Shi, C. Indexing and mining of the local patterns in sequence database. In Proceedings of the Third International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'02), pp. 68–73, Manchester, UK, August 12-14 2002b.
- Karypis, G., Han, E.-H., and Kumar, V. Chameleon: Hierarchical clustering using dynamic modeling. IEEE Computer, 32,8:68–75, 1999.
- Kasetty, S., Stafford, C., Walker, G. P., Wang, X., and Keogh, E. J. Real-time classification of streaming sensor data. In Proceedings of the 20<sup>th</sup> IEEE International Conference on Tools with Artificial Intelligence (ICTAI'08), pp. 149–156, Dayton, OH, USA, November 3-5 2008.
- Kaufman, L. and Rousseeuw, P. J. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley-Interscience, 2005.
- Keogh, E. J. and Folias, T. The UCR Time Series Data Mining Archive. [Online]. Available from : <http://www.cs.ucr.edu/~eamonn/TSDMA/> , 2011.



- Keogh, E. J. and Lin, J. Clustering of time-series subsequences is meaningless: implications for previous and future research. Knowledge and Information Systems, 8,2:154–177, 2005.
- Keogh, E. J. and Ratanamahatana, C. A. Exact indexing of dynamic time warping. Knowledge and Information Systems, 7,3:358–386, 2005.
- Keogh, E. J., Lonardi, S., and Chiu, B. Y.-C. Finding surprising patterns in a time series database in linear time and space. In Proceedings of the 8<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02), pp. 550–556, Edmonton, Alberta, Canada, July 23-26 2002.
- Keogh, E. J., Lin, J., and Truppel, W. Clustering of time series subsequences is meaningless: Implications for previous and future research. In Proceedings of the 3<sup>rd</sup> IEEE International Conference on Data Mining (ICDM'03), pp. 115–122, Melbourne, FL, USA, 19-22 December 2003.
- Keogh, E. J., Palpanas, T., Zordan, V. B., Gunopulos, D., and Cardle, M. Indexing large human-motion databases. In Proceedings of the 13<sup>th</sup> International Conference on Very Large Data Bases (VLDB'04), pp. 780–791, Toronto, Canada, August 31 - September 3 2004.
- Keogh, E. J., Lin, J., and Fu, A. W.-C. Hot SAX: Efficiently finding the most unusual time series subsequence. In Proceedings of the 5<sup>th</sup> IEEE International Conference on Data Mining (ICDM'05), pp. 226–233, Houston, TX, USA, November 27-30 2005.
- Keogh, E. J., Xi, X., Wei, L., and Ratanamahatana, C. A. UCR Time Series Classification/Clustering Page. [Online]. Available from : [http://www.cs.ucr.edu/~eamonn/time\\_series\\_data](http://www.cs.ucr.edu/~eamonn/time_series_data), 2011.
- Kontaki, M., Papadopoulos, A. N., and Manolopoulos, Y. Continuous subspace clustering in streaming time series. Information Systems, 33,2:240–260, 2008.
- Kumar, N., Lolla, V. N., Keogh, E. J., Lonardi, S., and Ratanamahatana, C. A. Time-series bitmaps: A practical visualization tool for working with large time series databases. In Proceedings of the 5<sup>th</sup> SIAM International Conference on Data Mining (SDM'05), pp. 531–535, Newport Beach, CA, USA, 2005.
- Kumar, R. P., Nagabhusan, P., and Chouakria, A. D. Wavesim and adaptive wavesim transform for subsequence time-series clustering. In Proceedings of 9<sup>th</sup> International Conference in Information Technology (ICIT'06), pp. 197–202, Orissa, India, 18-21 December 2006.
- Li, C.-S., Yu, P. S., and Castelli, V. MALM: A framework for mining sequence database at multiple abstraction levels. In Proceedings of the 1998 ACM CIKM International Conference

- on Information and Knowledge Management (CIKM'98), pp. 267–272, Bethesda, MD, USA, November 3-7 1998.
- Lin, J., Keogh, E. J., and Truppel, W. Clustering of streaming time series is meaningless. In Proceedings of the 8<sup>th</sup> ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'03), pp. 56–65, San Diego, CA, USA, June 13 2003.
- Lin, J., Keogh, E. J., Lonardi, S., Lankford, J. P., and Nystrom, D. M. Visually mining and monitoring massive time series. In Proceedings of the 10<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04), pp. 460–469, Seattle, WA, USA, August 22-25 2004a.
- Lin, J., Vlachos, M., Keogh, E. J., and Gunopulos, D. Iterative incremental clustering of time series. In Proceedings of 9<sup>th</sup> International Conference on Extending Database Technology (EDBT'04), pp. 106–122, Crete, Greece, March 14-18 2004b.
- Lloyd, S. P. Least squares quantization in PCM. IEEE Transactions on Information Theory, 28,2: 129–136, 1982.
- MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the 5<sup>th</sup> Berkeley Symposium on Mathematical Statistics and Probability, volume 1, pp. 281–297, San Francisco, CA, USA, 1967.
- Mori, T. and Kuni. Extraction of primitive motion and discovery of association rules from human motion. In Proceedings of the 10<sup>th</sup> IEEE International Workshop on Robot and Human Communication (ROMAN'01), pp. 18–21, Paris, France, 2001.
- Mueen, A. and Keogh, E. J. Online discovery and maintenance of time series motifs. In Proceedings of the 16<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10), pp. 1089–1098, Washington, DC, USA, 2010.
- Mueen, A., Keogh, E. J., Zhu, Q., Cash, S., and Westover, M. B. Exact discovery of time series motifs. In Proceedings of the SIAM International Conference on Data Mining (SDM'09), pp. 473–484, Sparks, NV, USA, April 30 - May 2 2009.
- Niennattrakul, V. and Ratanamahatana, C. A. Inaccuracies of shape averaging method using dynamic time warping for time series data. In Proceedings of 7<sup>th</sup> International Conference on Computational Science (ICCS'07), pp. 513–520, Beijing, China, May 27-30 2007a.
- Niennattrakul, V. and Ratanamahatana, C. A. On clustering multimedia time series data using k-means and dynamic time warping. In Proceedings of the International Conference on Multimedia and Ubiquitous Engineering (MUE'07), pp. 733–738, Seoul, Korea, April 26-28 2007b.

- Niennattrakul, V. and Ratanamahatana, C. A. Meaningful subsequence matching under time warping distance for data stream. In Proceedings of 13<sup>th</sup> Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD'09), pp. 1013–1020, Bangkok, Thailand, 2009.
- Niennattrakul, V. and Ratanamahatana, C. A. Clustering multimedia data using time series. In Proceedings of the International Conference on Hybrid Information Technology (ICHIT'06), pp. 372–379, Cheju Island, Korea, November 9-11 2006.
- Niennattrakul, V. and Ratanamahatana, C. A. Learning DTW global constraint for time series classification. In Time Series Classification Challenges at 13<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, CA, USA, August 12-15 2007c.
- Niennattrakul, V., Wanichsan, D., and Ratanamahatana, C. A. Accurate subsequence matching on data stream under time warping distance. In Proceedings of 13<sup>th</sup> Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD'09) International Workshops, pp. 156–167, Bangkok, Thailand, 2009.
- Niennattrakul, V., Ratanamahatana, C. A., and Keogh, E. Data editing techniques to allow the application of distance-based outlier detection to streams. In Proceedings of the 10<sup>th</sup> IEEE International Conference on Data Mining (ICDM'10), pp. 947–952, Sydney, Australia, 2010a.
- Niennattrakul, V., Ruengronghirunya, P., and Ratanamahatana, C. A. Exact indexing for massive time series databases under time warping distance. Data Mining and Knowledge Discovery, 21,3:509–541, 2010b.
- Osaki, R., Shimada, M., and Uehara, K. A motion recognition method by using primitive motions. In Proceedings of the 5<sup>th</sup> Working Conference on Visual Database Systems (VDB'00), pp. 117–128, Fukuoka, Japan, May 10-12 2000.
- Peker, K. A. Subsequence time series (STS) clustering techniques for meaningful pattern discovery. In Proceedings of the International Conference on Integration of Knowledge Intensive Multi-Agent Systems (KIMAS'05), pp. 360–365, Waltham, MA, USA, 18-21, 2005.
- Protopapas, P., Jimenez, R., and Alcock, C. Fast identification of transits from light-curves. Monthly Notices of the Royal Astronomical Society, 362,2, 2005.
- Radhakrishnan, N., Wilson, J., and Loizou, P. An alternate partitioning technique to quantify the

- regularity of complex time series. International Journal of Bifurcation and Chaos, 10,7: 1773–1780, 2000.
- Ratanamahatana, C. A. and Keogh, E. J. Multimedia retrieval using time series representation and relevance feedback. In Proceedings of 8<sup>th</sup> International Conference on Asian Digital Libraries (ICADL'05), pp. 400–405, Bangkok, Thailand, December 12-15 2005a.
- Ratanamahatana, C. A. and Keogh, E. J. Making time-series classification more accurate using learned constraints. In Proceedings of the Fourth SIAM International Conference on Data Mining (SDM'04), pp. 11–22, Lake Buena Vista, FL, USA, April 22-24 2004.
- Ratanamahatana, C. A. and Keogh, E. J. Three myths about dynamic time warping data mining. In Proceedings of the 5<sup>th</sup> SIAM International Conference on Data Mining (SDM'05), pp. 506–510, Newport Beach, CL, USA, April 21-23 2005b.
- Rodrigues, P. P., Gama, J., and Pedroso, J. P. ODAC: Hierarchical clustering of time series data streams. In Proceedings of the 6<sup>th</sup> SIAM International Conference on Data Mining (SDM'06), Bethesda, MD, USA, April 20-22 2006.
- Rodrigues, P. P., Gama, J., and Pedroso, J. P. Hierarchical clustering of time-series data streams. IEEE Transactions on Knowledge and Data Engineering, 20,5:615–627, 2008.
- Sacchi, L., Larizza, C., Combi, C., and Bellazzi, R. Data mining with temporal abstractions: learning rules from time series. Data Mining and Knowledge Discovery, 15,2:217–247, 2007.
- Saito, N. Local feature extraction and its applications using a library of bases. PhD thesis, Yale University, 1994.
- Sakurai, Y., Yoshikawa, M., and Faloutsos, C. FTW: Fast similarity search under the time warping distance. In Proceedings of the 24<sup>th</sup> ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'05), pp. 326–337, Baltimore, MD, USA, 2005.
- Salvador, S. and Chan, P. Toward accurate dynamic time warping in linear time and space. Intelligent Data Analysis, 11,5:561–580, 2007.
- Sarker, B. K., Mori, T., Hirata, T., and Uehara, K. Parallel algorithms for mining association rules in time series data. In Proceedings of the International Symposium on Parallel and Distributed Processing and Applications (ISPA'03), pp. 273–284, Aizu, Japan, July 2-4 2003.
- Schittenkopf, C., Tino, P., and Dorffner, G. The benefit of information reduction for trading strategies. Report Series for Adaptive Information Systems and Management in Economics and Management, 2000.

- Shieh, J. and Keogh, E. J. SAX: Disk-aware mining and indexing of massive time series datasets. Data Mining and Knowledge Discovery, 19,1:24–57, 2009.
- Simon, G., Lee, J. A., and Verleysen, M. Unfolding preprocessing for meaningful time series clustering. Neural Networks, 19,6-7:877–888, 2006.
- Struzik, Z. R. Time series rule discovery: Tough, not meaningless. In Proceedings of 14<sup>th</sup> International Symposium on Foundations of Intelligent Systems (ISMIS'03), pp. 32–39, Maebashi City, Japan, October 28-31 2003.
- Uehara, K. and Shimada, M. Extraction of primitive motion and discovery of association rules from human motion data. In Progress in Discovery Science, pp. 338–348, 2002.
- Ueno, K., Xi, X., Keogh, E. J., and Lee, D.-J. Anytime classification using the nearest neighbor algorithm with applications to stream mining. In Proceedings of the 6<sup>th</sup> IEEE International Conference on Data Mining (ICDM'06), pp. 623–632, Hong Kong, China, 18-22 December 2006.
- Wan, D., Zhang, Y., and Li, S. Discovery association rules in time series of hydrology. In Proceedings of IEEE International Conference on Integration Technology (ICIT'07), pp. 653 –657, Shenzhen, China, 2007.
- Wikipedia.org. 12 lead ECG. [Online]. Available from : <http://en.wikipedia.org/wiki/Image:12leadECG.jpg> , 2011a.
- Wikipedia.org. Comparison of three stock indices after 1975. [Online]. Available from : [http://en.wikipedia.org/wiki/Image:Comparison\\_of\\_three\\_stock\\_indices\\_after\\_1975.svg](http://en.wikipedia.org/wiki/Image:Comparison_of_three_stock_indices_after_1975.svg) , 2011b.
- Wikipedia.org. Instrumental Temperature Record. [Online]. Available from : [http://en.wikipedia.org/wiki/Image:Instrumental\\_Temperature\\_Record.png](http://en.wikipedia.org/wiki/Image:Instrumental_Temperature_Record.png) , 2011c.
- Wu, W., Au, L., Jordan, B., Stathopoulos, T., Batalin, M., Kaiser, W., Vahdatpour, A., Sarrafzadeh, M., Fang, M., and Chodosh, J. The SmartCane system: An assistive device for geriatrics. In Proceedings of the ICST 3<sup>rd</sup> International Conference on Body Area Networks (BodyNets'08), pp. 2:1–2:4, Brussels, Belgium, March 13-15 2008.
- Yairi, T., Kato, Y., and Hori, K. Fault detection by mining association rules from house-keeping data. In Proceedings of the 4<sup>th</sup> International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS 2001), p. 555, Montreal, CA, USA, June 19-21 2001.

- Yankov, D. and Keogh, E. J. Manifold clustering of shapes. In Proceedings of the 6<sup>th</sup> IEEE International Conference on Data Mining (ICDM'06), pp. 1167–1171, Hong Kong, China, December 18-22 2006.
- Yankov, D., Keogh, E. J., Medina, J., Chiu, B., and Zordan, V. B. Detecting time series motifs under uniform scaling. In Proceedings of the 13<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07), pp. 844–853, San Jose, CA, USA, 2007.
- Yankov, D., Keogh, E. J., and Rebbapragada, U. Disk aware discord discovery: Finding unusual time series in terabyte sized datasets. Knowledge and Information Systems, 17,2:241–262, 2008a.
- Yankov, D., Keogh, E. J., Wei, L., Xi, X., and Hodges, W. L. Fast best-match shape searching in rotation-invariant metric spaces. IEEE Transactions on Multimedia, 10,2:230–239, 2008b.
- Ye, L. and Keogh, E. J. Time series shapelets: A novel technique that allows accurate, interpretable and fast classification. Data Mining and Knowledge Discovery, 22,1-2:149–182, 2011.
- Zhang, T., Ramakrishnan, R., and Livny, M. BIRCH: An efficient data clustering method for very large databases. In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (SIGMOD'96), pp. 103–114, Montreal, Quebec, Canada, June 4-6 1996.



APPENDICES

ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

# APPENDIX A

## DATASETS



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



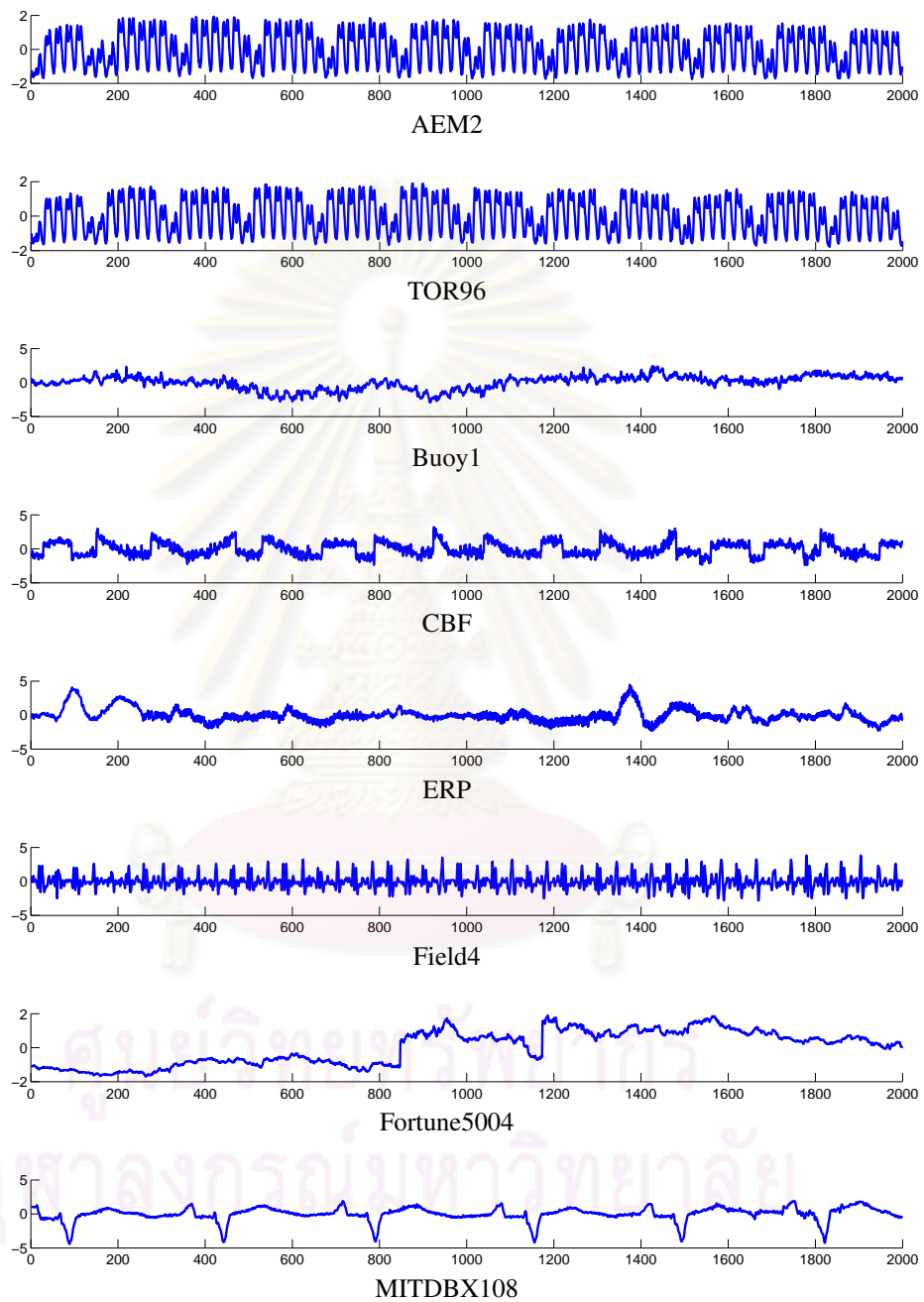


Figure A.1: Datasets from TSDMA used in the experiments of Chapters 2, 4, and 6.

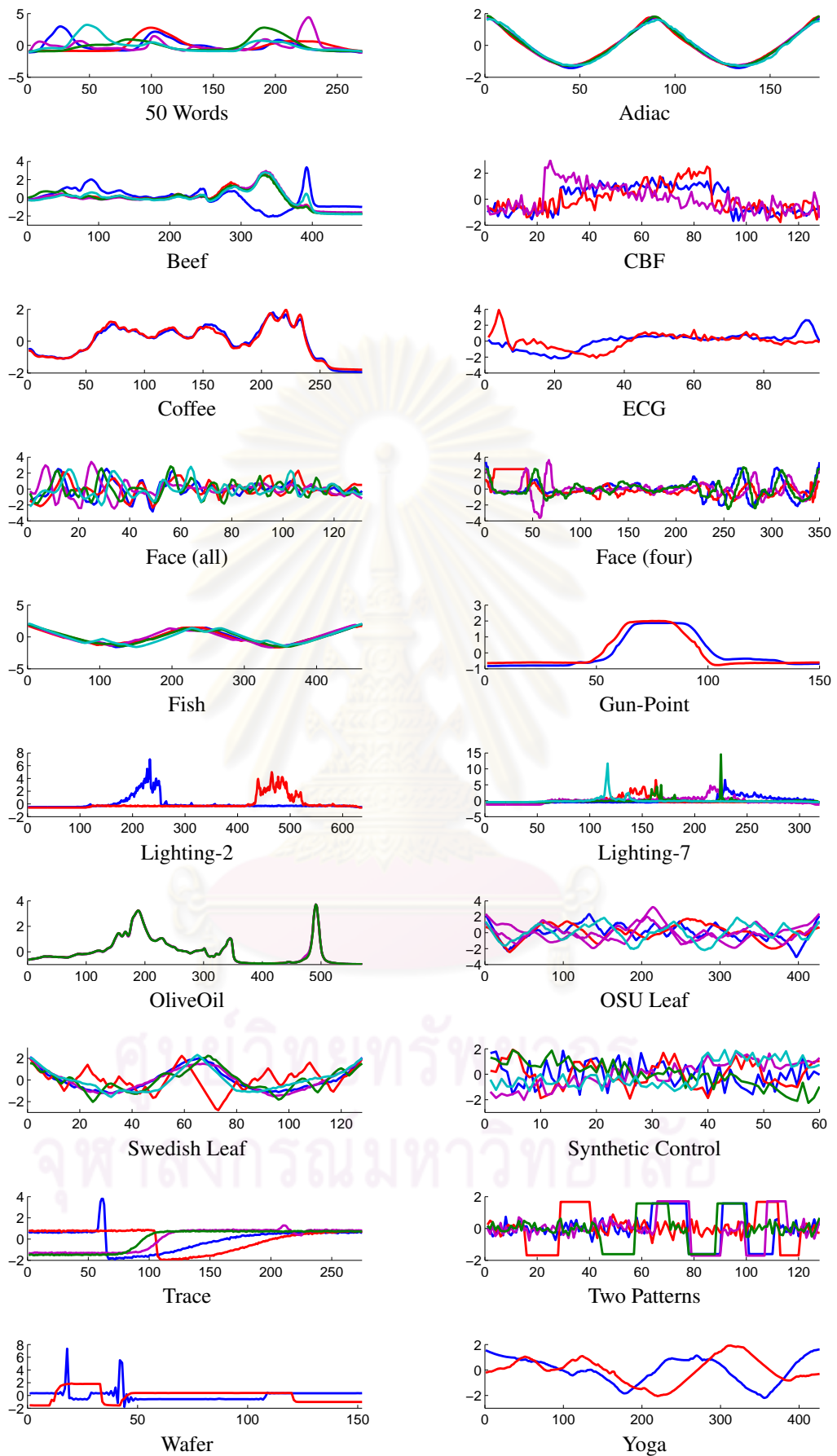


Figure A.2: Examples of some classes of the UCR classification/clustering datasets used in Chapters 3 and 5.

Table A.1: Details of the UCR classification/clustering datasets used in Chapters 3 and 5

Dataset	Number of classes	Length	Size of datasets
50words	50	270	905
Adiac	37	176	781
Beef	5	470	60
CBF	3	128	930
Coffee	2	286	56
ECG	2	96	200
Face (all)	14	131	2250
Face (four)	4	350	112
Fish	7	463	350
Gun-Point	2	150	200
Lighting-2	2	637	121
Lighting-7	7	319	143
Oliveoil	4	570	60
OSULeaf	6	427	442
SwedishLeaf	15	128	1125
Synthetic	6	60	600
Trace	4	275	200
TwoPatterns	4	128	5000
Wafer	2	152	7174
Yoga	2	426	3300

## APPENDIX B

### COMPLETE EXPERIMENTAL RESULTS OF THE FIRST EXPERIMENT IN CHAPTER 2



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

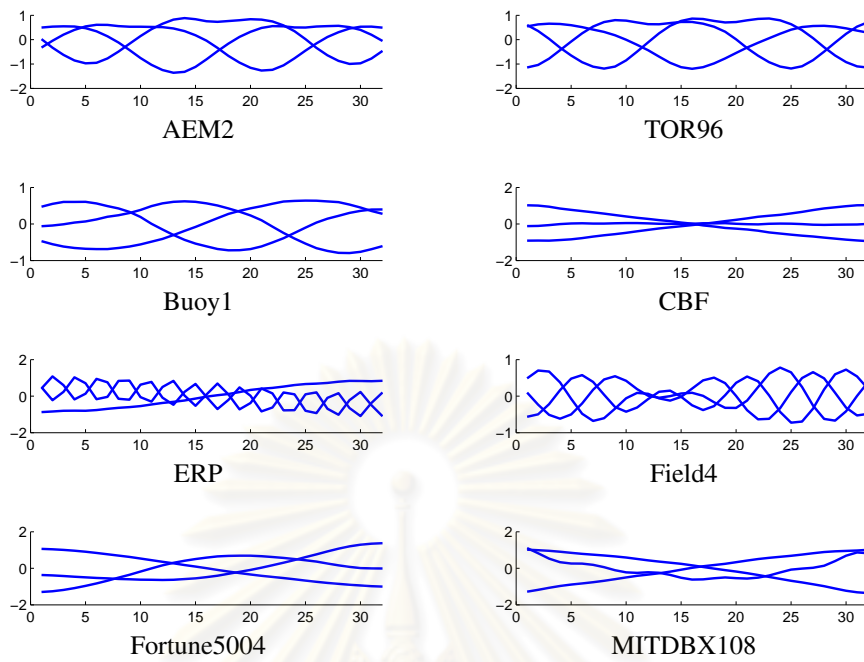


Figure B.1: Cluster representatives generated from STSC using  $k$ -means clustering when  $k = 3$  and  $w = 32$ .

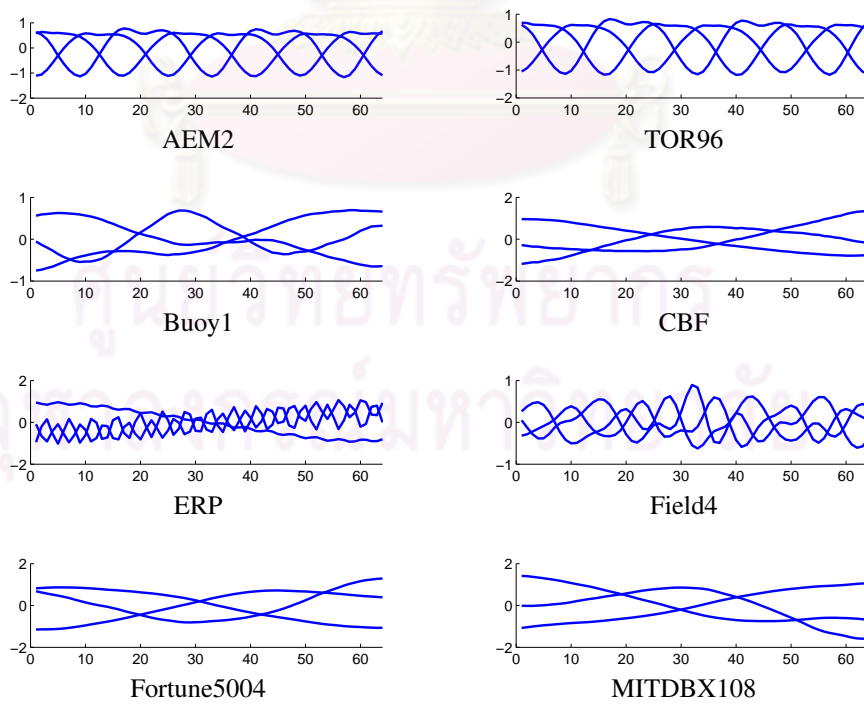


Figure B.2: Cluster representatives generated from STSC using  $k$ -means clustering when  $k = 3$  and  $w = 64$ .

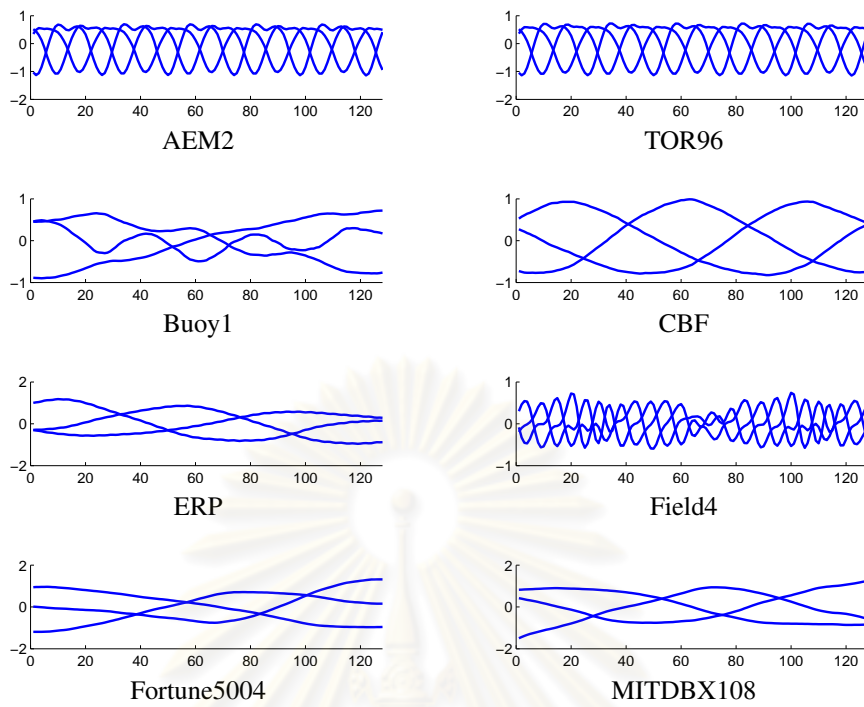


Figure B.3: Cluster representatives generated from STSC using  $k$ -means clustering when  $k = 3$  and  $w = 128$ .

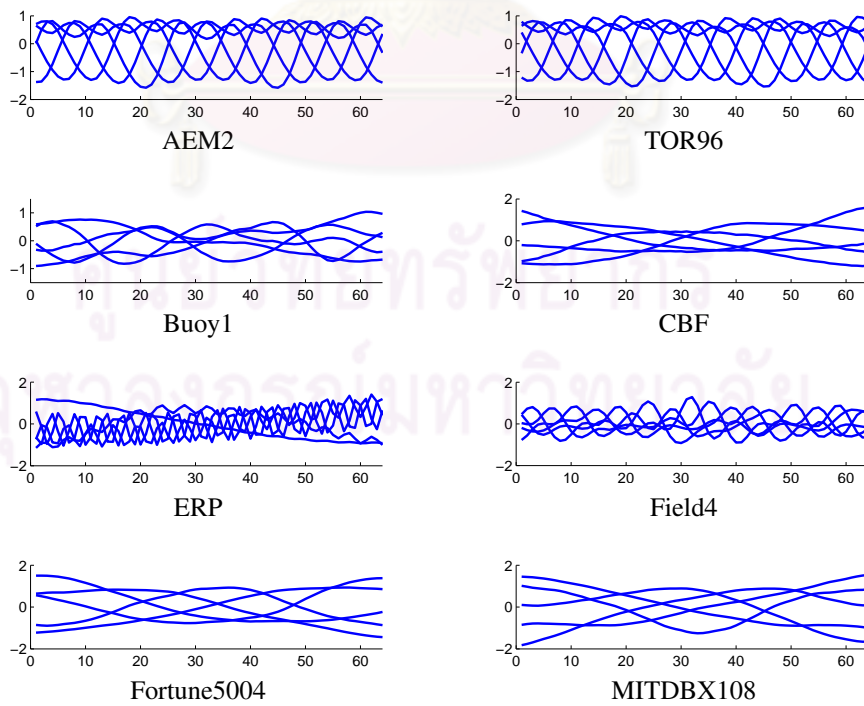


Figure B.4: Cluster representatives generated from STSC using  $k$ -means clustering when  $k = 5$  and  $w = 64$ .

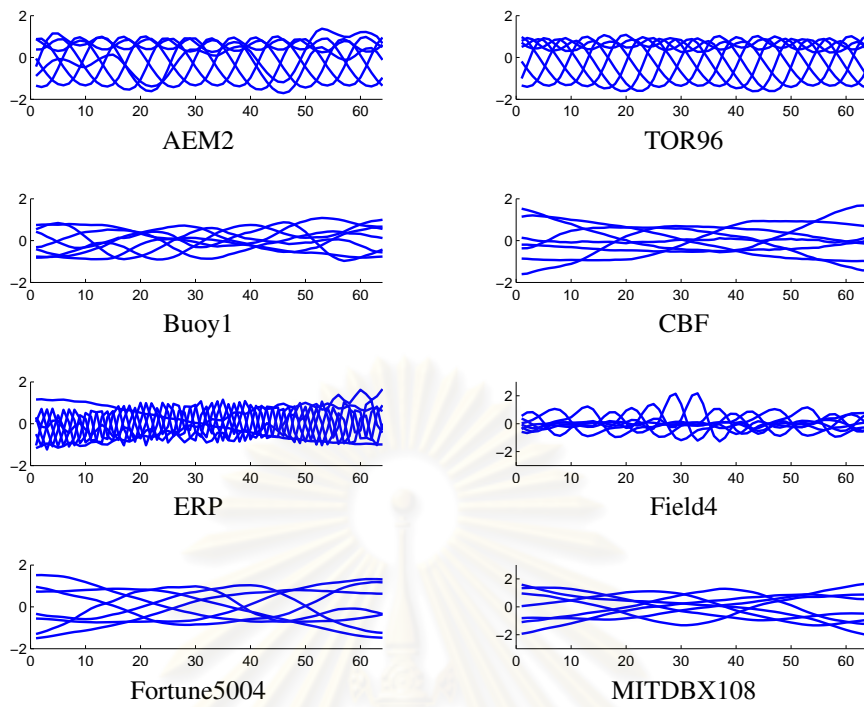


Figure B.5: Cluster representatives generated from STSC using  $k$ -means clustering when  $k = 7$  and  $w = 64$ .

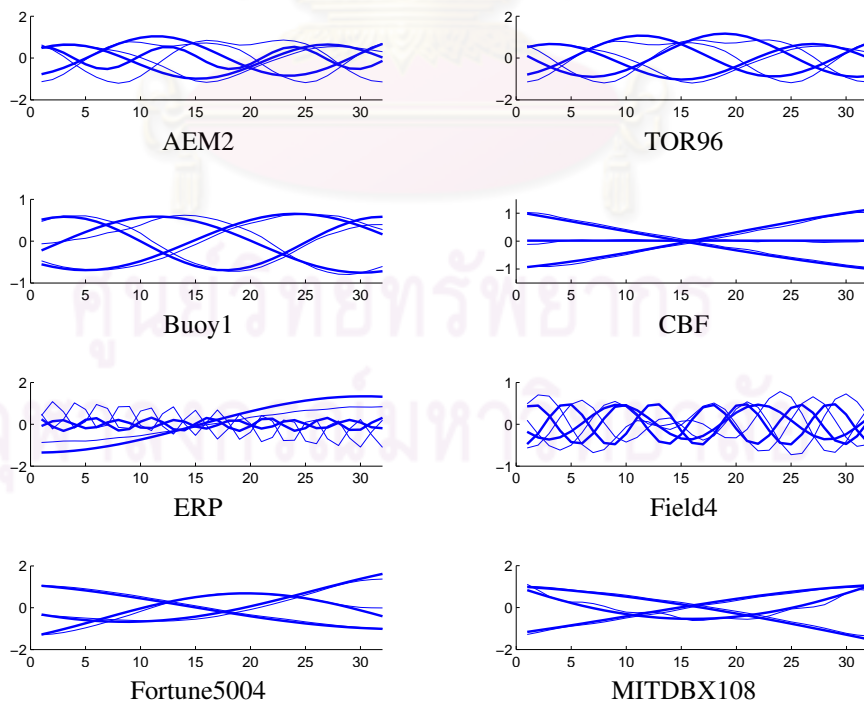


Figure B.6: Constructed sine waves generated from STSC using  $k$ -means clustering when  $k = 3$  and  $w = 32$ .

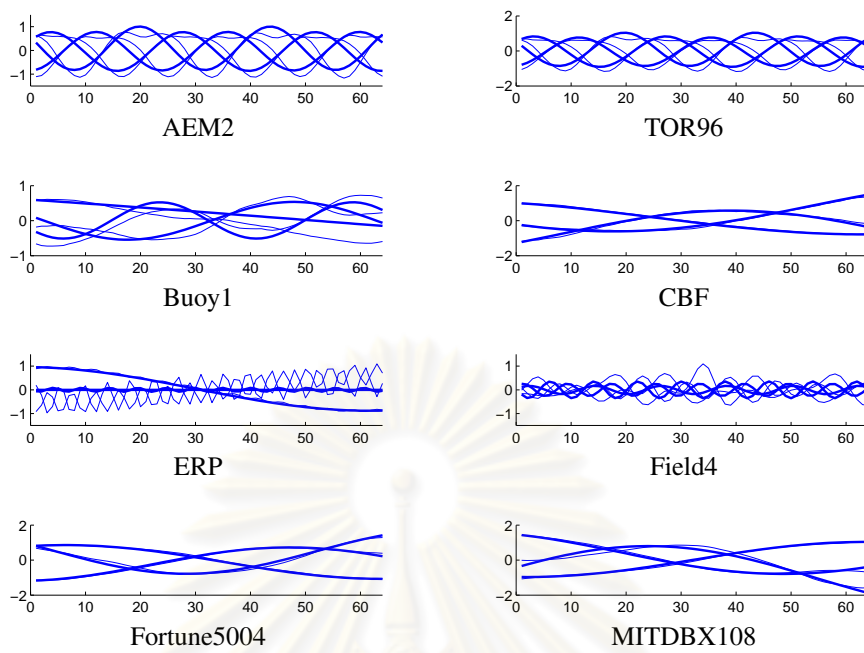


Figure B.7: Constructed sine waves generated from STSC using  $k$ -means clustering when  $k = 3$  and  $w = 64$ .

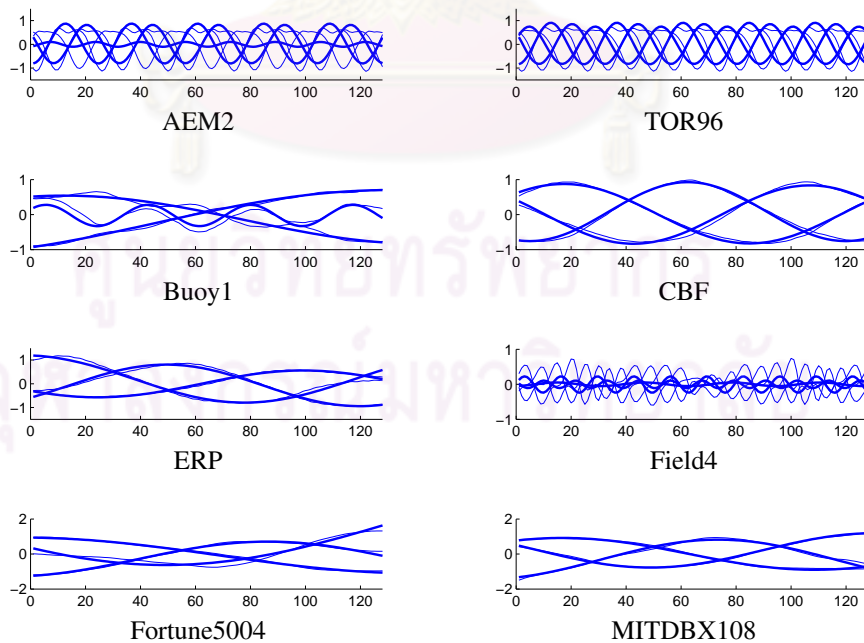


Figure B.8: Constructed sine waves generated from STSC using  $k$ -means clustering when  $k = 3$  and  $w = 128$ .



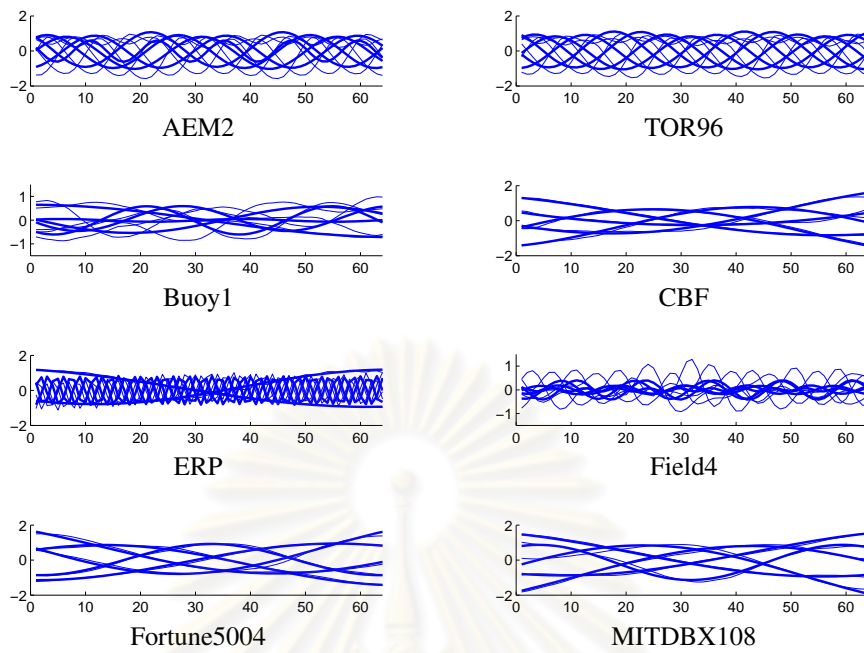


Figure B.9: Constructed sine waves generated from STSC using  $k$ -means clustering when  $k = 5$  and  $w = 64$ .

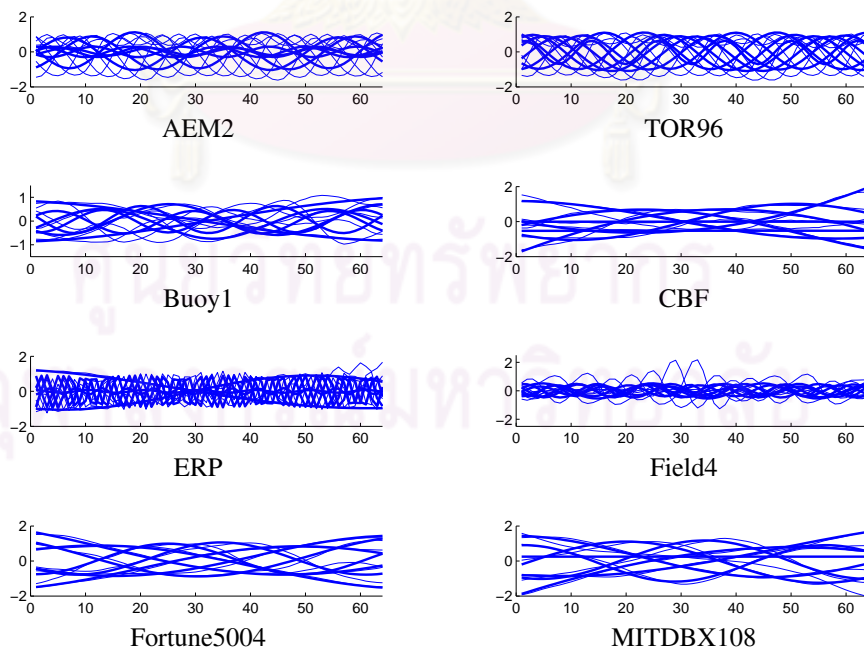


Figure B.10: Constructed sine waves generated from STSC using  $k$ -means clustering when  $k = 7$  and  $w = 64$ .

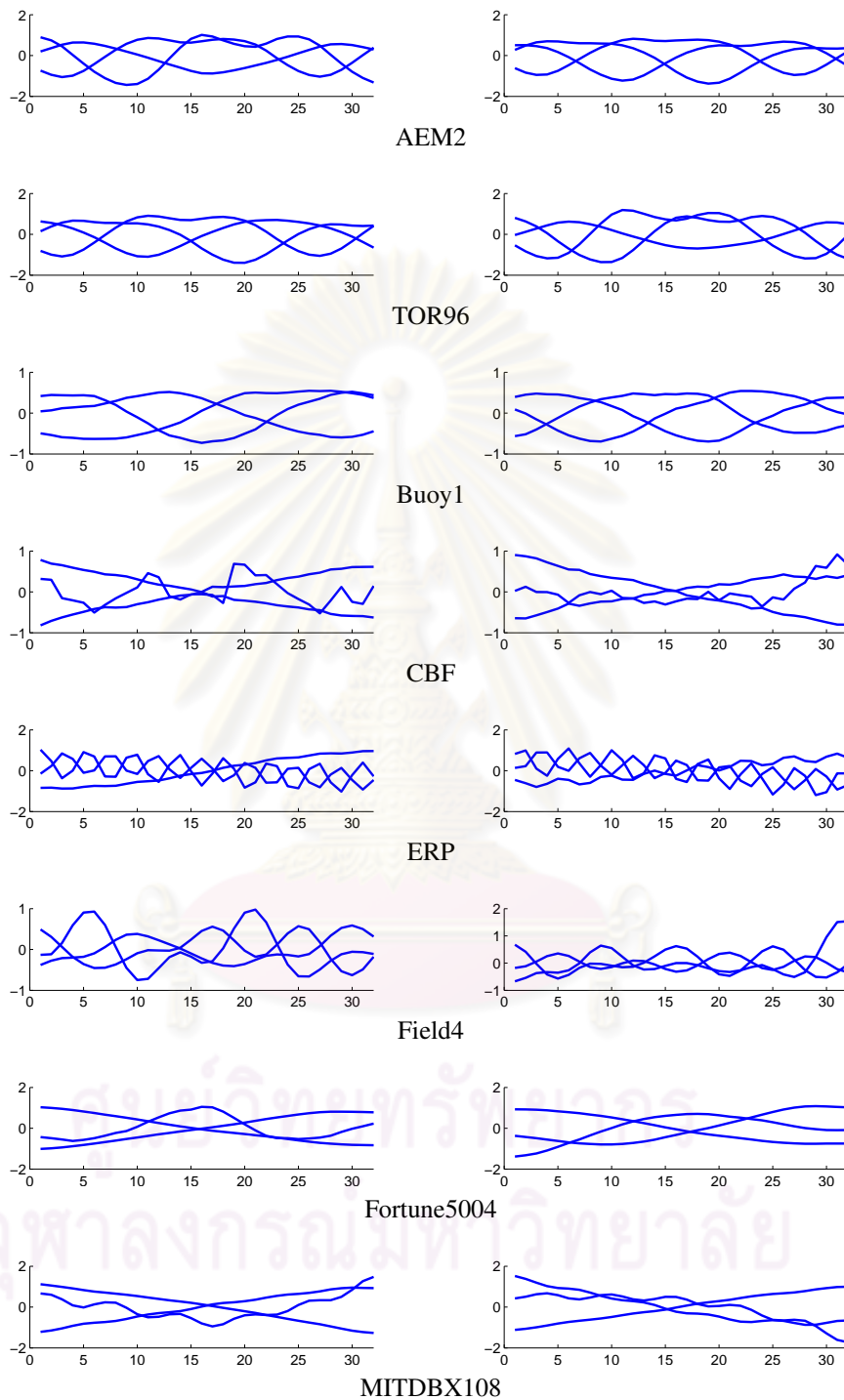


Figure B.11: Cluster representatives generated from STSC using  $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when  $k = 3$  and  $w = 32$ .

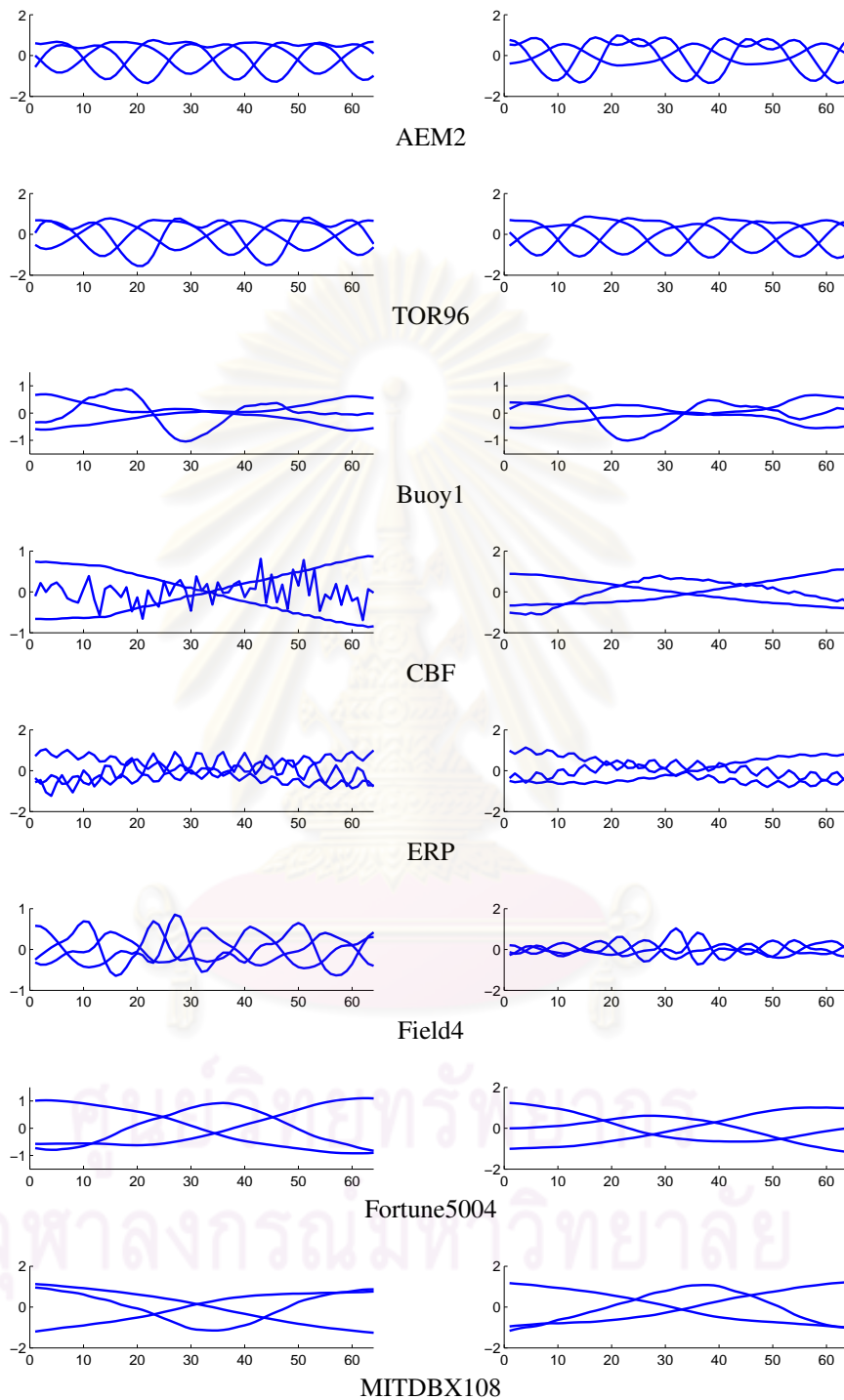


Figure B.12: Cluster representatives generated from STSC using  $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when  $k = 3$  and  $w = 64$ .

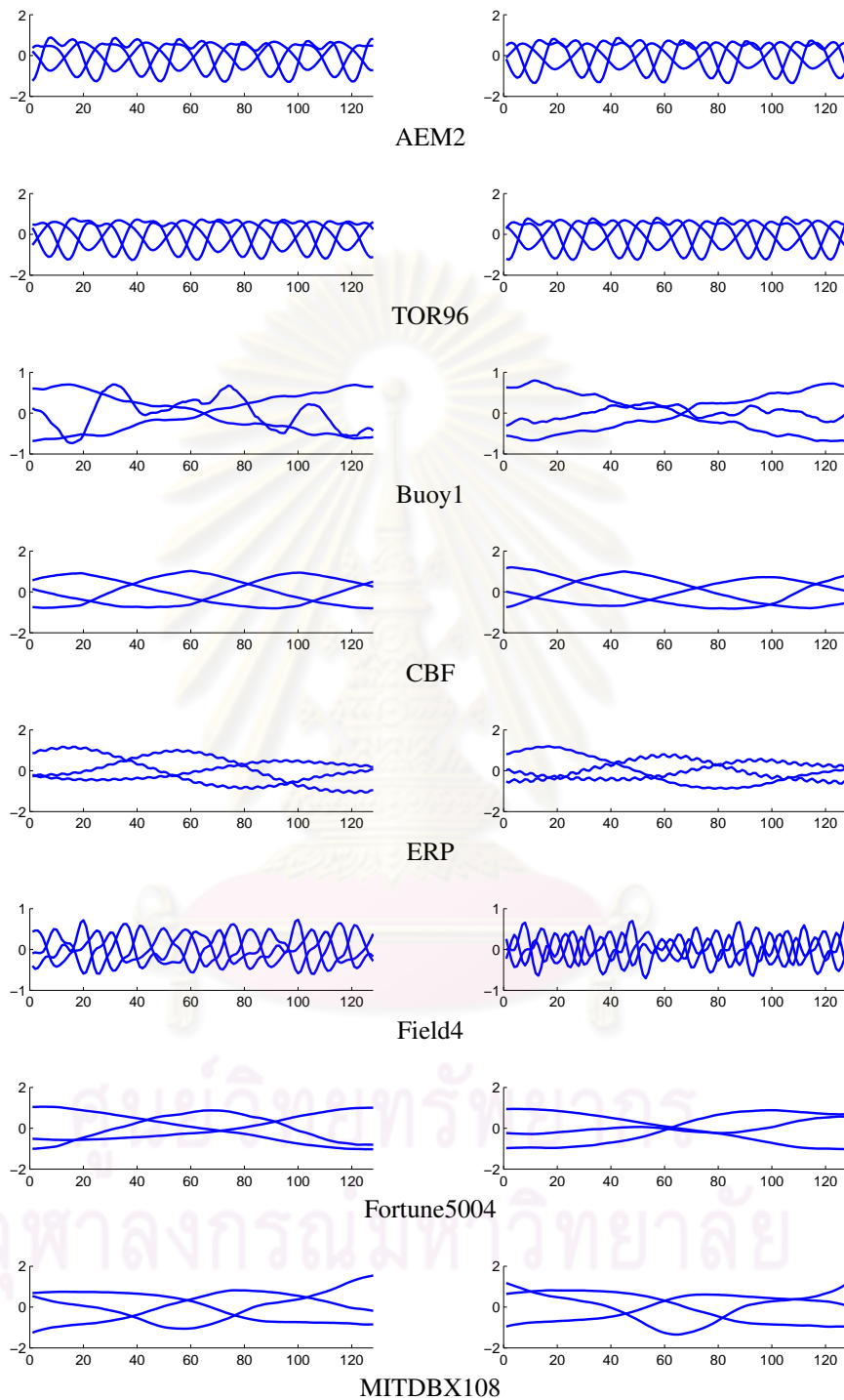


Figure B.13: Cluster representatives generated from STSC using  $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when  $k = 3$  and  $w = 128$ .

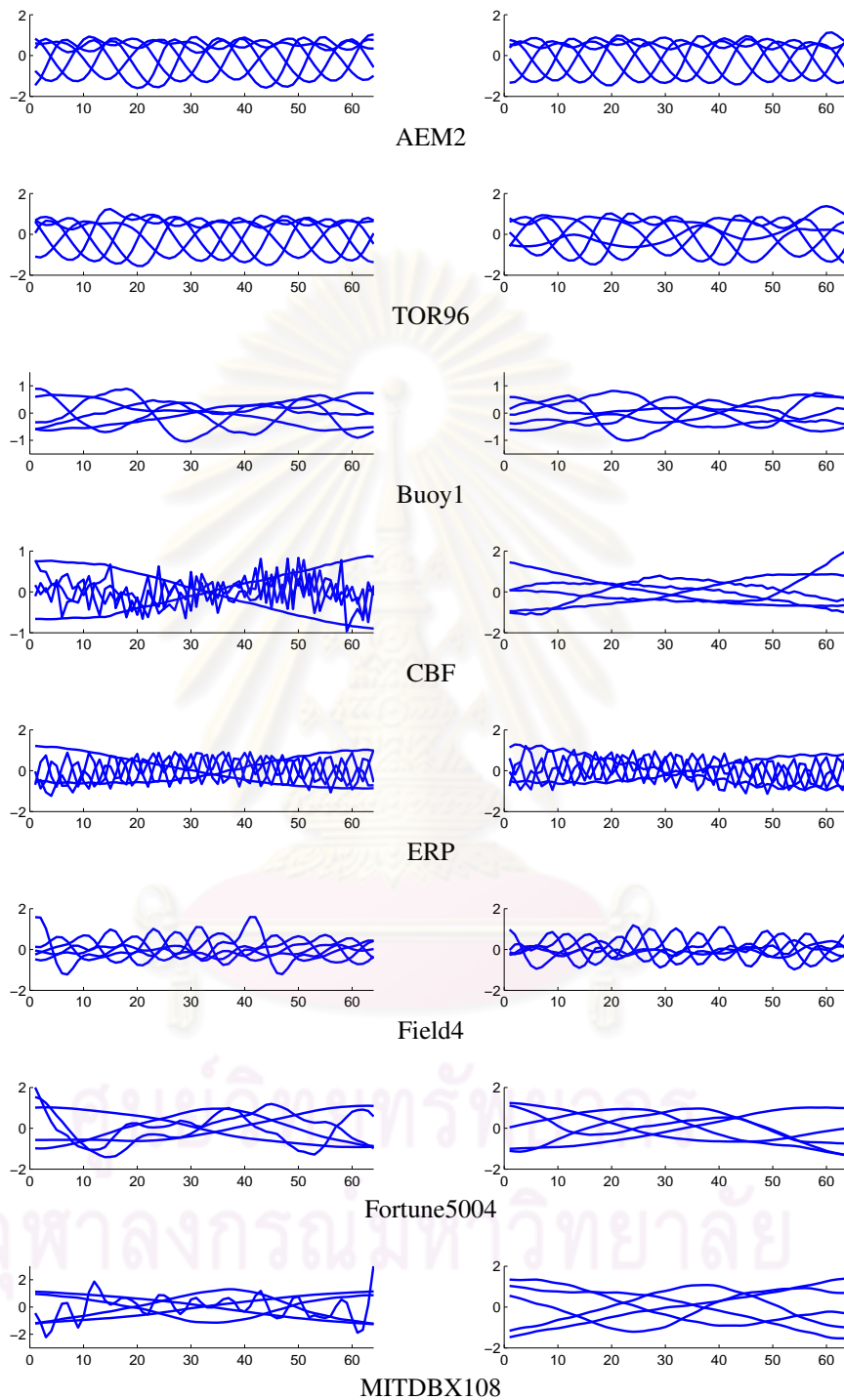


Figure B.14: Cluster representatives generated from STSC using  $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when  $k = 5$  and  $w = 64$ .

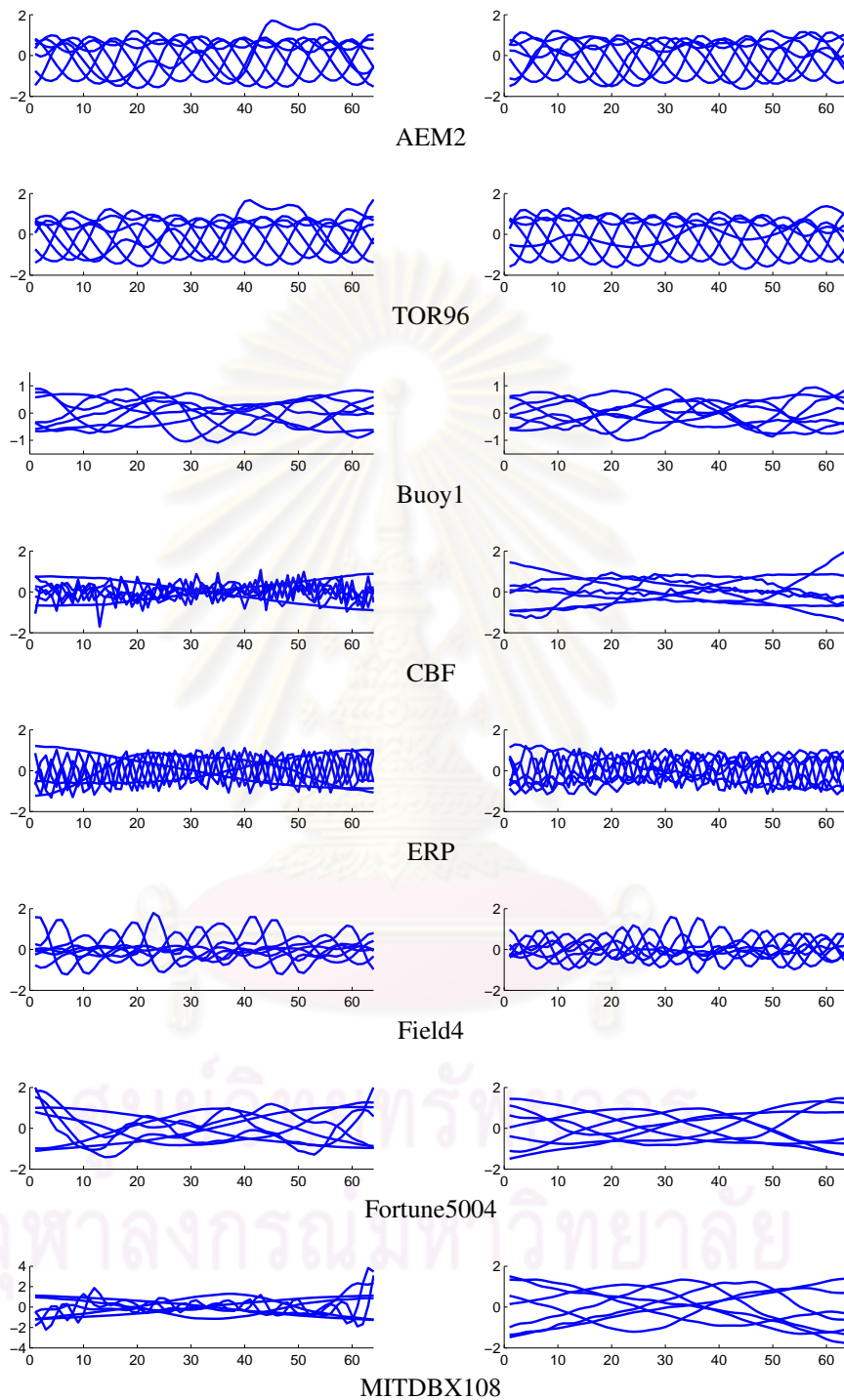


Figure B.15: Cluster representatives generated from STSC using  $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when  $k = 7$  and  $w = 64$ .

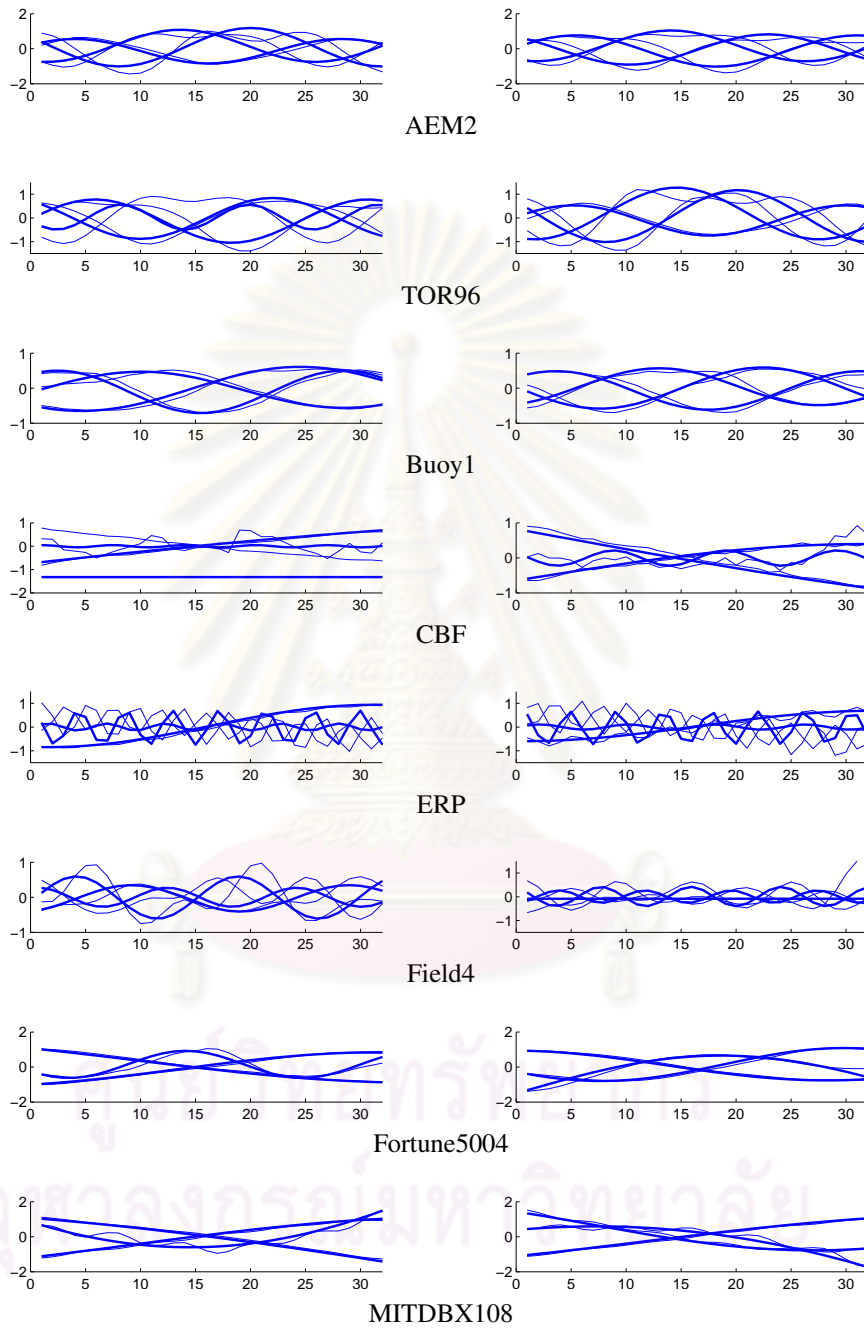


Figure B.16: Constructed sine waves generated from STSC using  $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when  $k = 3$  and  $w = 32$ .

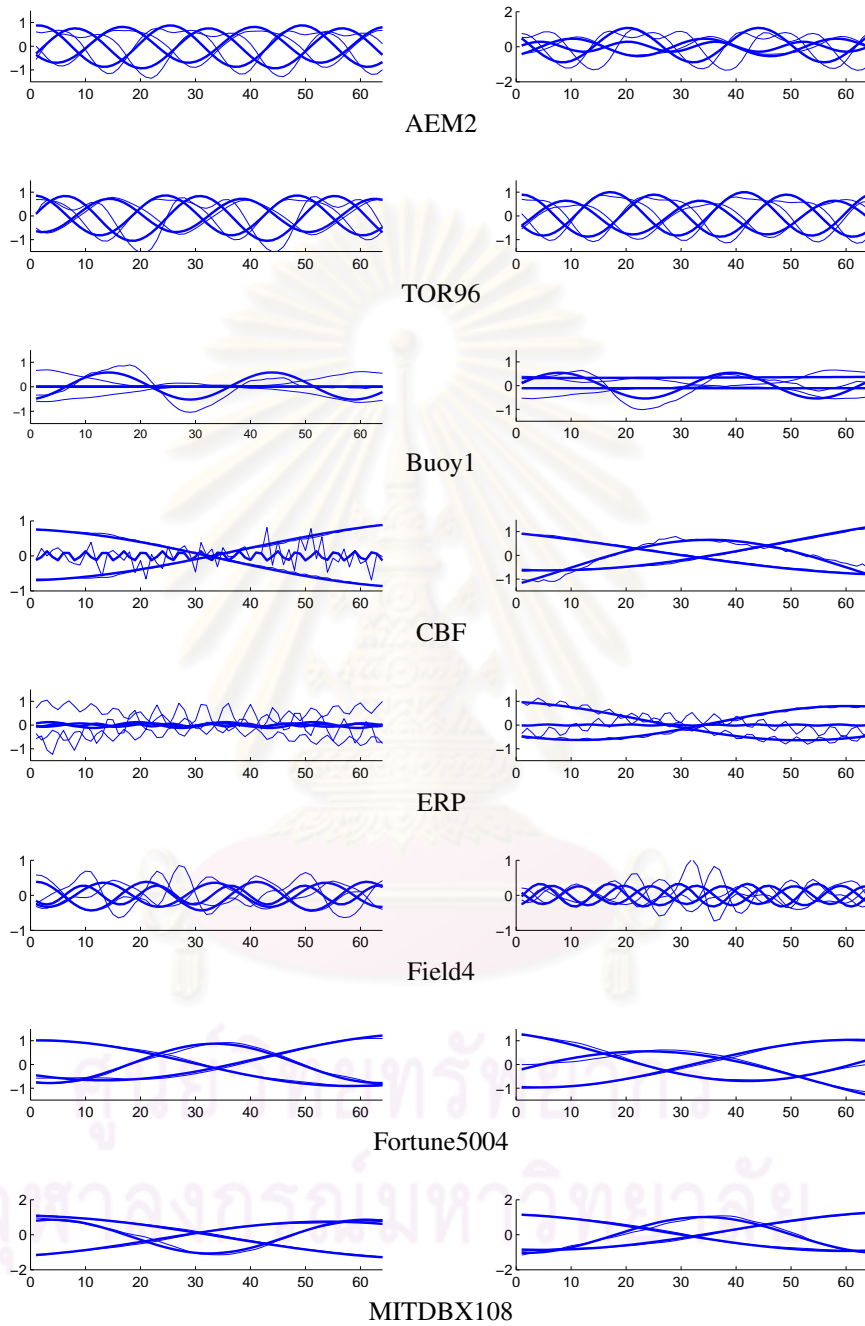


Figure B.17: Constructed sine waves generated from STSC using  $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when  $k = 3$  and  $w = 64$ .



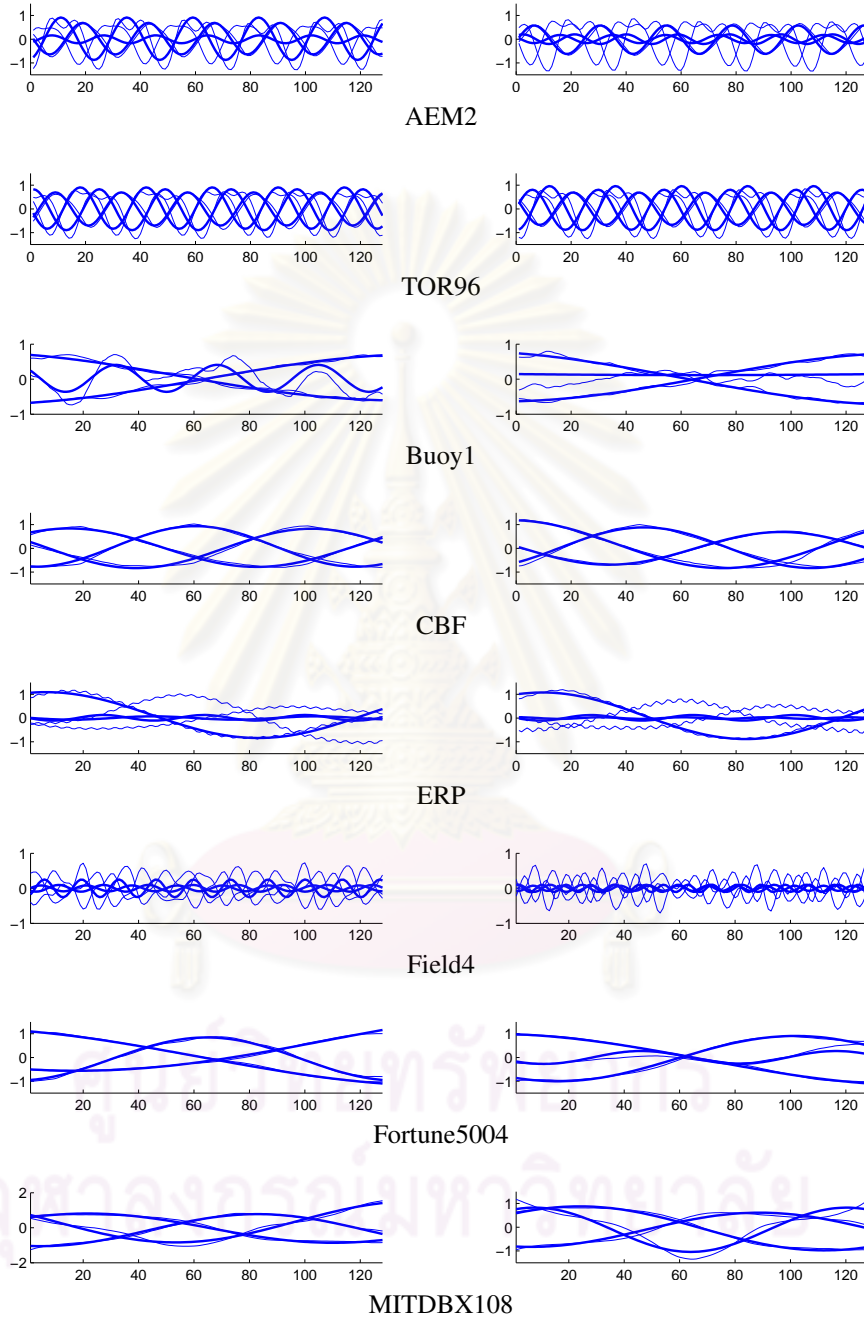


Figure B.18: Constructed sine waves generated from STSC using  $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when  $k = 3$  and  $w = 128$ .

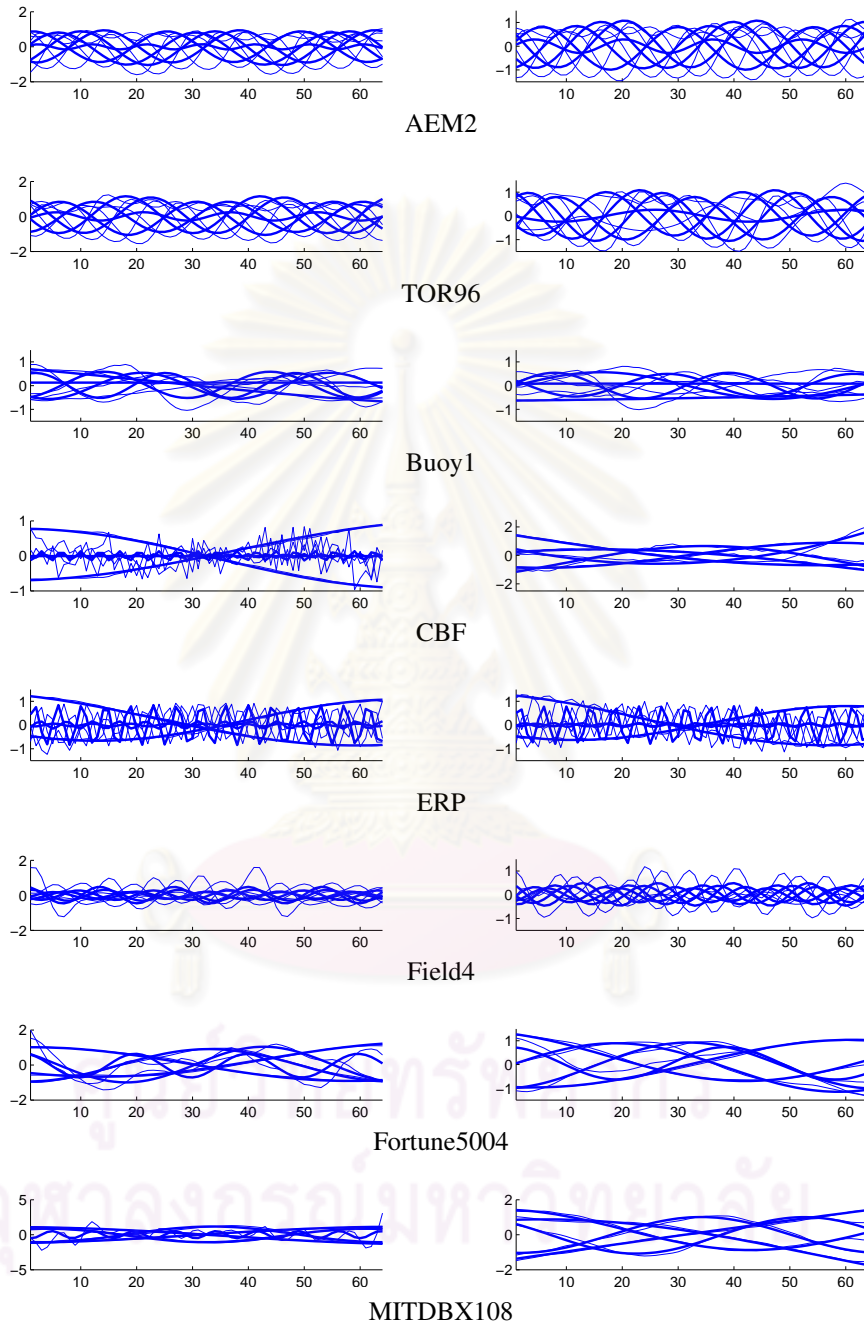


Figure B.19: Constructed sine waves generated from STSC using  $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when  $k = 5$  and  $w = 64$ .

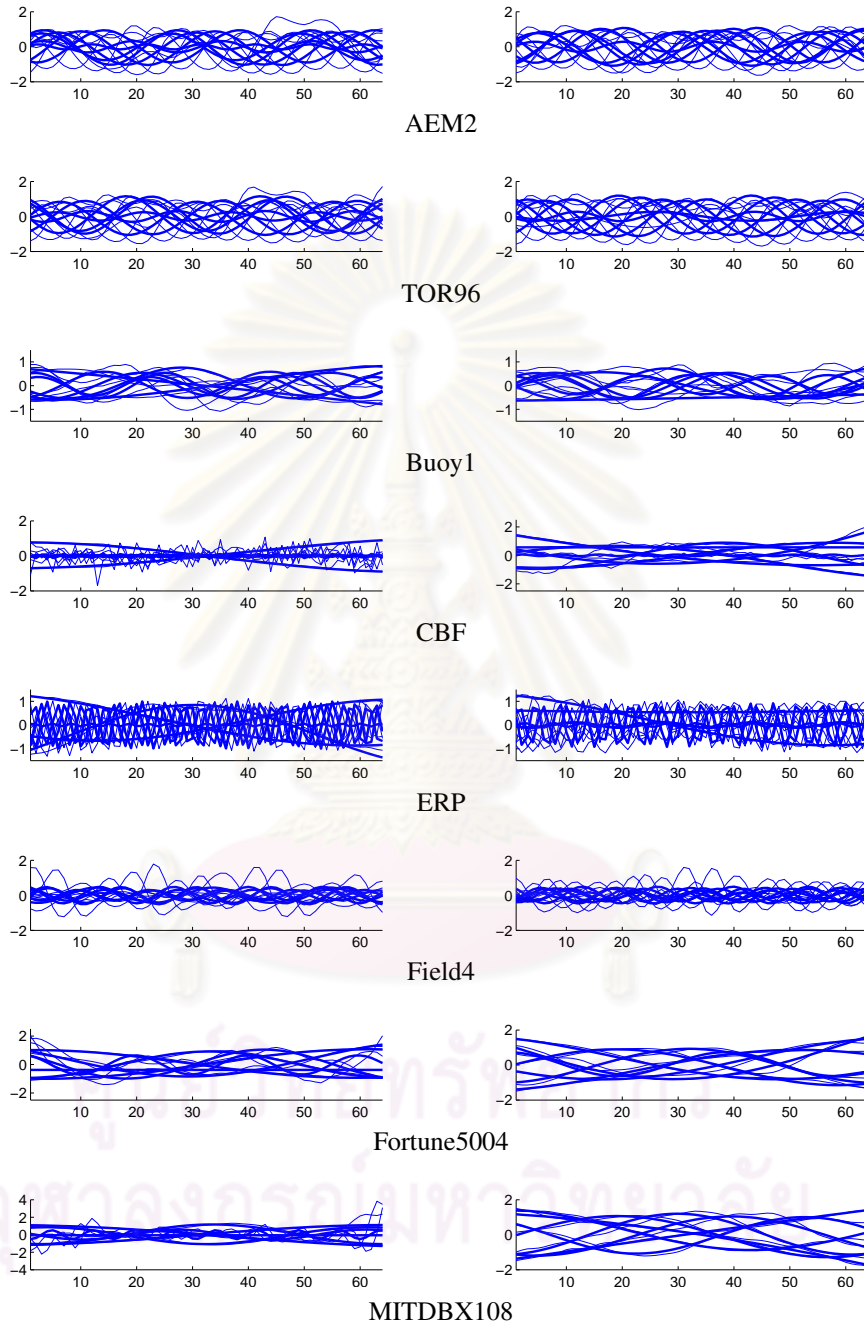


Figure B.20: Constructed sine waves generated from STSC using  $k$ -hierarchical clustering with complete linkage (left) and average linkage (right) inter-distance functions when  $k = 7$  and  $w = 64$ .

**APPENDIX C**

**COMPLETE EXPERIMENTAL RESULTS OF THE  
EXPERIMENT IN CHAPTER 3**



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

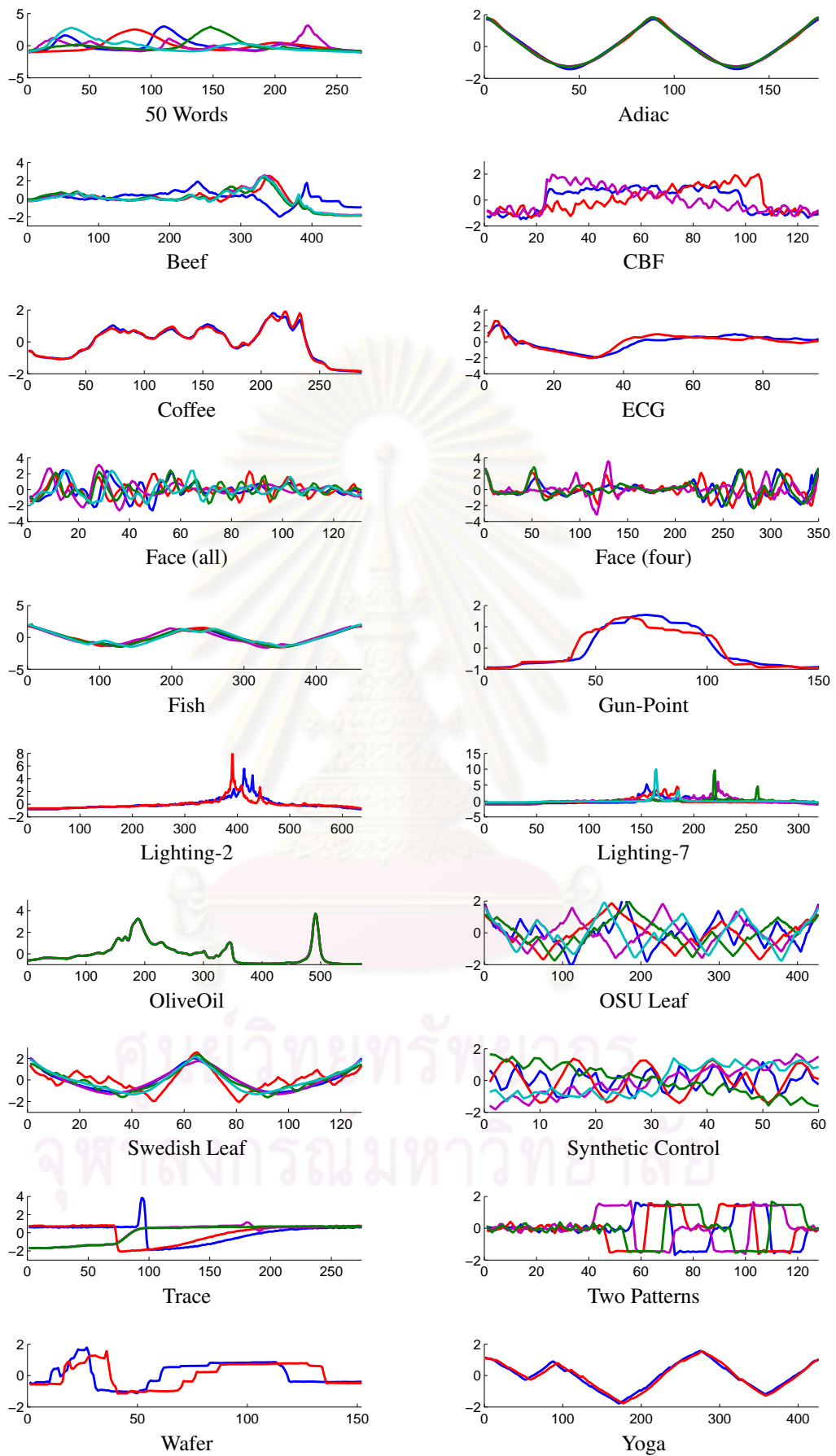


Figure C.1: Averaged results generated from CDTW function of each dataset

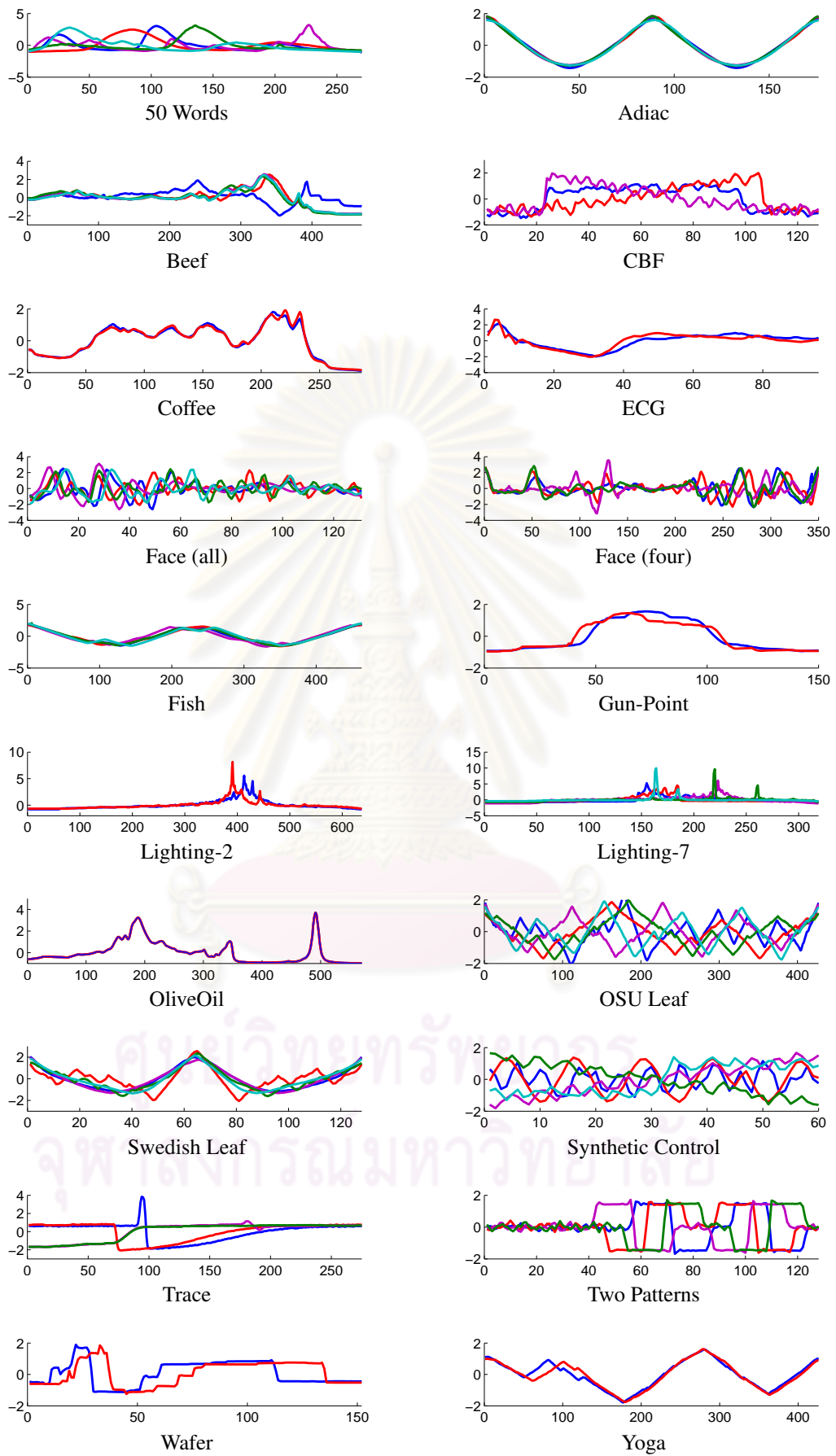


Figure C.2: Averaged results generated from ICDTW function of each dataset

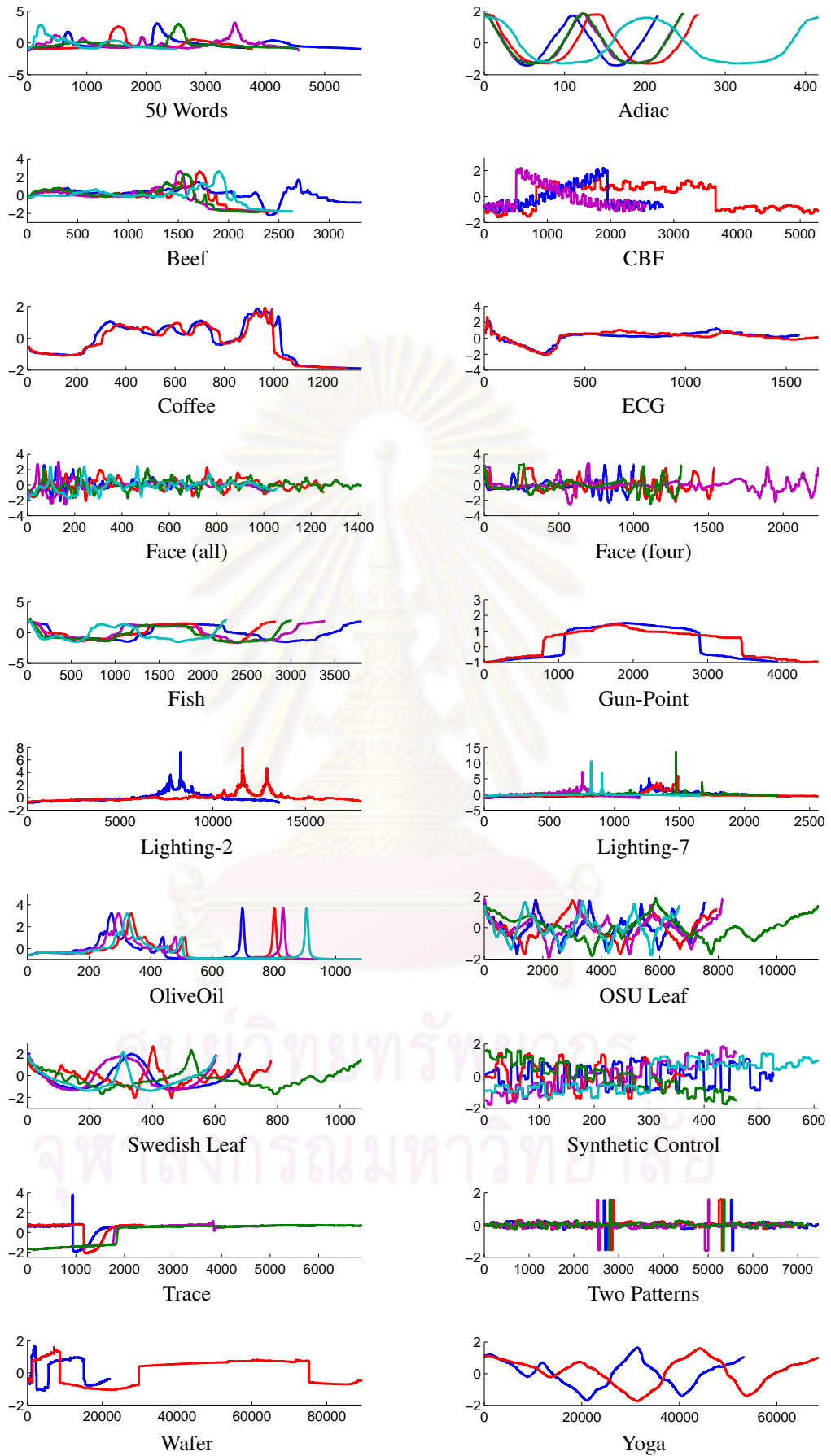


Figure C.3: Averaged results generated from NLAFF of each dataset.

**APPENDIX D**

**COMPLETE EXPERIMENTAL RESULTS OF THE  
EXPERIMENT IN CHAPTER 4**



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



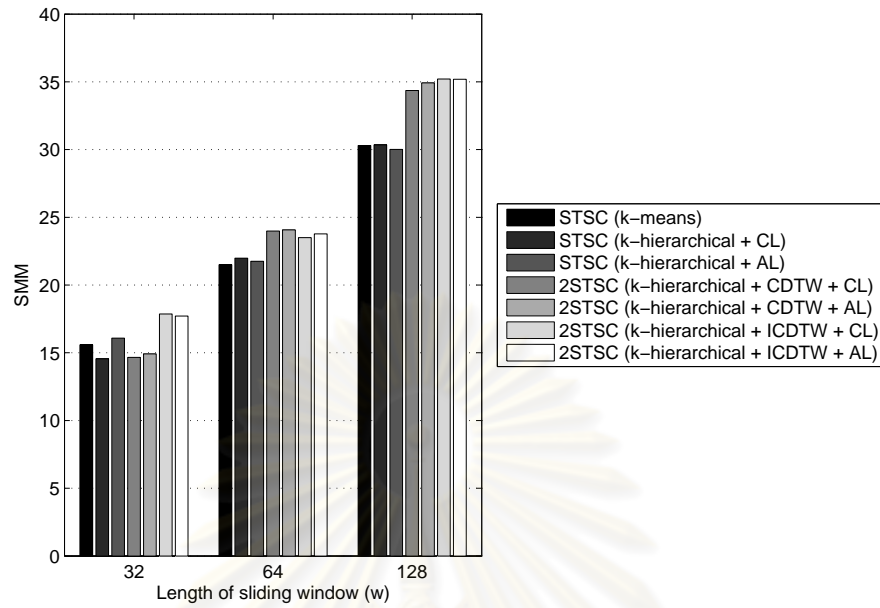


Figure D.1: SMMs of AEM2 when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied.

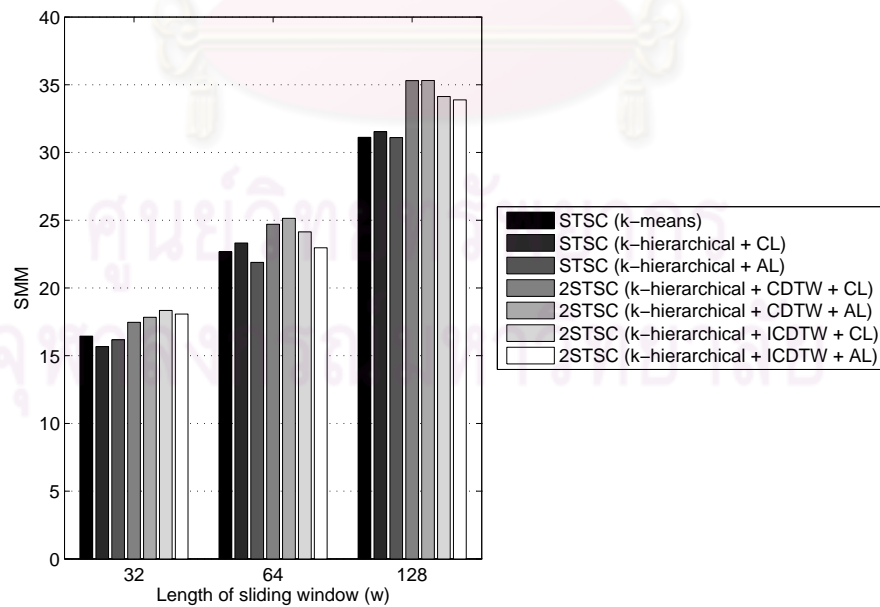


Figure D.2: SMMs of TOR96 when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied.

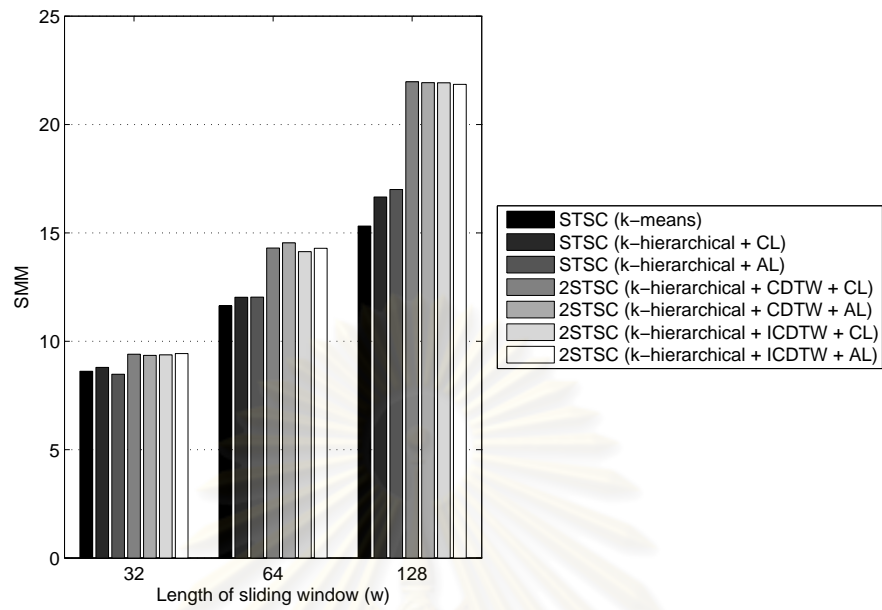


Figure D.3: SMMs of Buoy1 when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied.

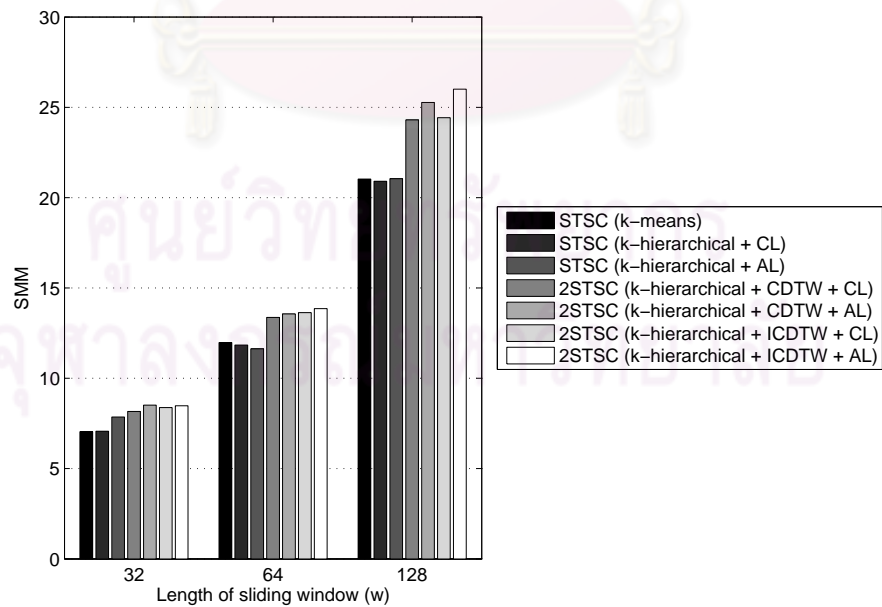


Figure D.4: SMMs of CBF when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied.

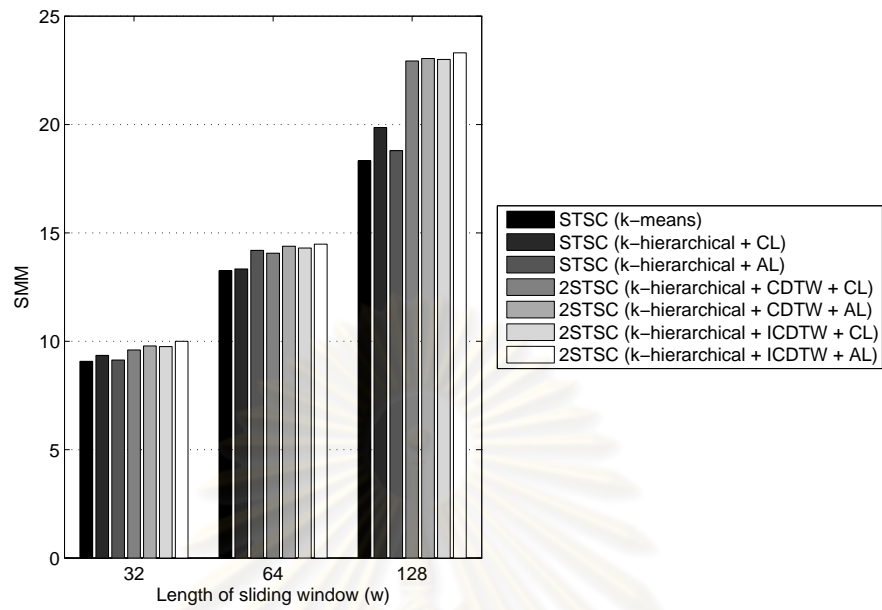


Figure D.5: SMMs of ERP when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied.

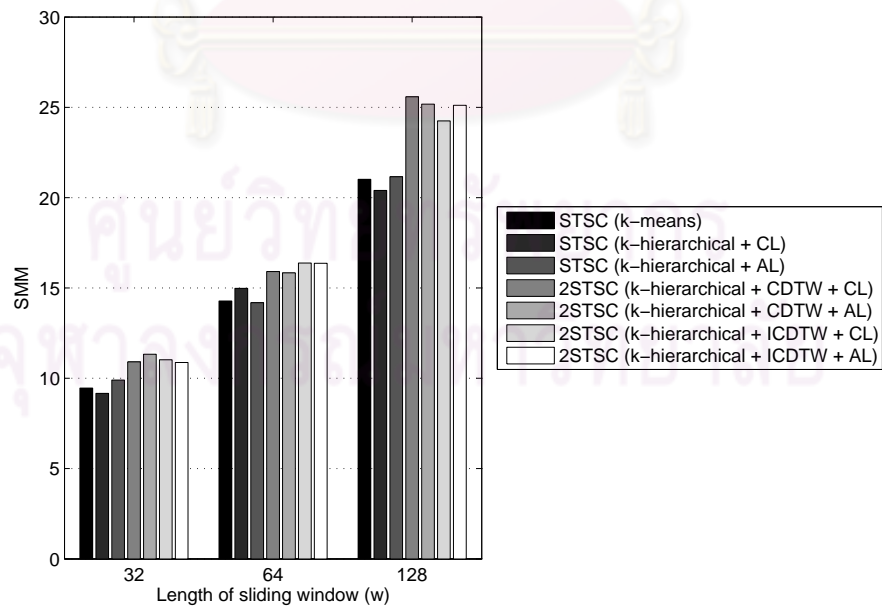


Figure D.6: SMMs of Field4 when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied.

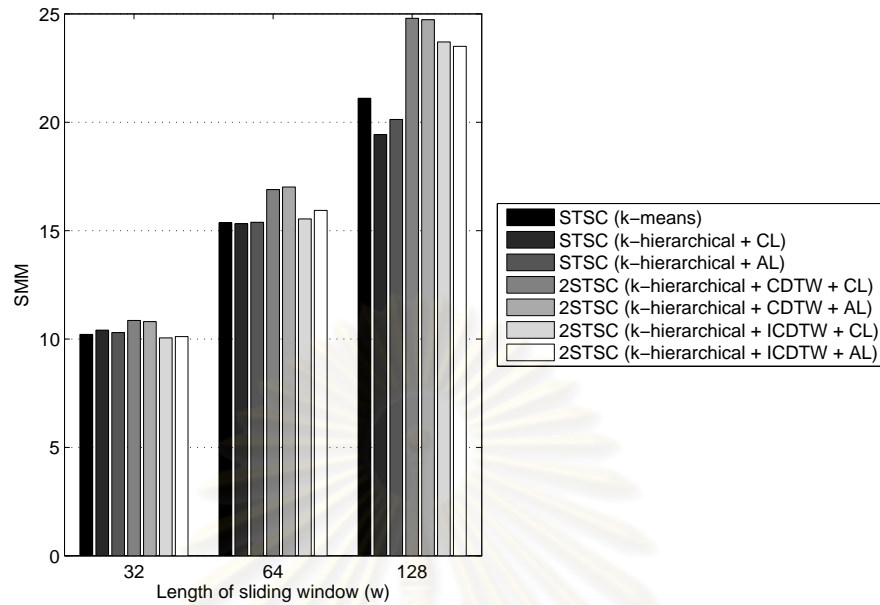


Figure D.7: SMMs of Fortune5004 when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied.

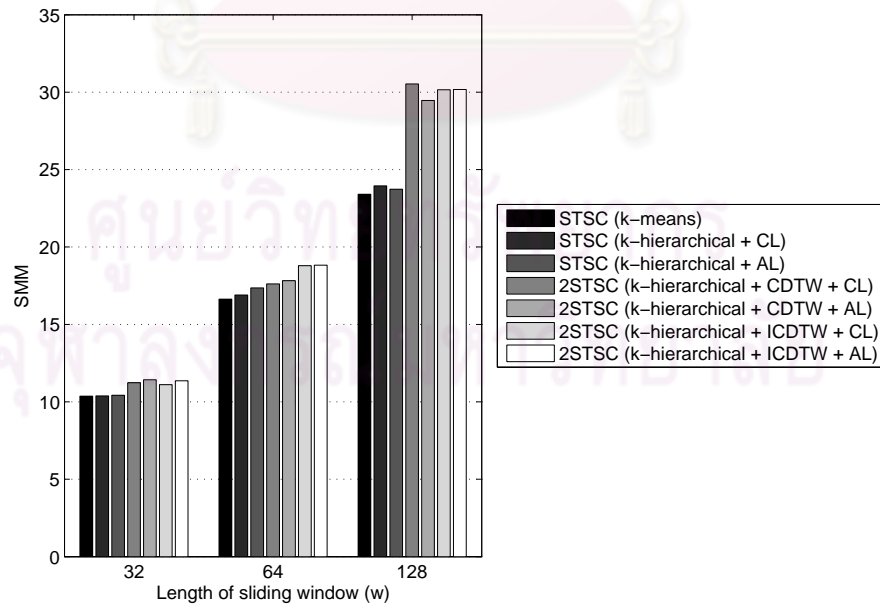


Figure D.8: SMMs of MITDBX108 when the number of clusters ( $k$ ) is 3 and the length of sliding window ( $w$ ) is varied.

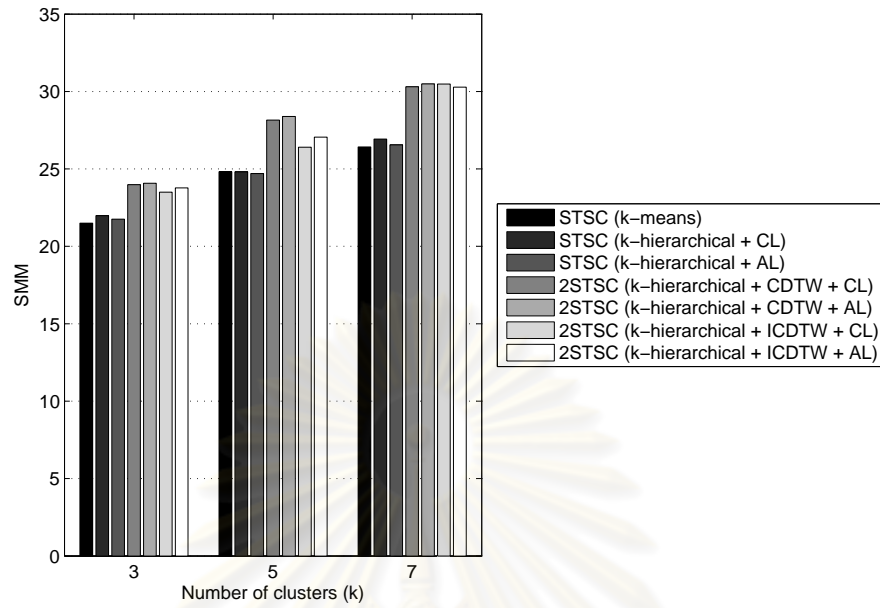


Figure D.9: SMMs of AEM2 when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied.

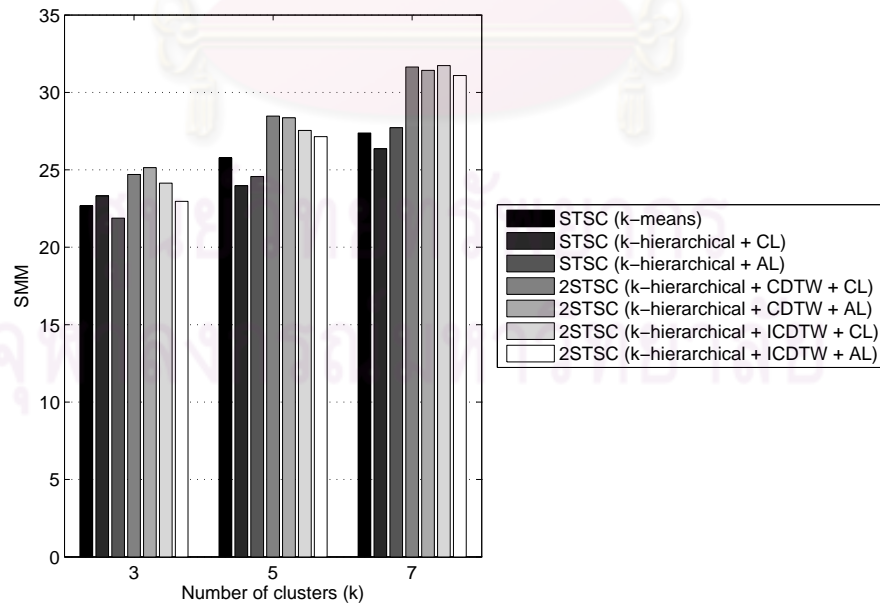


Figure D.10: SMMs of TOR96 when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied.

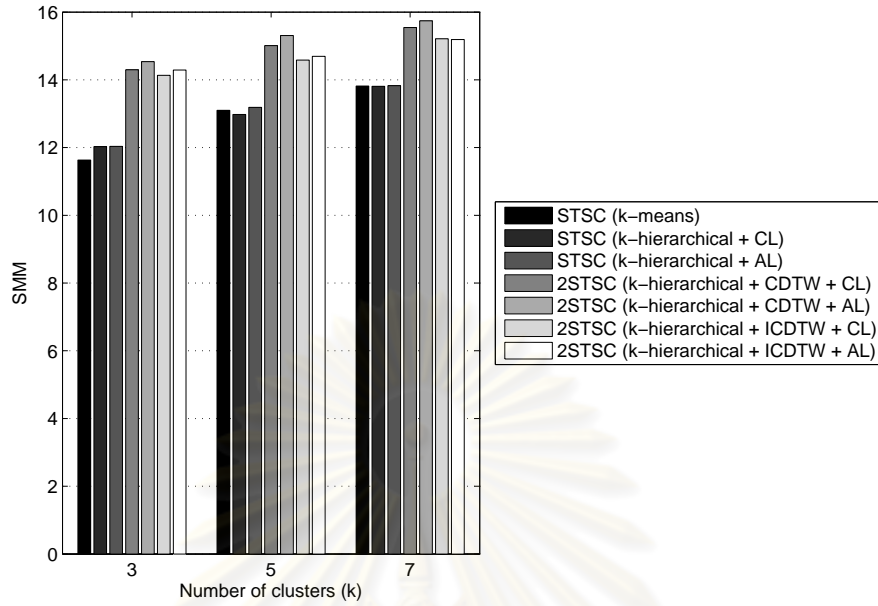


Figure D.11: SMMs of Buoy1 when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied.

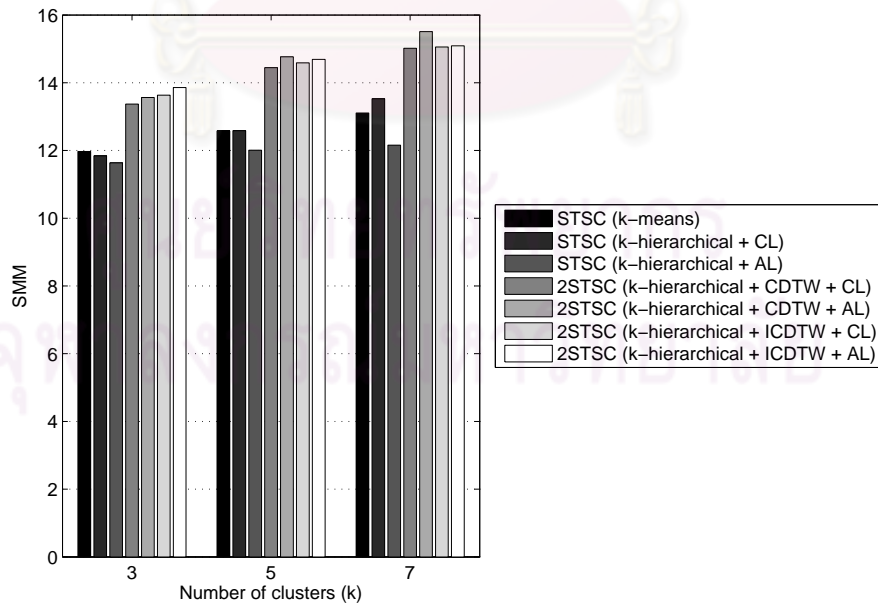


Figure D.12: SMMs of CBF when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied.

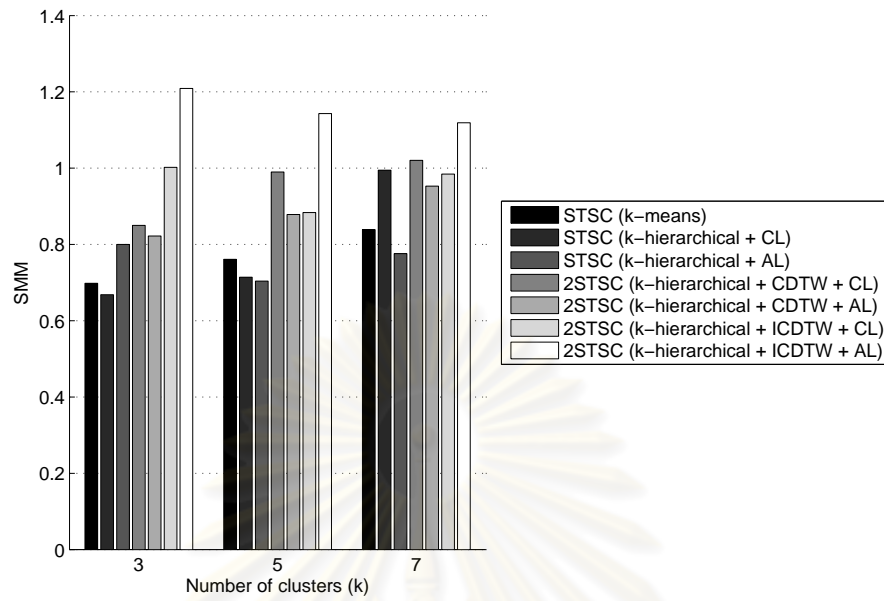


Figure D.13: SMMs of ERP when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied.

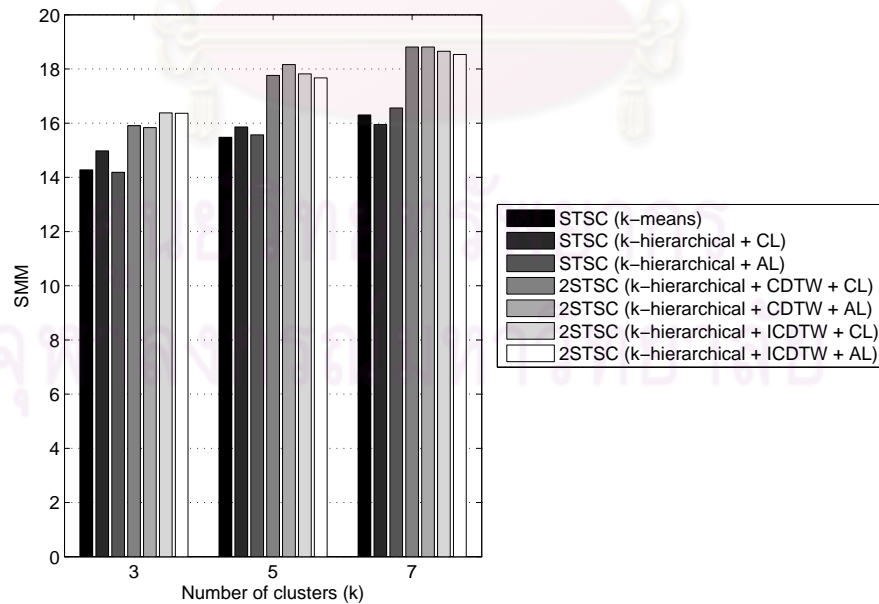


Figure D.14: SMMs of Field4 when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied.

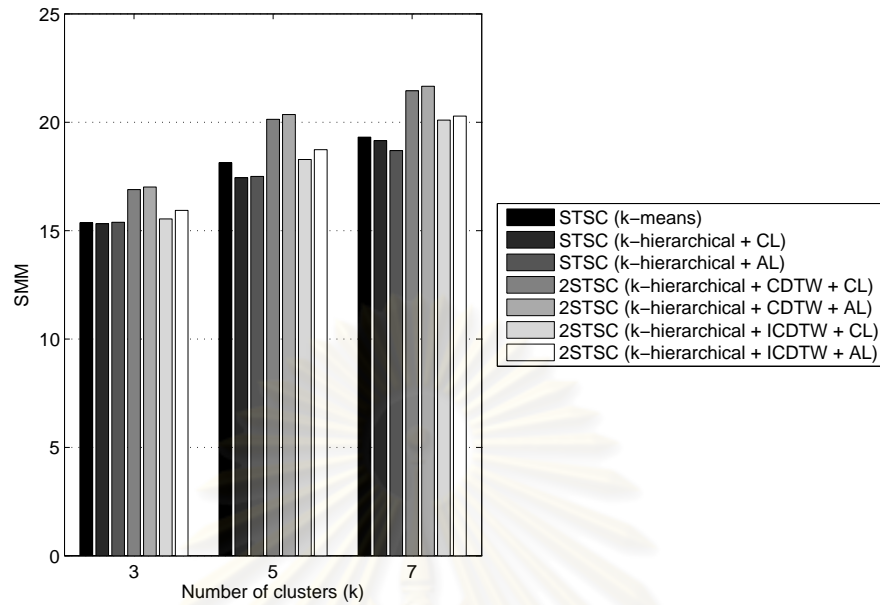


Figure D.15: SMMs of Fortune5004 when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied.

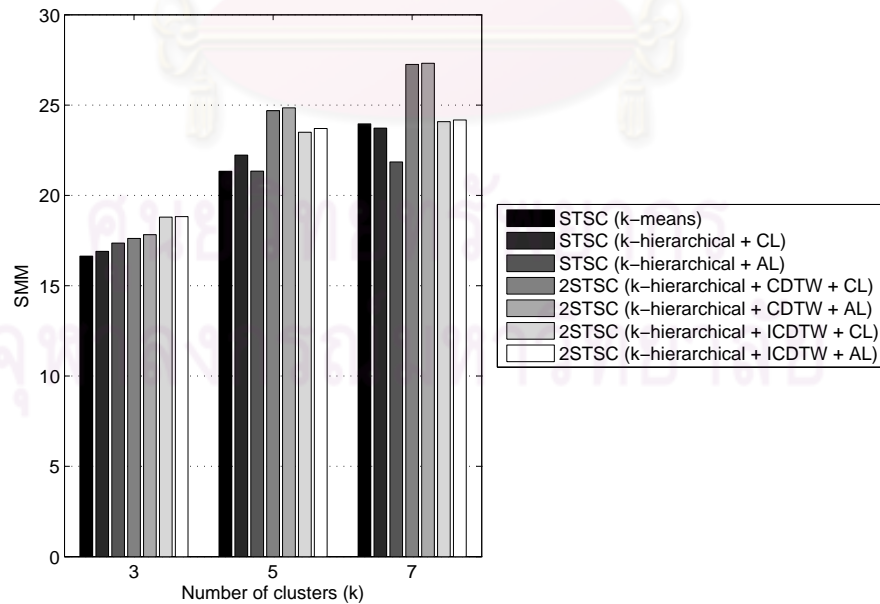


Figure D.16: SMMs of MITDBX108 when the length of sliding window ( $w$ ) is 64 and the number of clusters ( $k$ ) is varied.



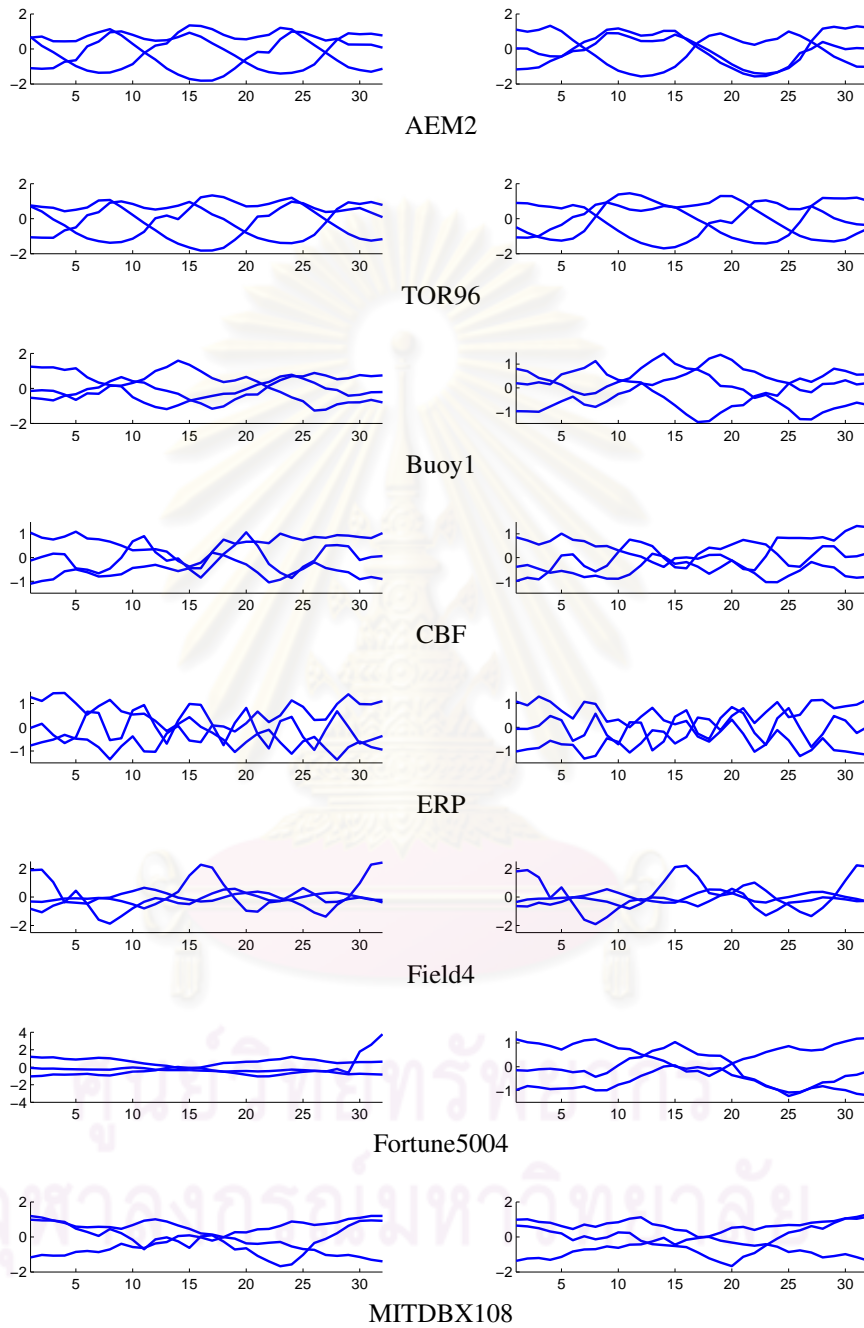


Figure D.17: Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using CDTW function when  $k = 3$  and  $w = 32$ .

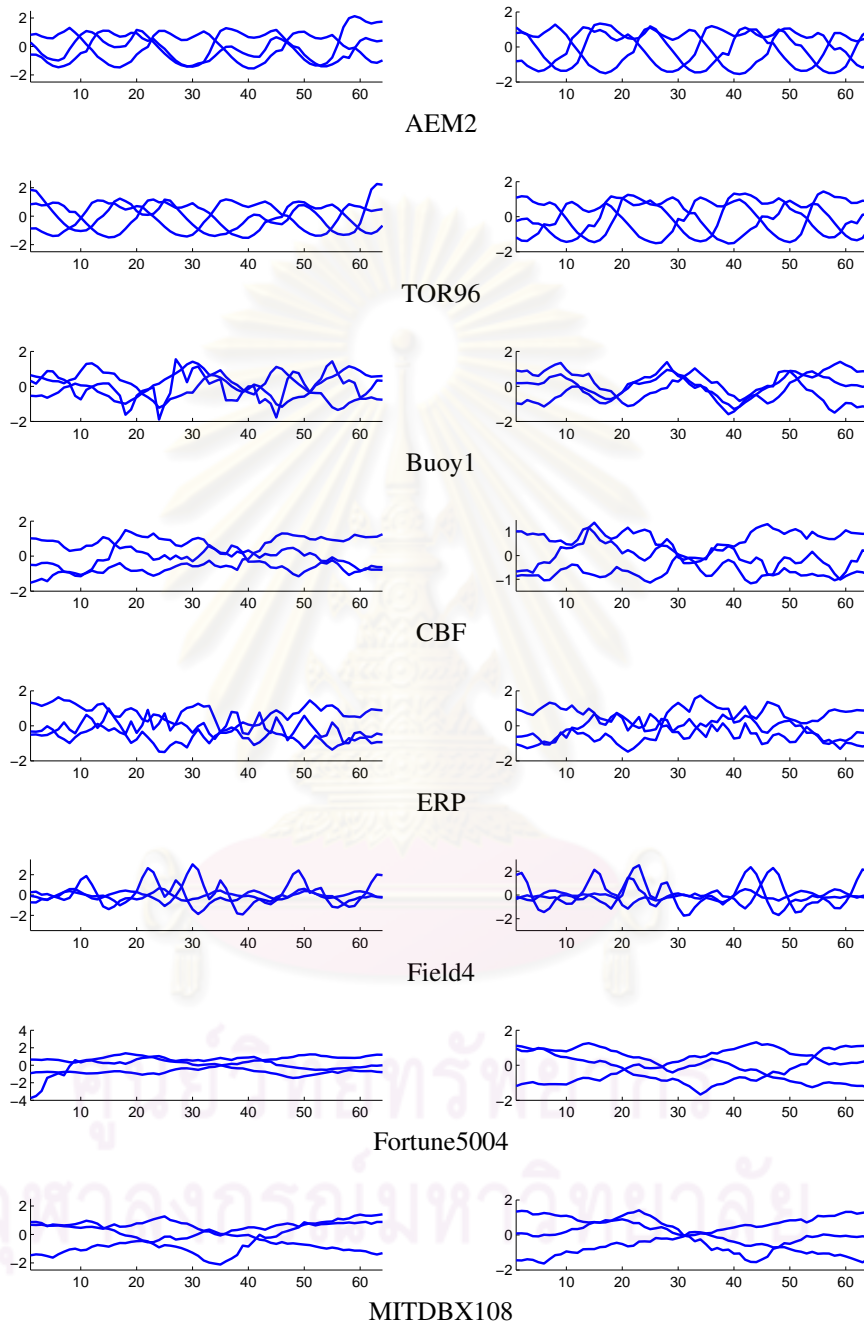


Figure D.18: Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using CDTW function when  $k = 3$  and  $w = 64$ .

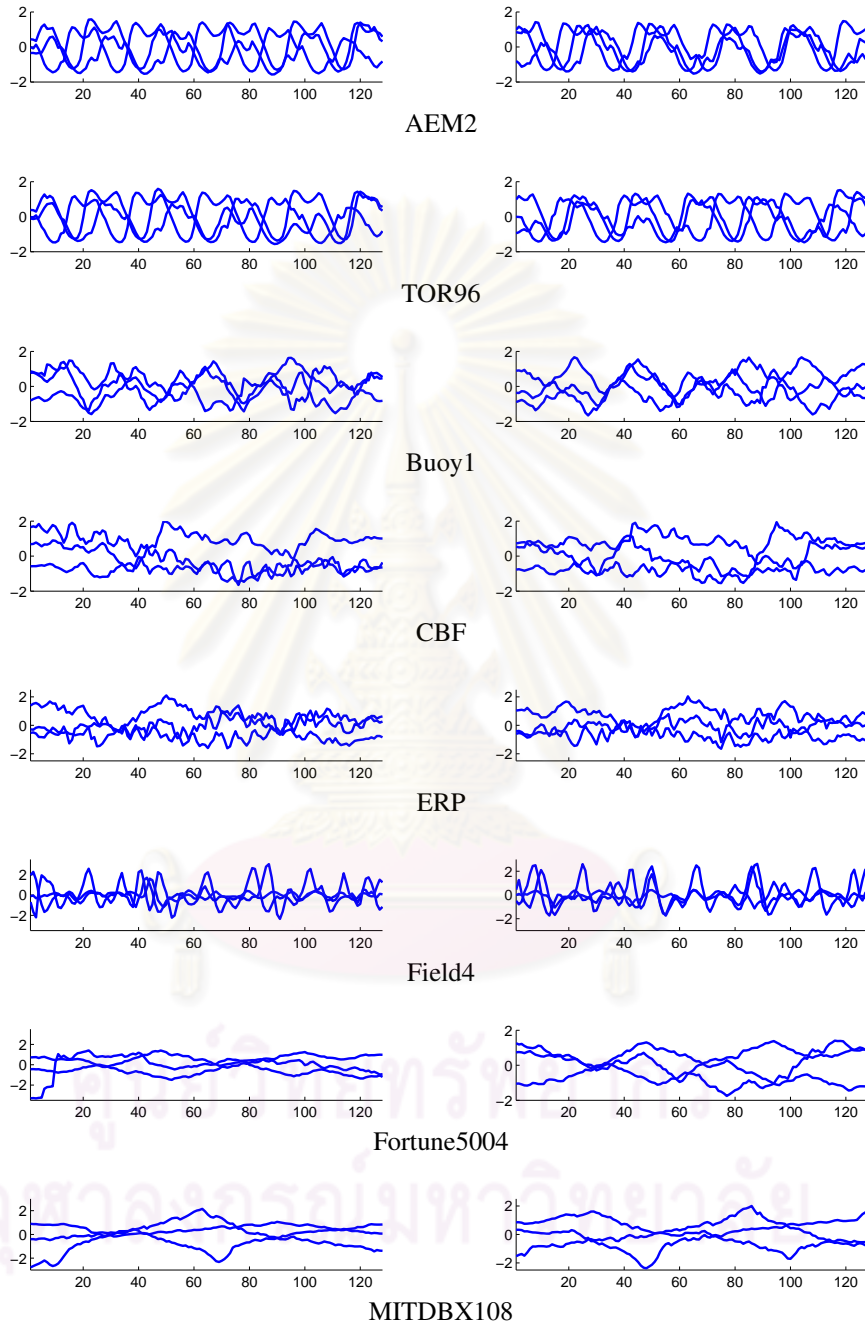


Figure D.19: Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using CDTW function when  $k = 3$  and  $w = 128$ .

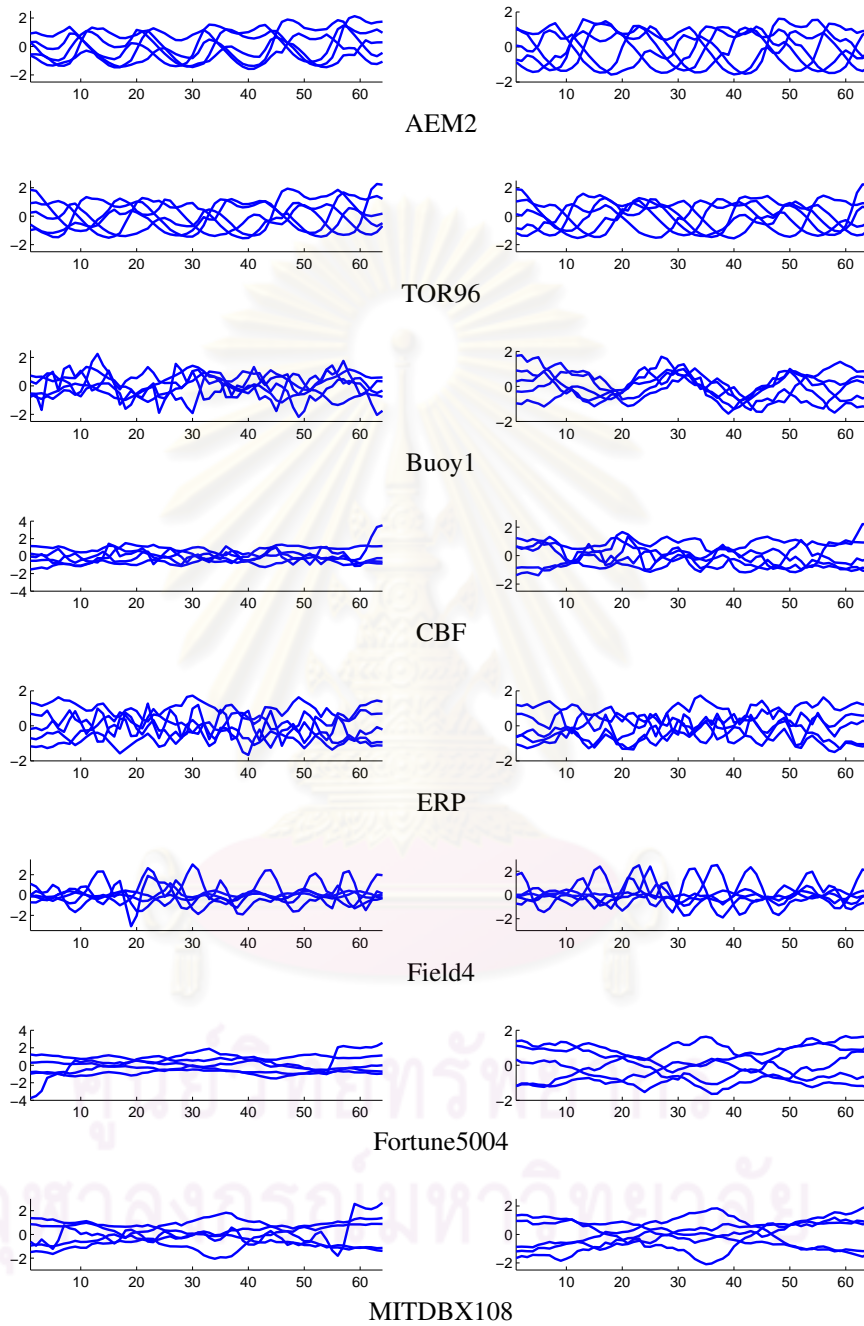


Figure D.20: Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using ICDTW function when  $k = 5$  and  $w = 64$ .

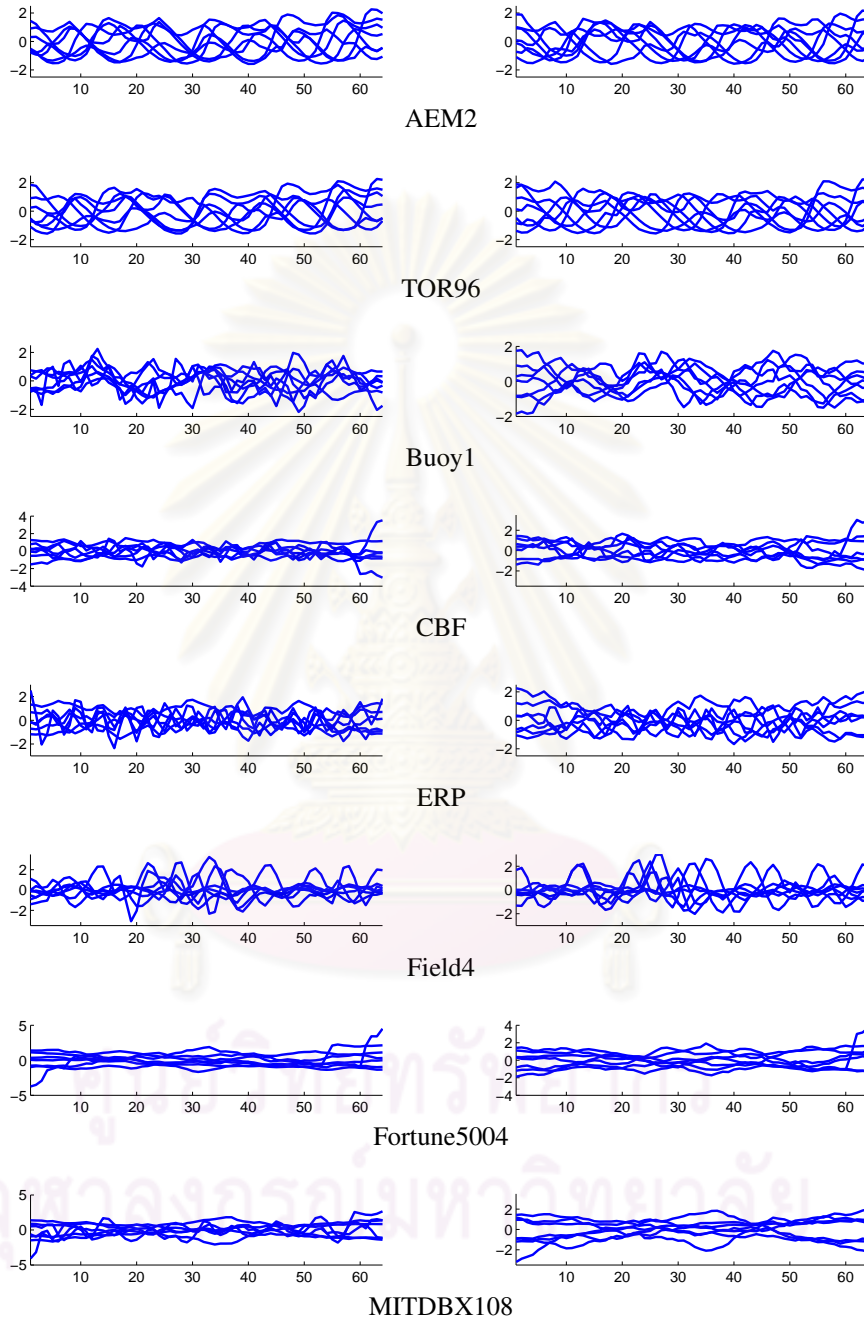


Figure D.21: Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using ICDTW function when  $k = 7$  and  $w = 64$ .

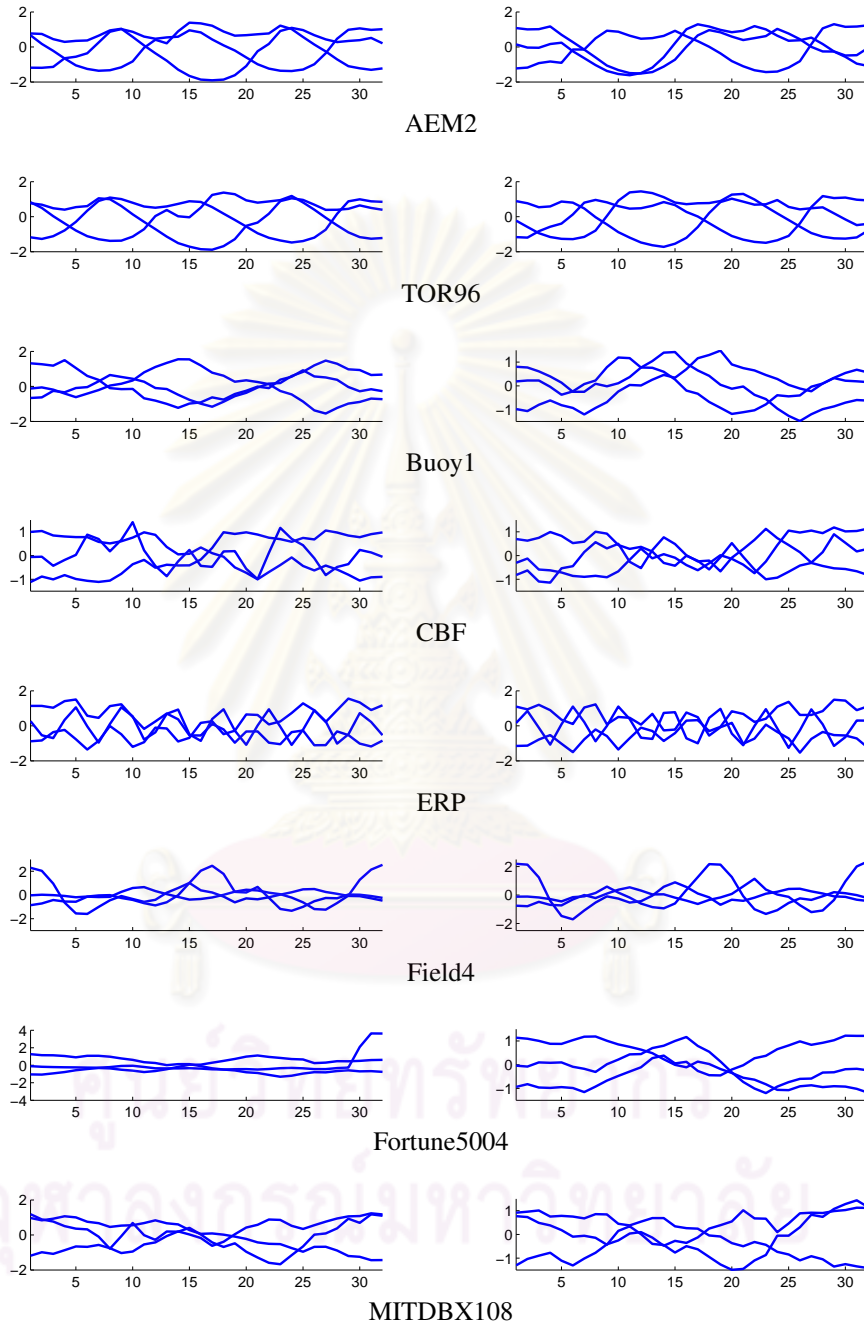


Figure D.22: Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using ICDTW function when  $k = 3$  and  $w = 32$ .

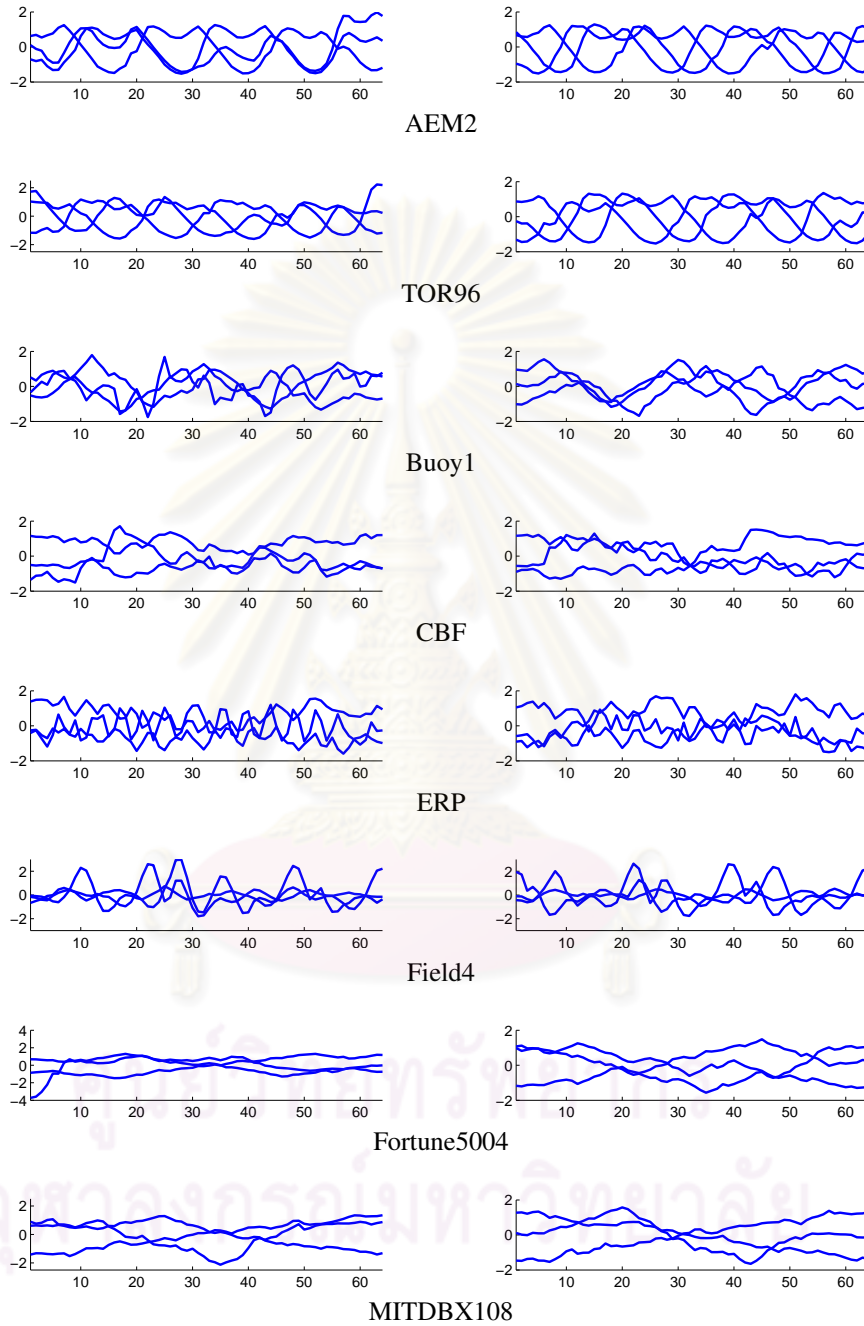


Figure D.23: Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using ICDTW function when  $k = 3$  and  $w = 64$ .

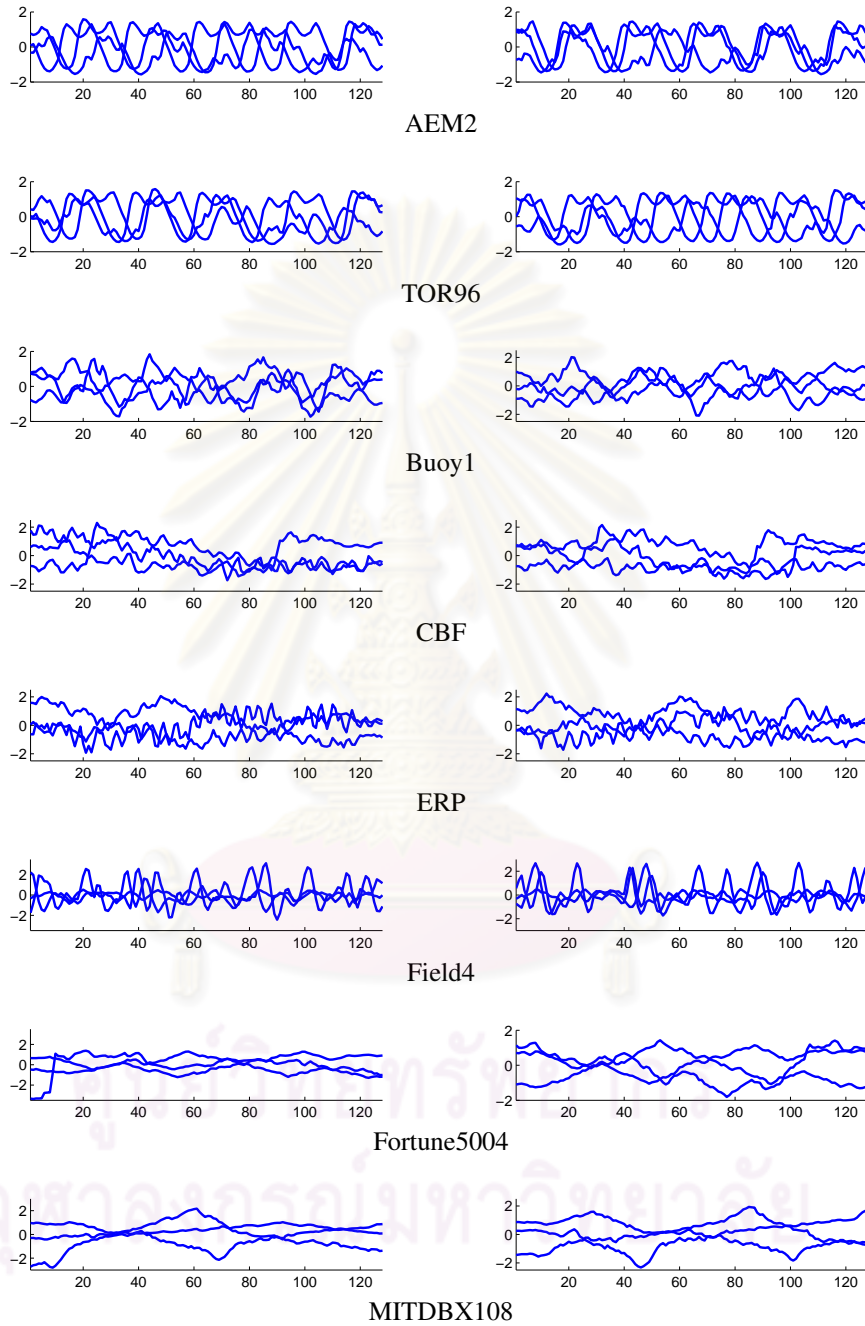


Figure D.24: Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using ICDTW function when  $k = 3$  and  $w = 128$ .



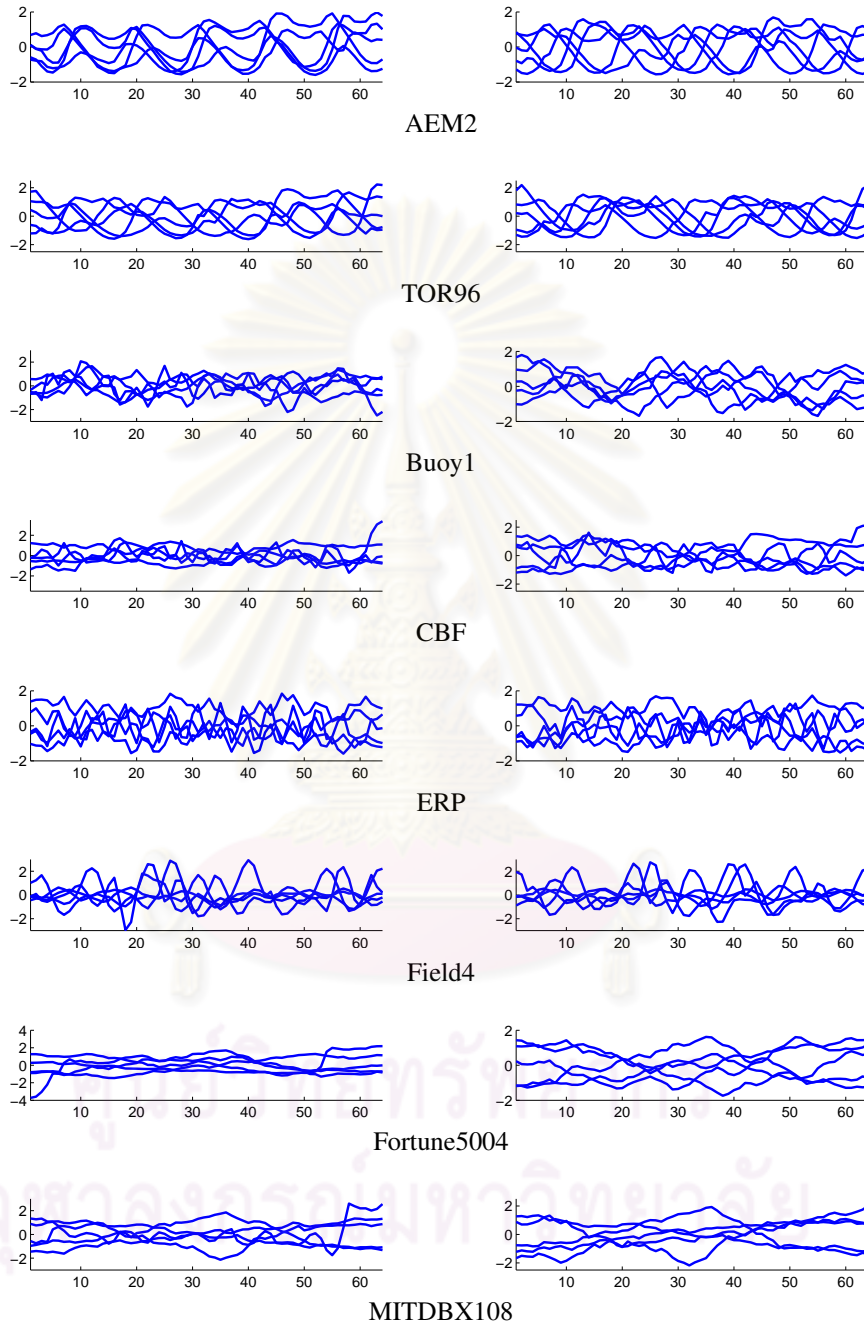


Figure D.25: Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using ICDTW function when  $k = 5$  and  $w = 64$ .

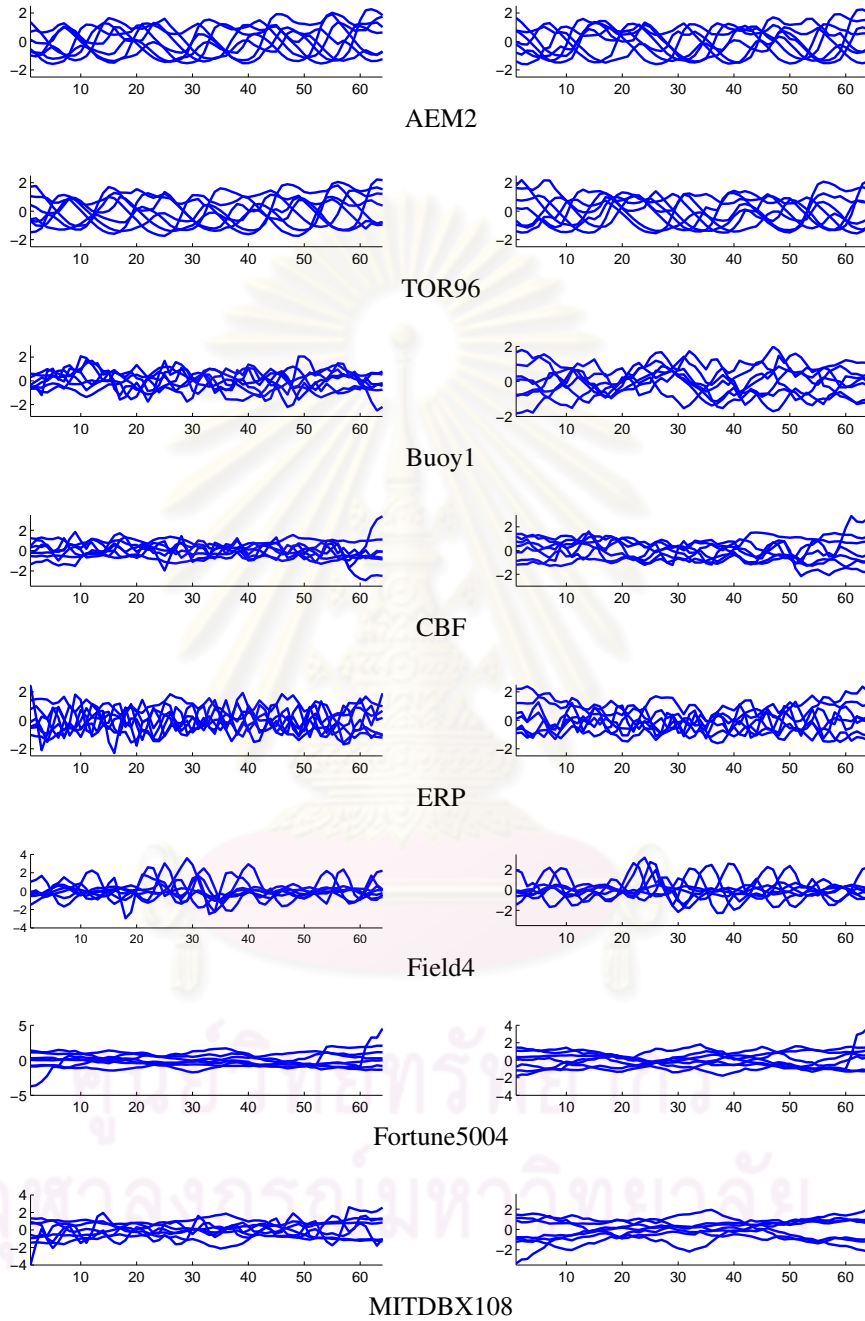


Figure D.26: Cluster representatives generated from 2STSC with complete linkage (left) and average linkage (right) using ICDTW function when  $k = 7$  and  $w = 64$ .

**APPENDIX E**

**COMPLETE EXPERIMENTAL RESULTS OF THE FIRST  
EXPERIMENT IN CHAPTER 5**



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

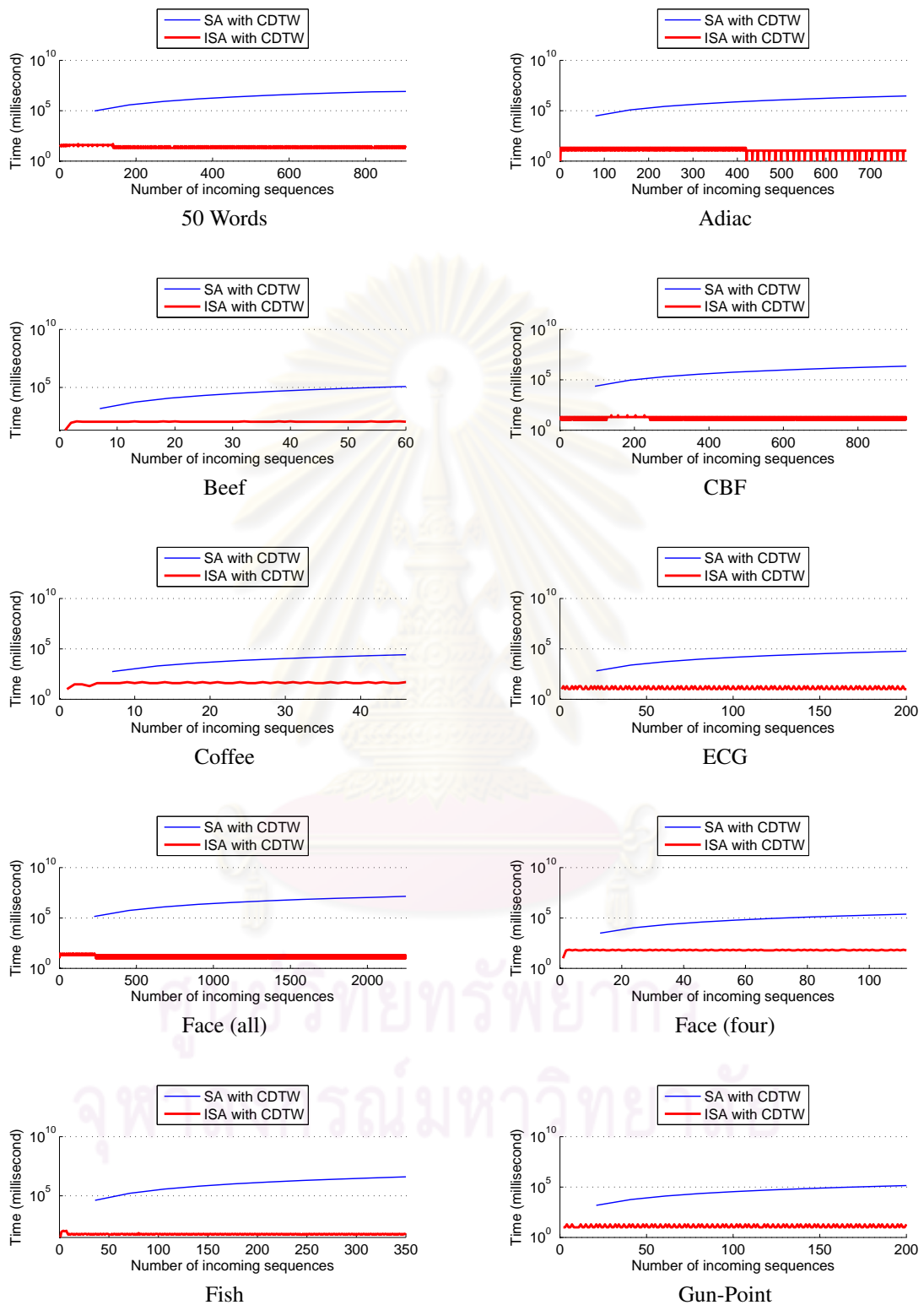


Figure E.1: Computational time of Incremental Shape-based Averaging and Shape-based Averaging with CDTW function when a new incoming sequence arrives.

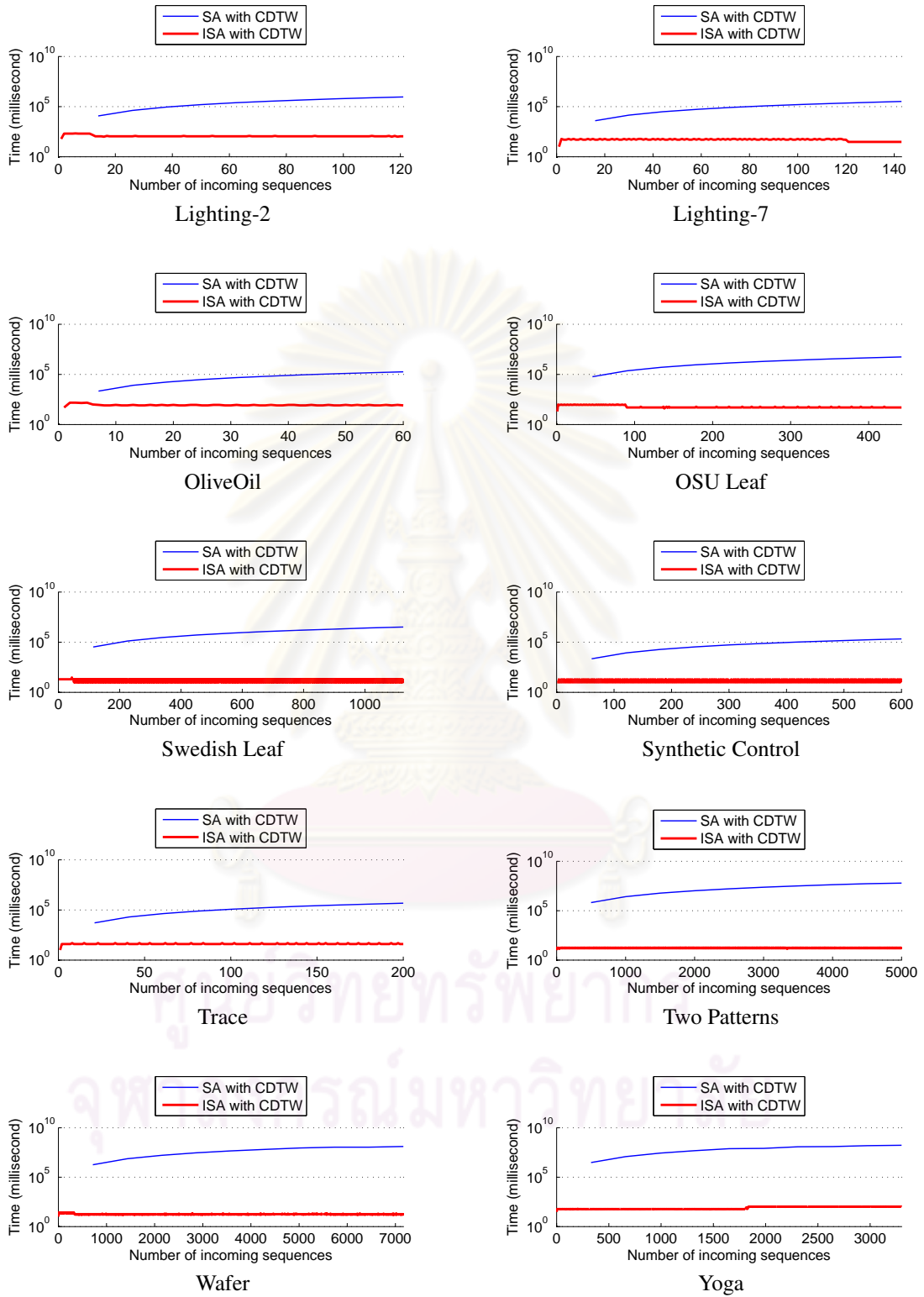


Figure E.2: Computational time of Incremental Shape-based Averaging and Shape-based Averaging with CDTW function when a new incoming sequence arrives. (cont.)

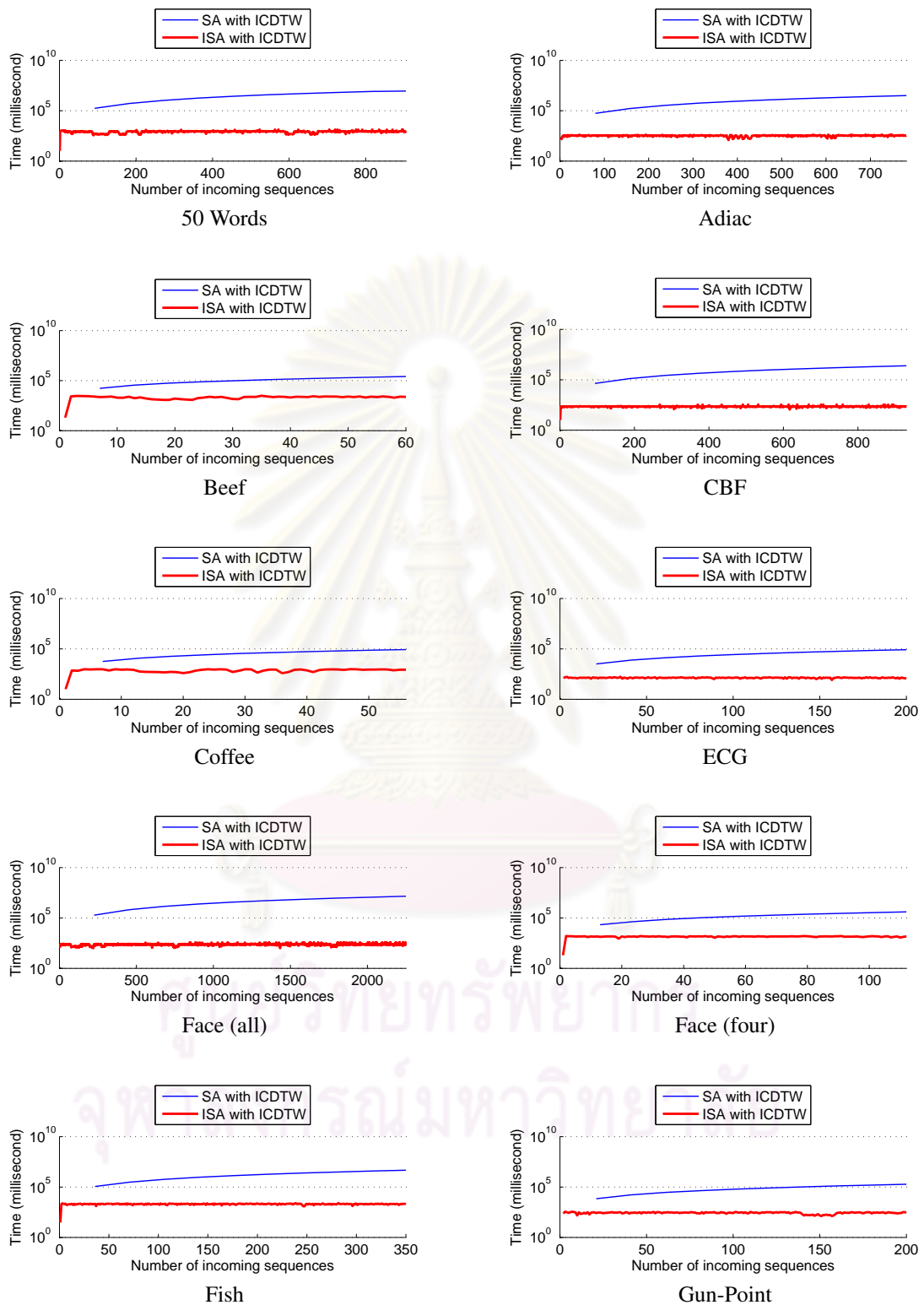


Figure E.3: Computational time of Incremental Shape-based Averaging and Shape-based Averaging with ICDTW function when a new incoming sequence arrives.

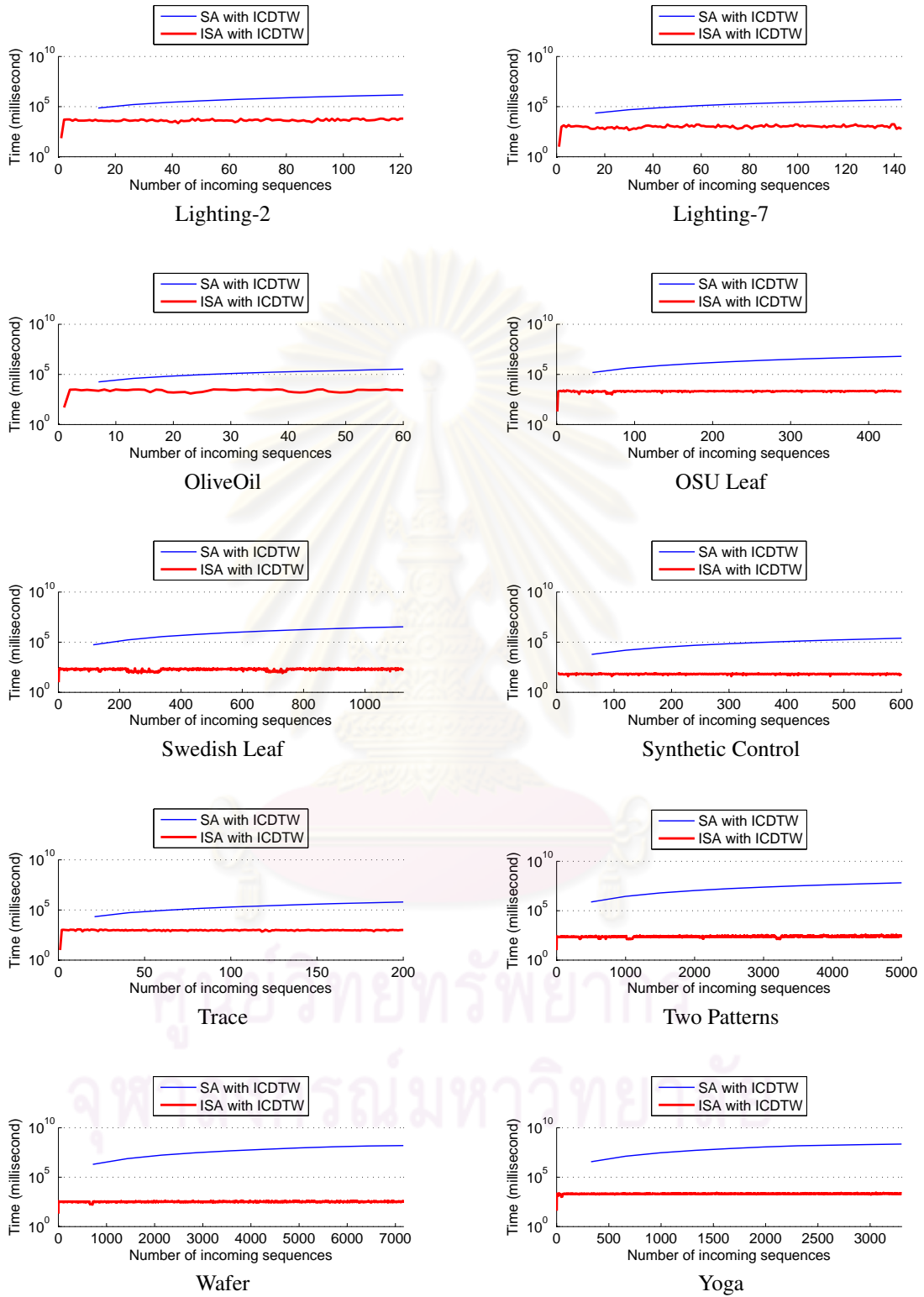


Figure E.4: Computational time of than Shape-based Averaging around Incremental Shape-based Averaging and Shape-based Averaging with ICDTW function when a new incoming sequence arrives.

**APPENDIX F**

**COMPLETE EXPERIMENTAL RESULTS OF THE SECOND  
EXPERIMENT ON CHAPTER 5**



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย



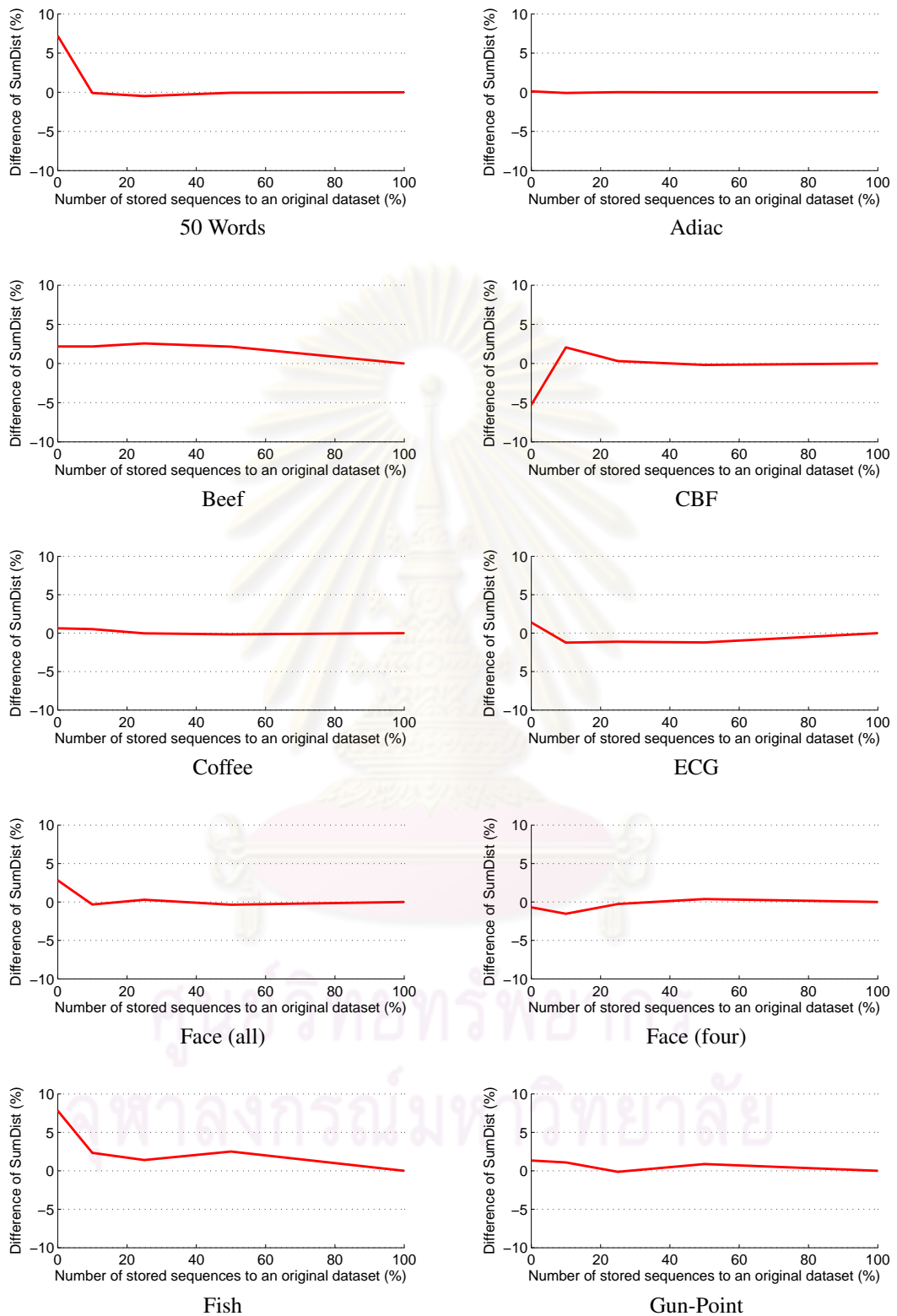


Figure F.1: Difference of SUMDIST of Incremental Shape-based Averaging with CDTW when the number of stored sequences to an original dataset is varied.

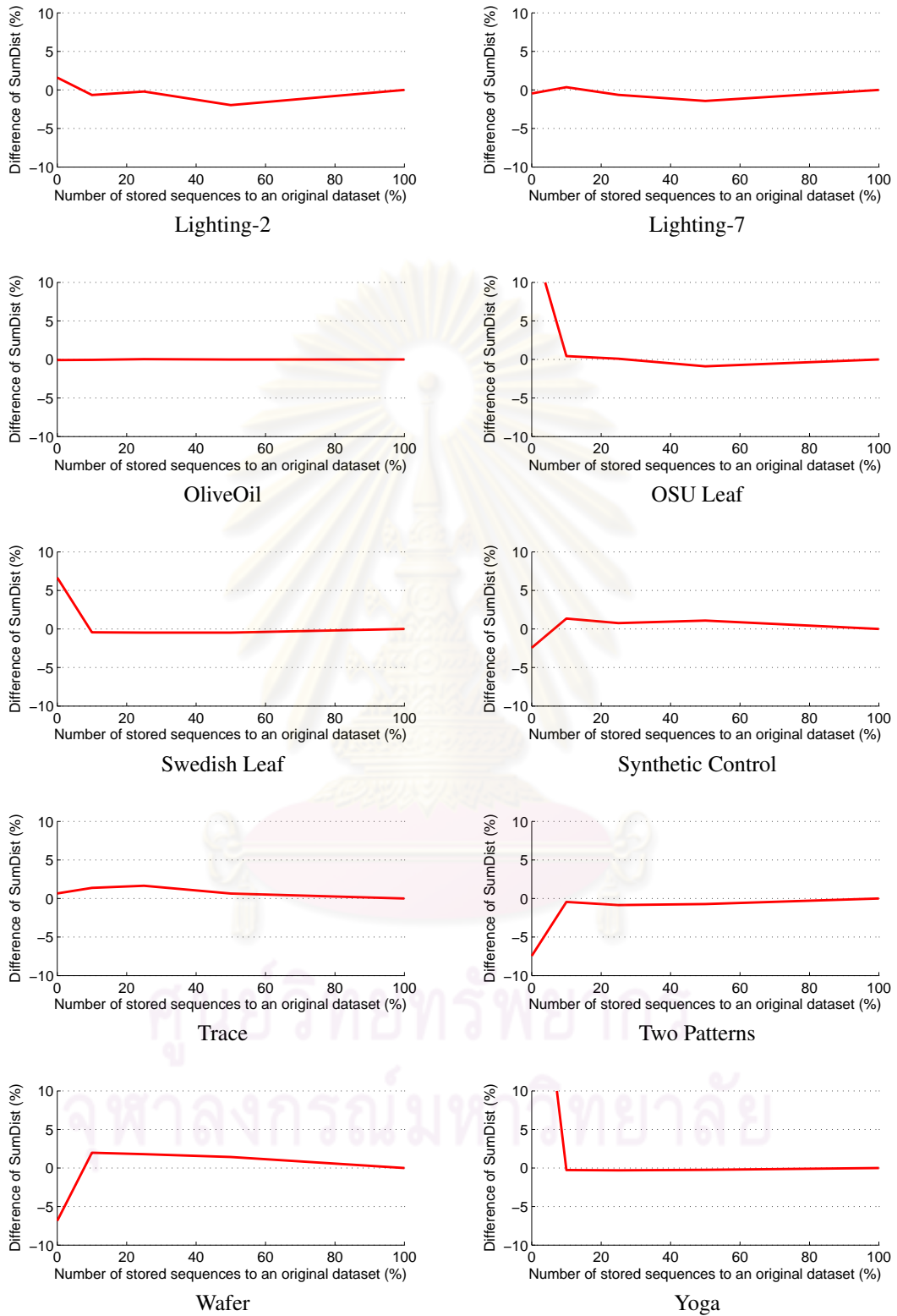


Figure F.2: Difference of SUMDIST of Incremental Shape-based Averaging with CDTW when the number of stored sequences to an original dataset is varied. (cont.)

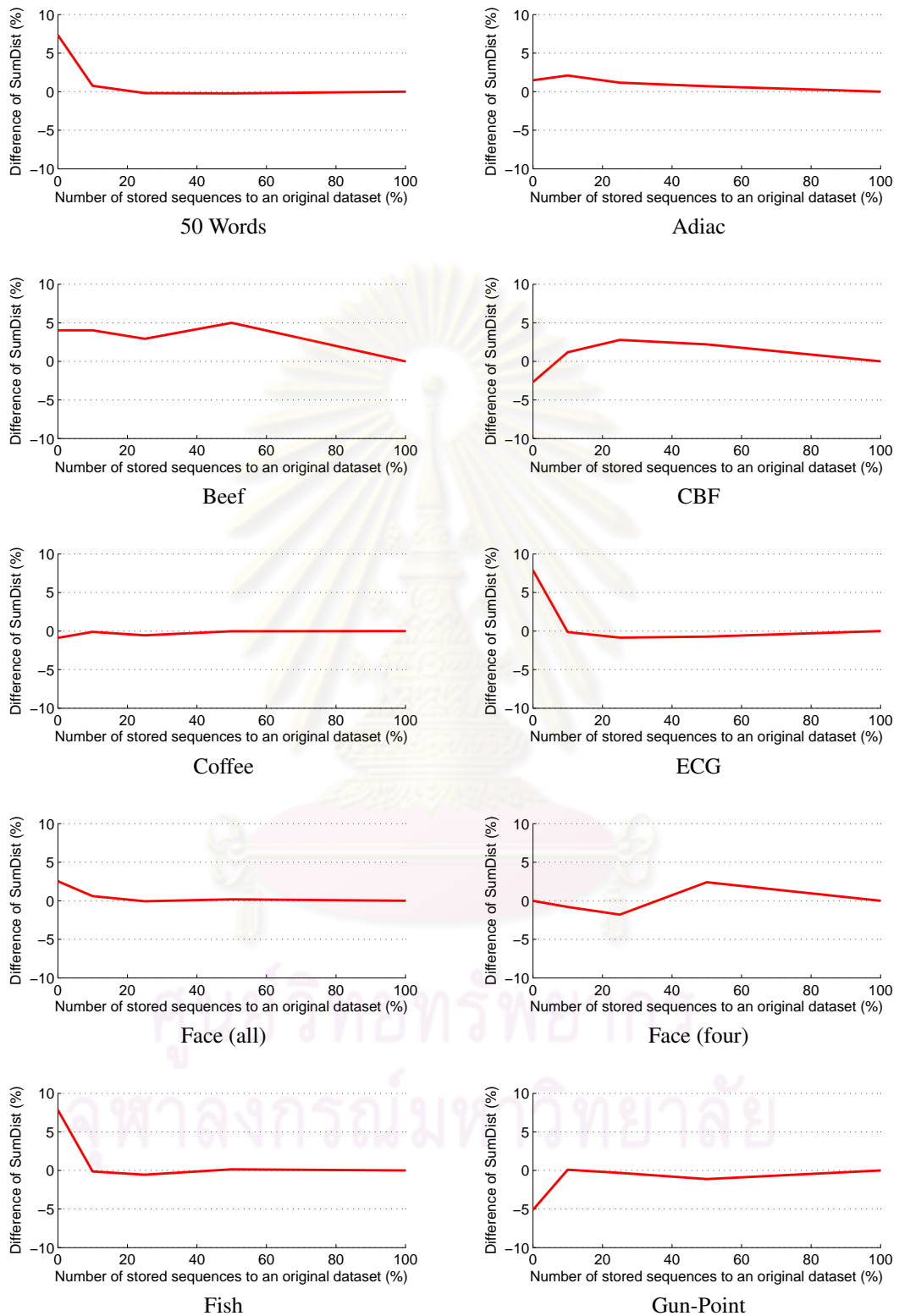


Figure F.3: Difference of SUMDIST of Incremental Shape-based Averaging with ICDTW when the number of stored sequences to an original dataset is varied.

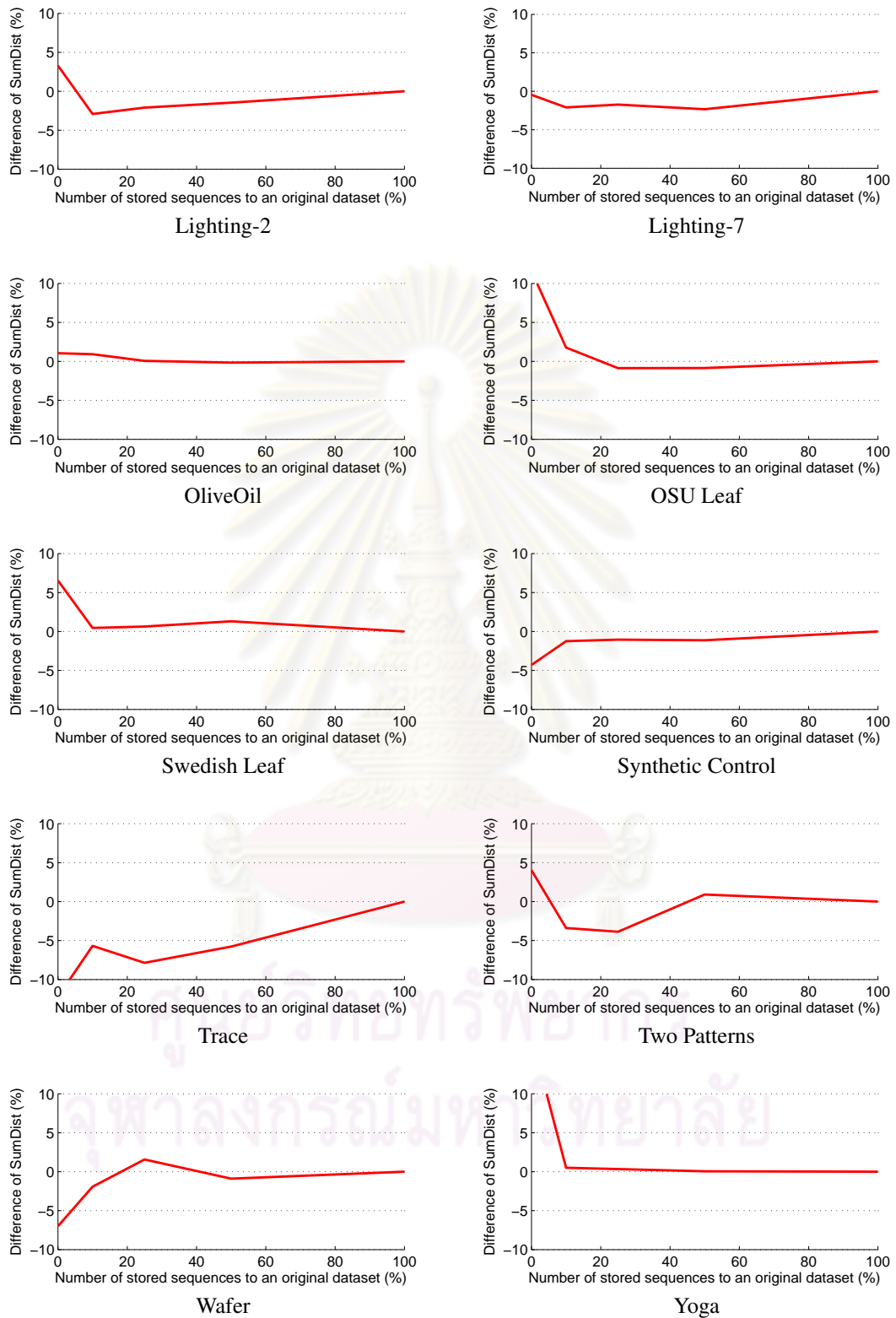


Figure F.4: Difference of SUMDIST of Incremental Shape-based Averaging with ICDTW when the number of stored sequences to an original dataset is varied. (cont.)

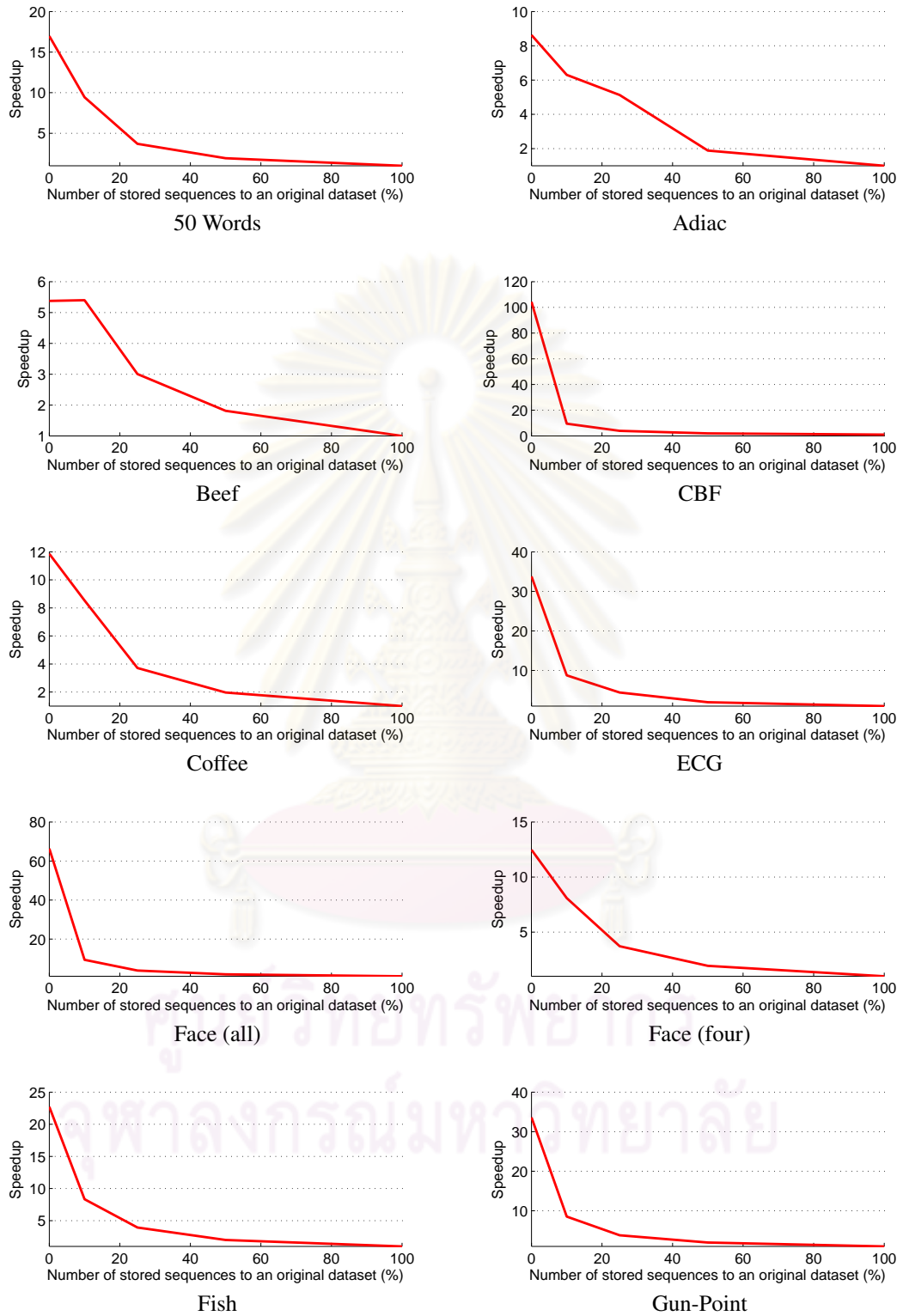


Figure F.5: Speedup of Incremental Shape-based Averaging with CDTW when the number of stored sequences to an original dataset is varied.

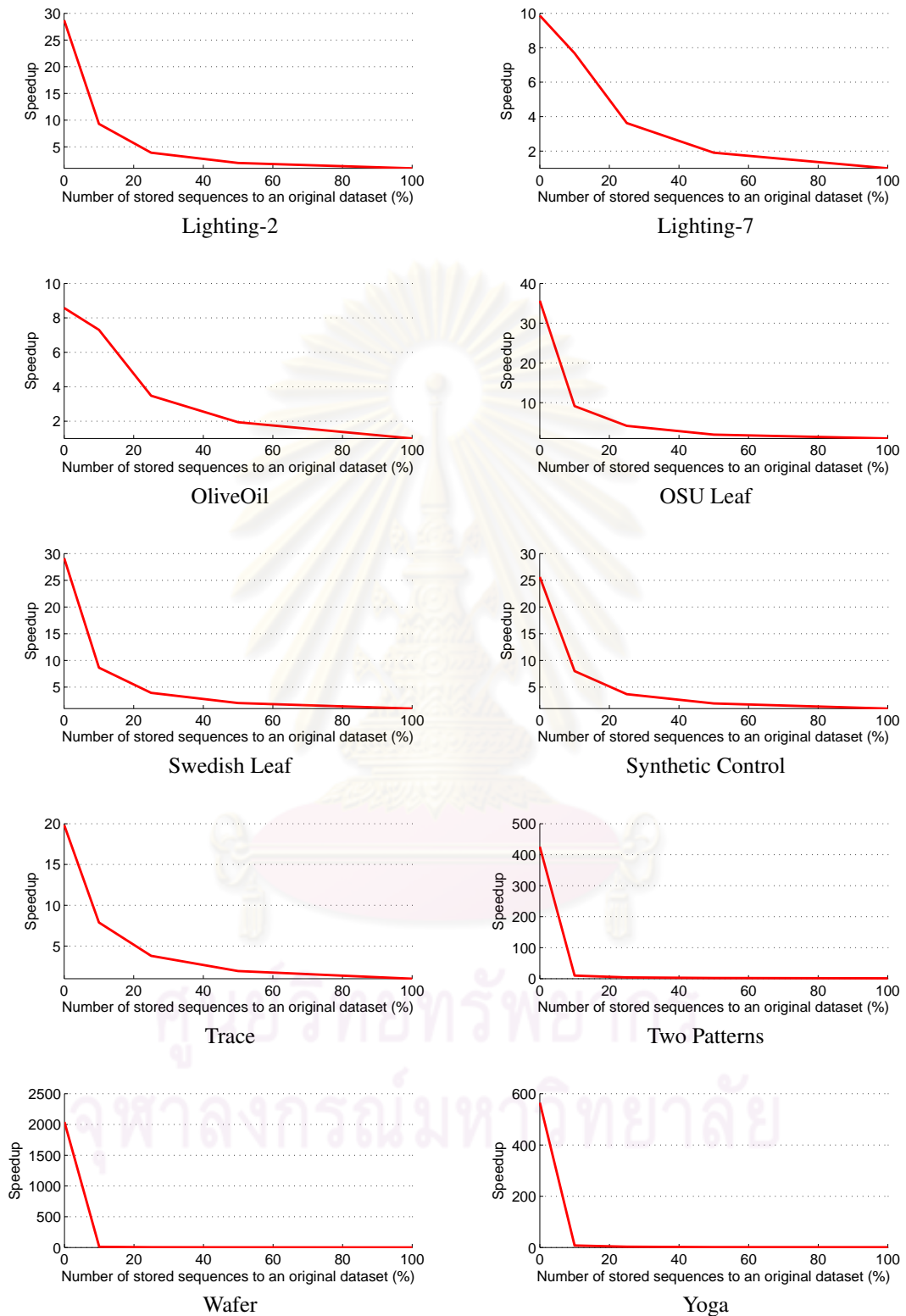


Figure F.6: Speedup of Incremental Shape-based Averaging with CDTW when the number of stored sequences to an original dataset is varied. (cont.)

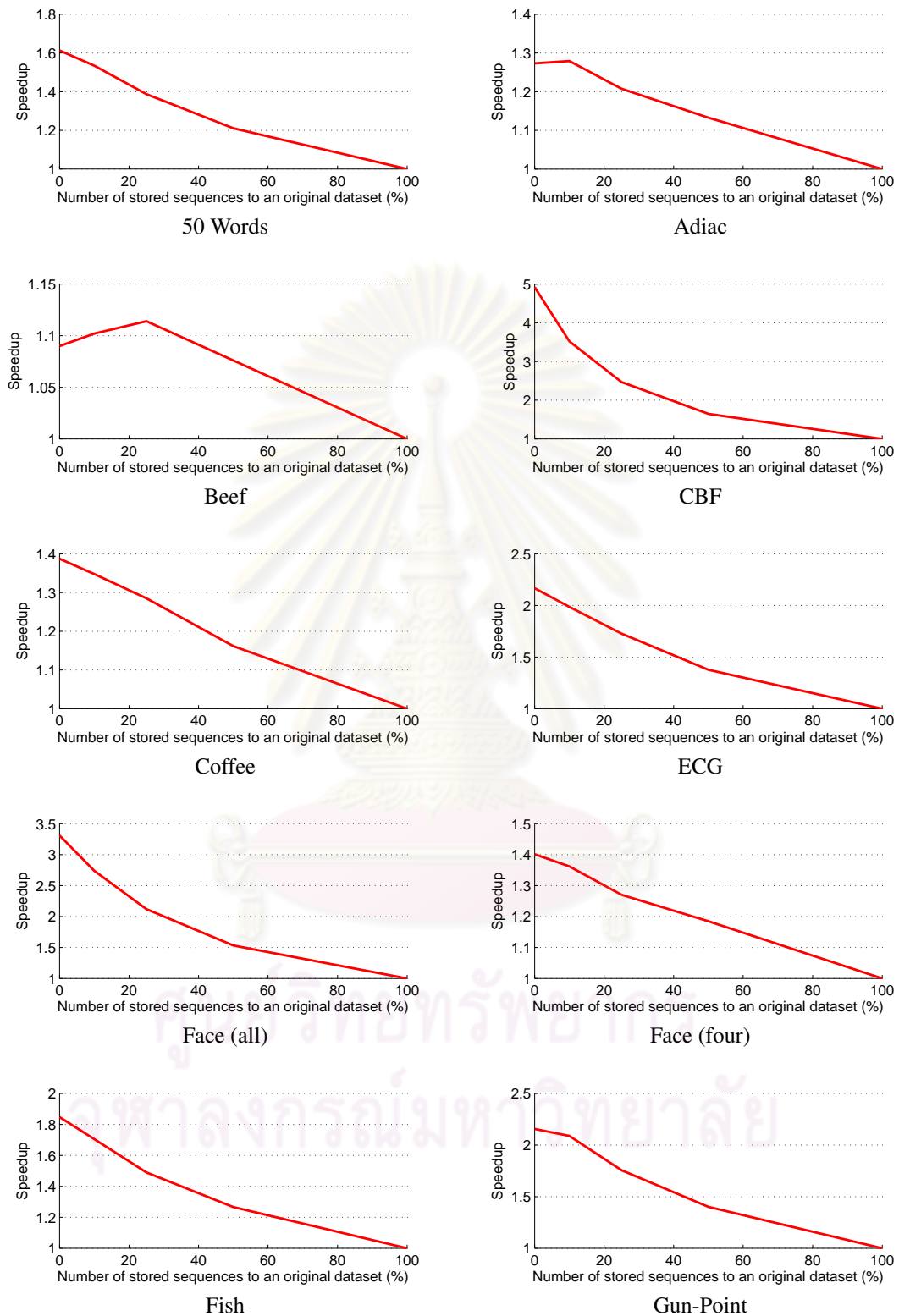


Figure F.7: Speedup of Incremental Shape-based Averaging with ICDTW when the number of stored sequences to an original dataset is varied.

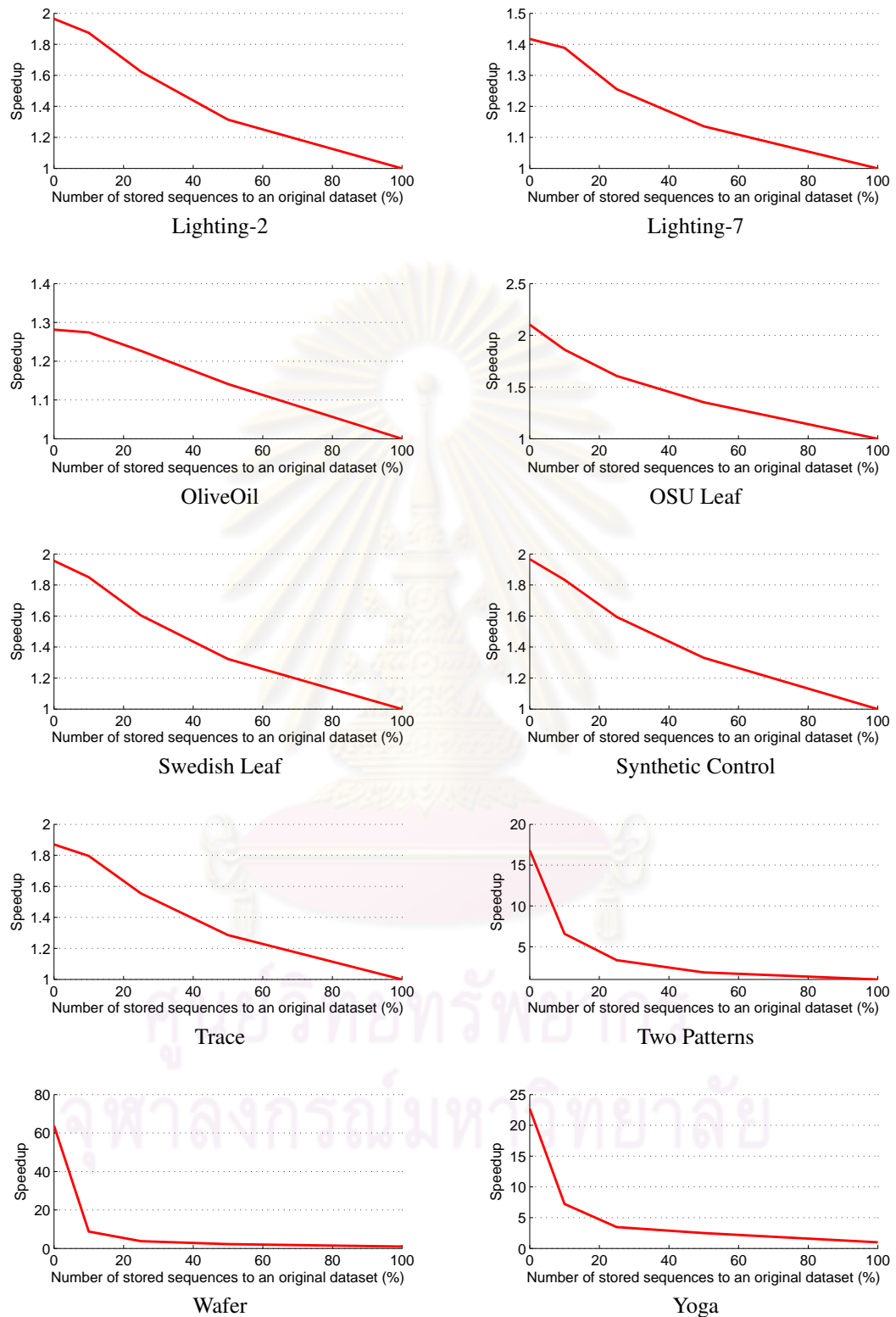


Figure F.8: Speedup of Incremental Shape-based Averaging with CDTW when the number of stored sequences to an original dataset is varied. (cont.)



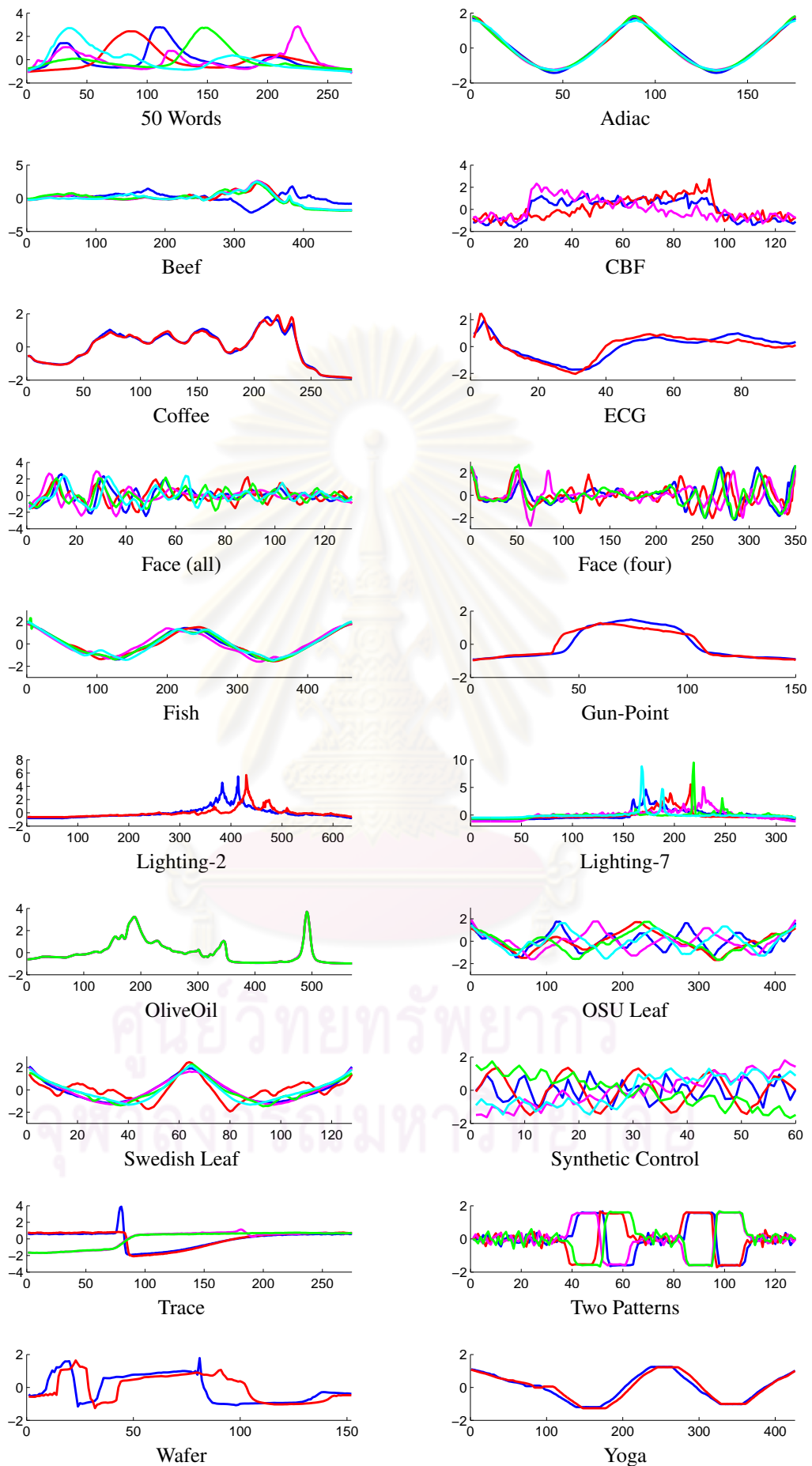


Figure F.9: Averaged results of some classes from Incremental Shape-based Averaging with CDTW when  $\alpha = 1$ .

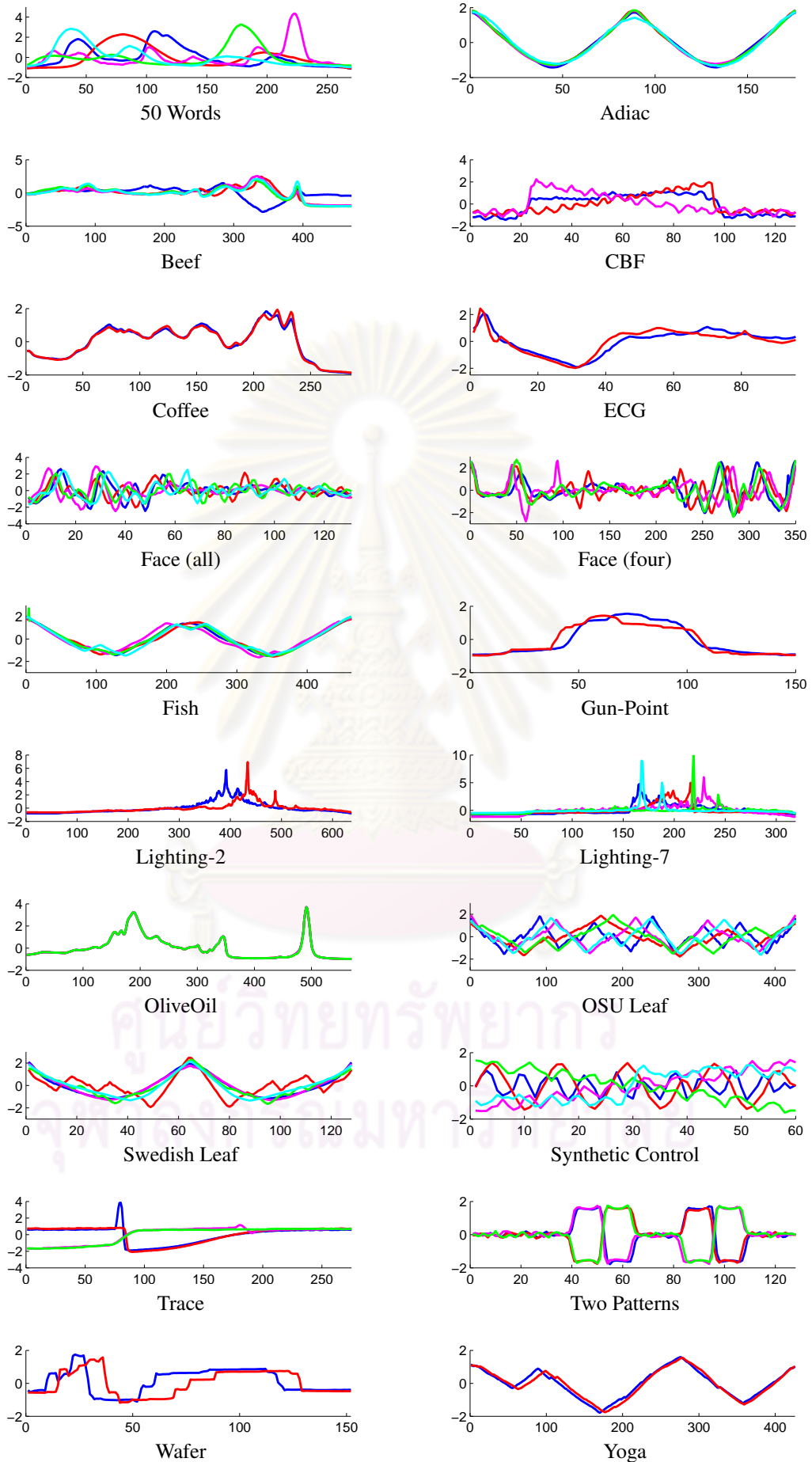


Figure F.10: Averaged results of some classes from Incremental Shape-based Averaging with CDTW when  $\alpha$  is 25% of total number of each class.

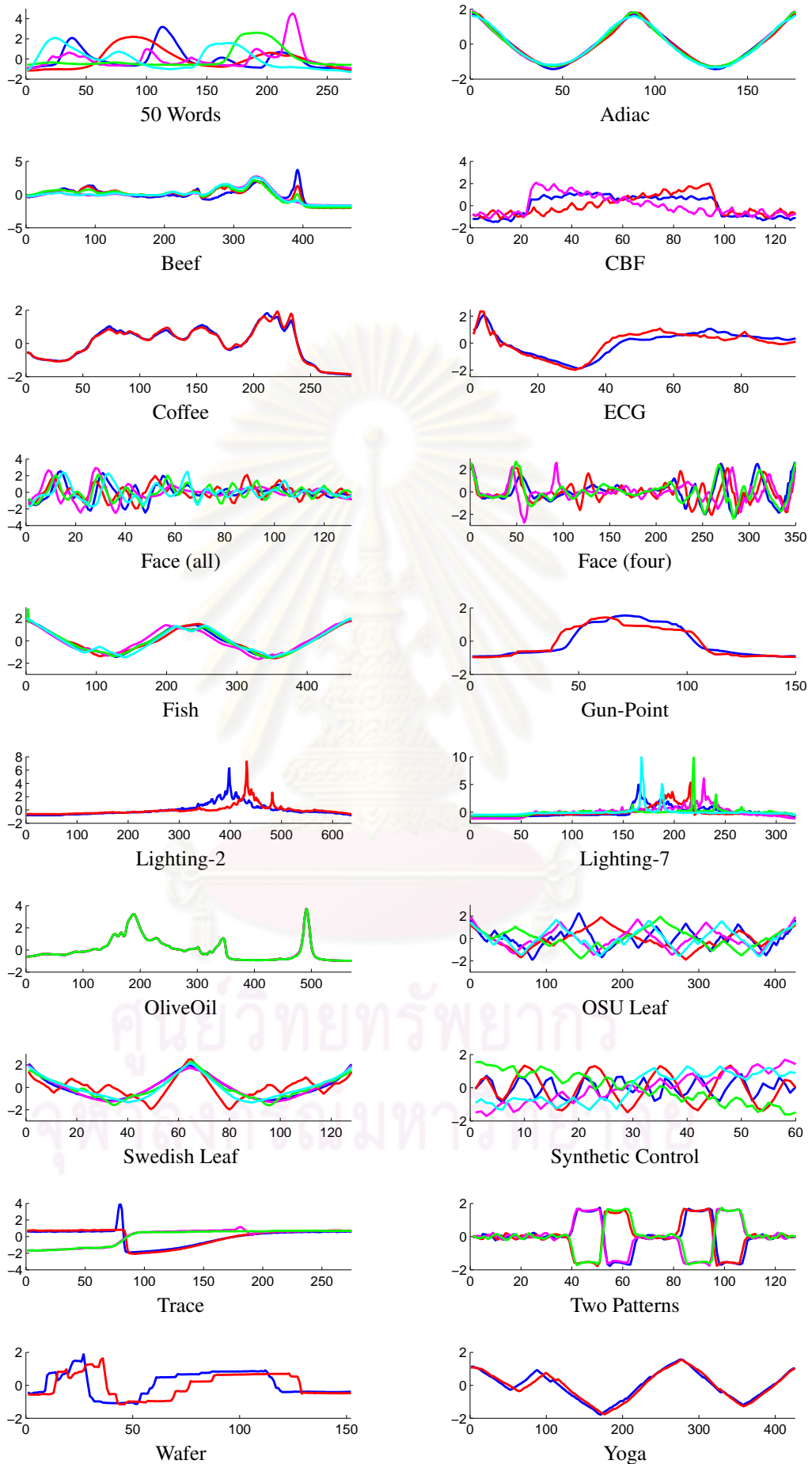


Figure F.11: Averaged results of some classes from Incremental Shape-based Averaging with CDTW when  $\alpha$  is 50% of total number of each class.

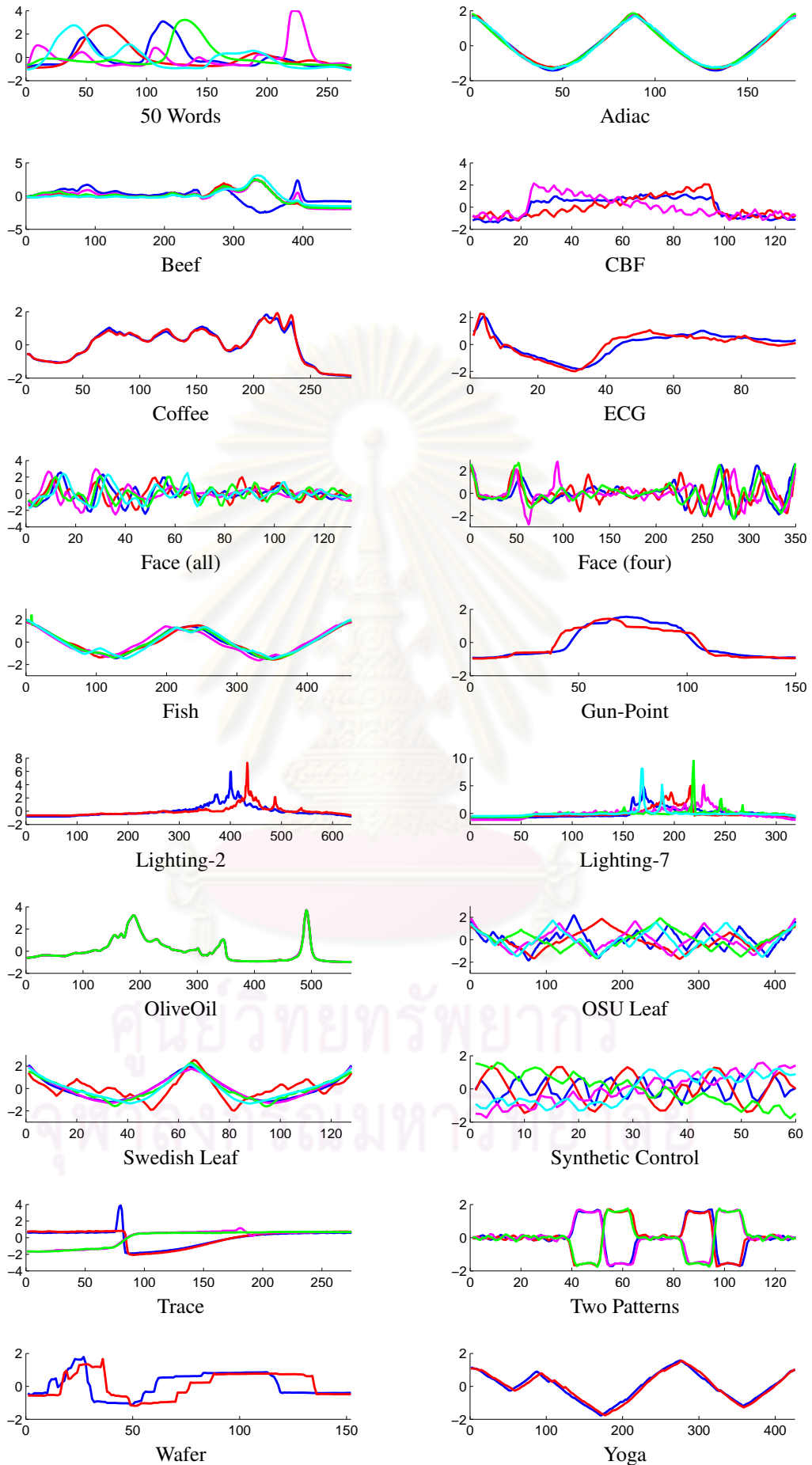


Figure F.12: Averaged results of some classes from Incremental Shape-based Averaging with CDTW when  $\alpha$  is 100% of total number of each class.

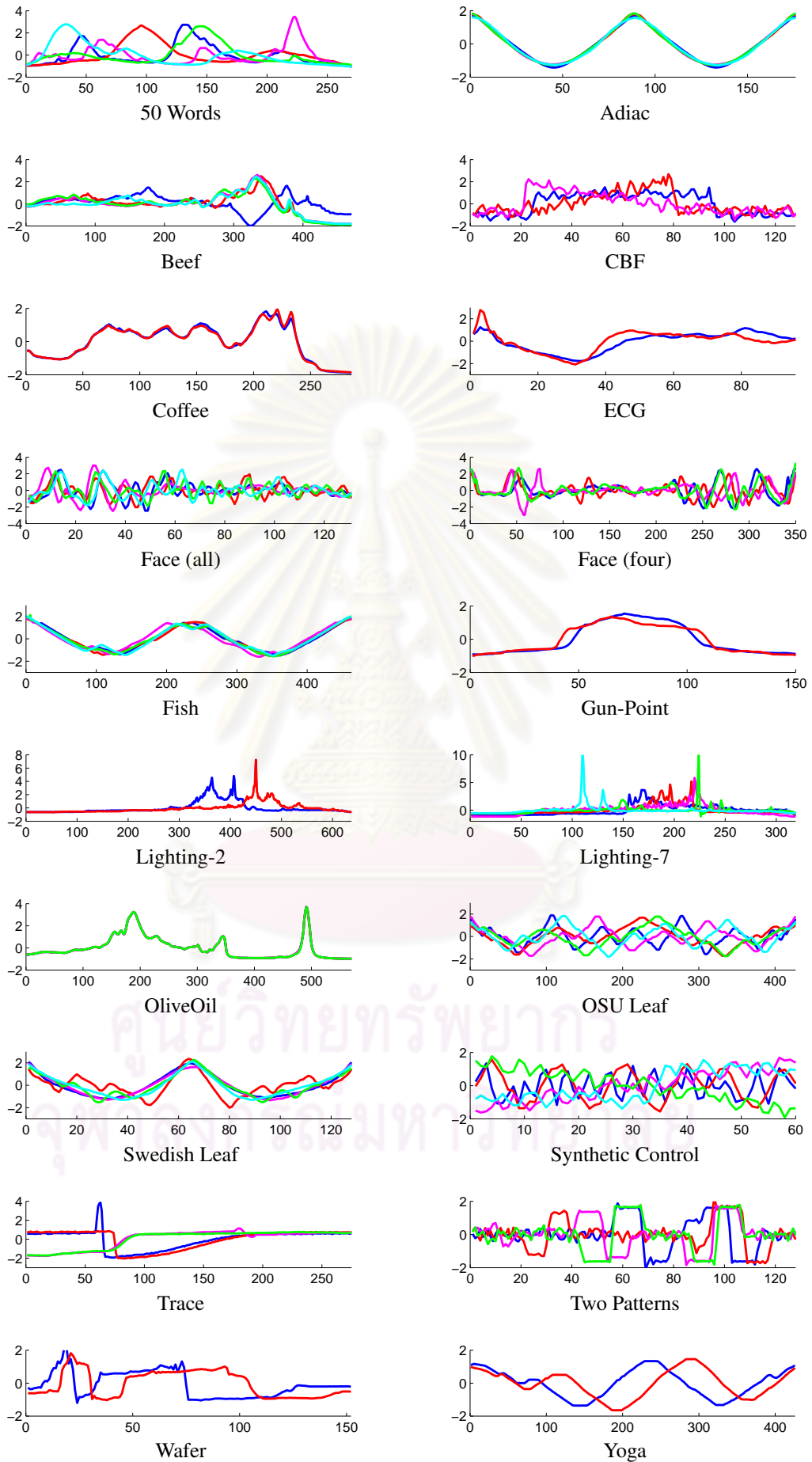


Figure F.13: Averaged results of some classes from Incremental Shape-based Averaging with ICDTW when  $\alpha = 1$ .

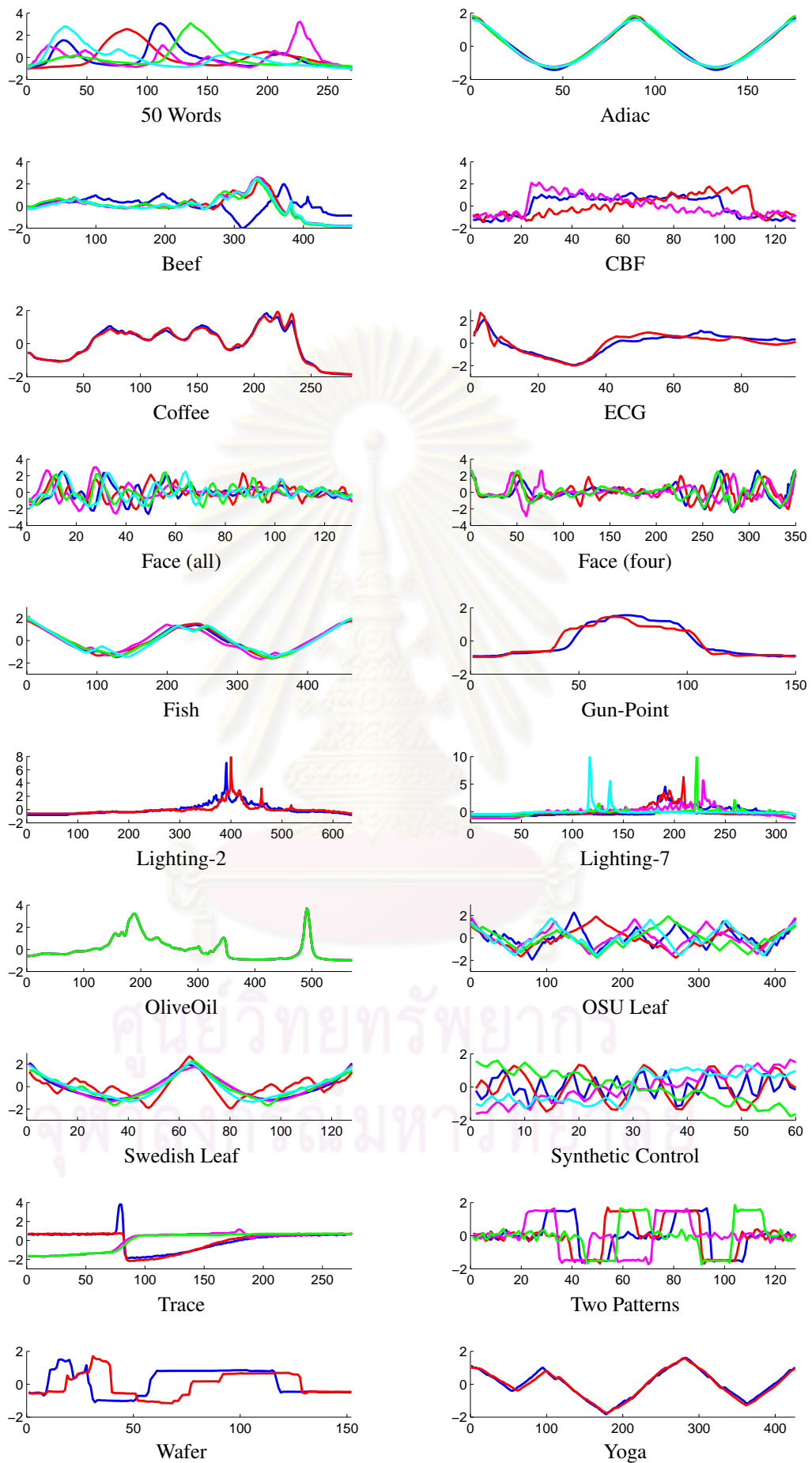


Figure F.14: Averaged results of some classes from Incremental Shape-based Averaging with ICDTW when  $\alpha$  is 25% of total number of each class.

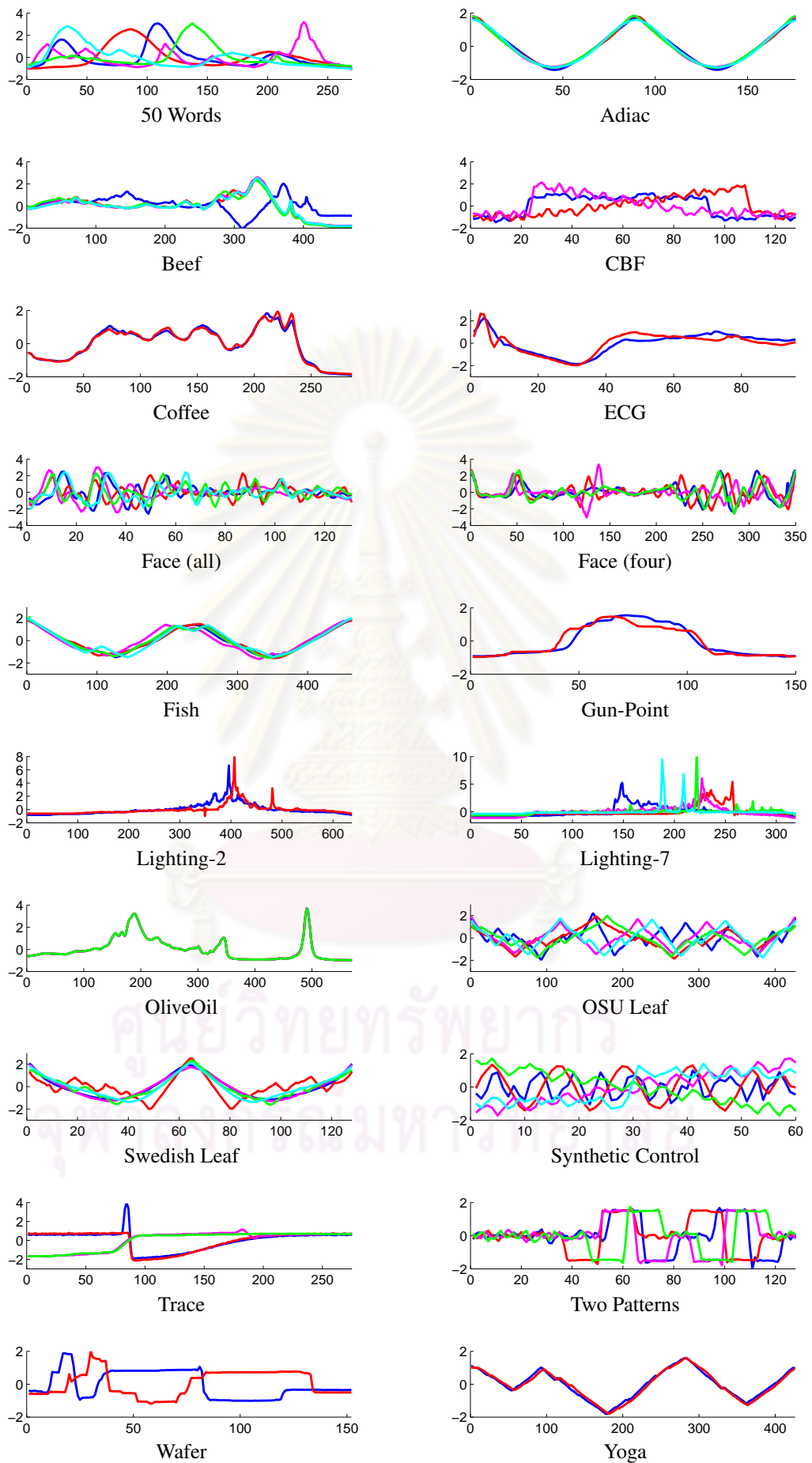


Figure F.15: Averaged results of some classes from Incremental Shape-based Averaging with ICDTW when  $\alpha$  is 50% of total number of each class.

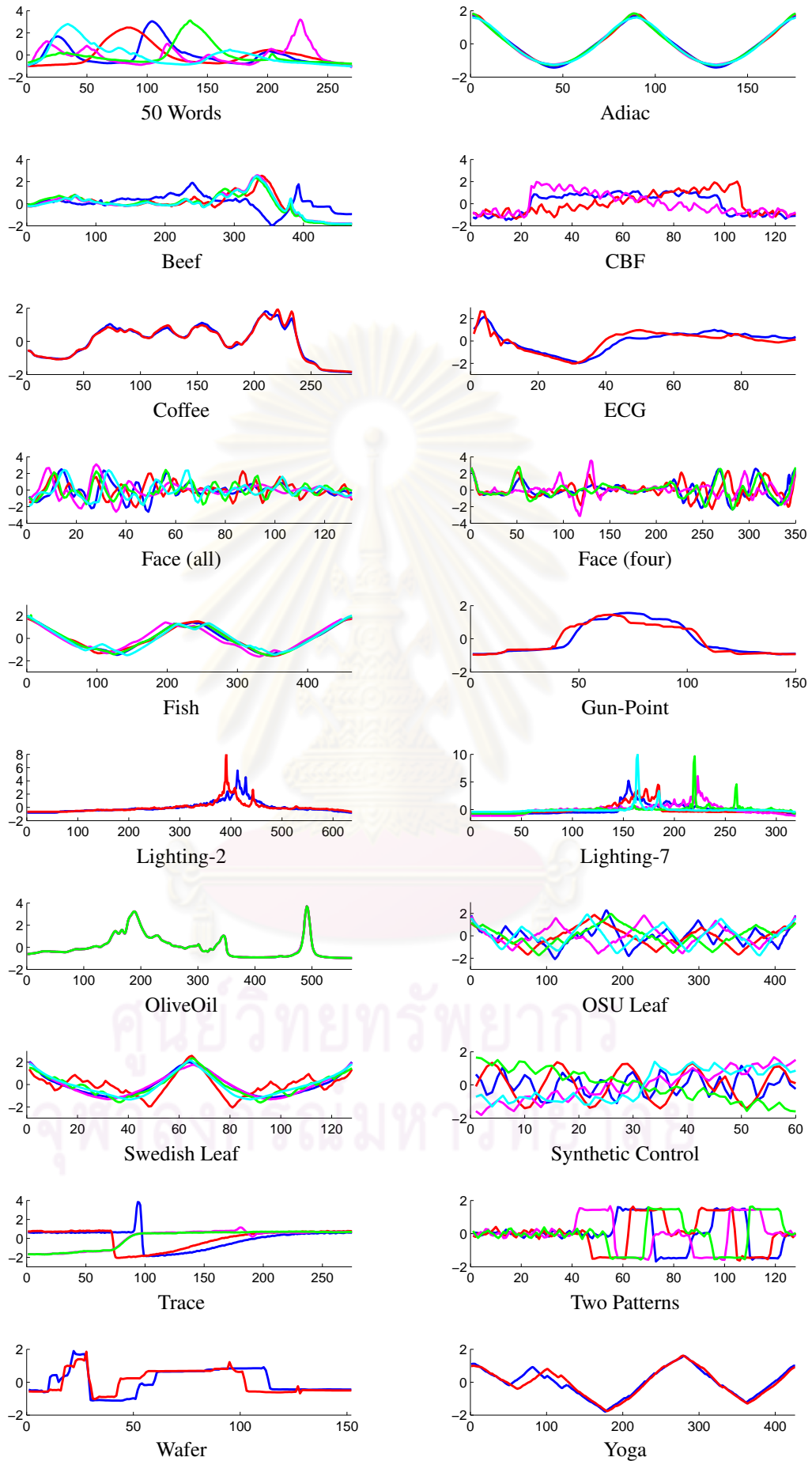


Figure F.16: Averaged results of some classes from Incremental Shape-based Averaging with ICDTW when  $\alpha$  is 100% of total number of each class.



**APPENDIX G**

**COMPLETE EXPERIMENTAL RESULTS OF THE FIRST  
EXPERIMENT IN CHAPTER 6**



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

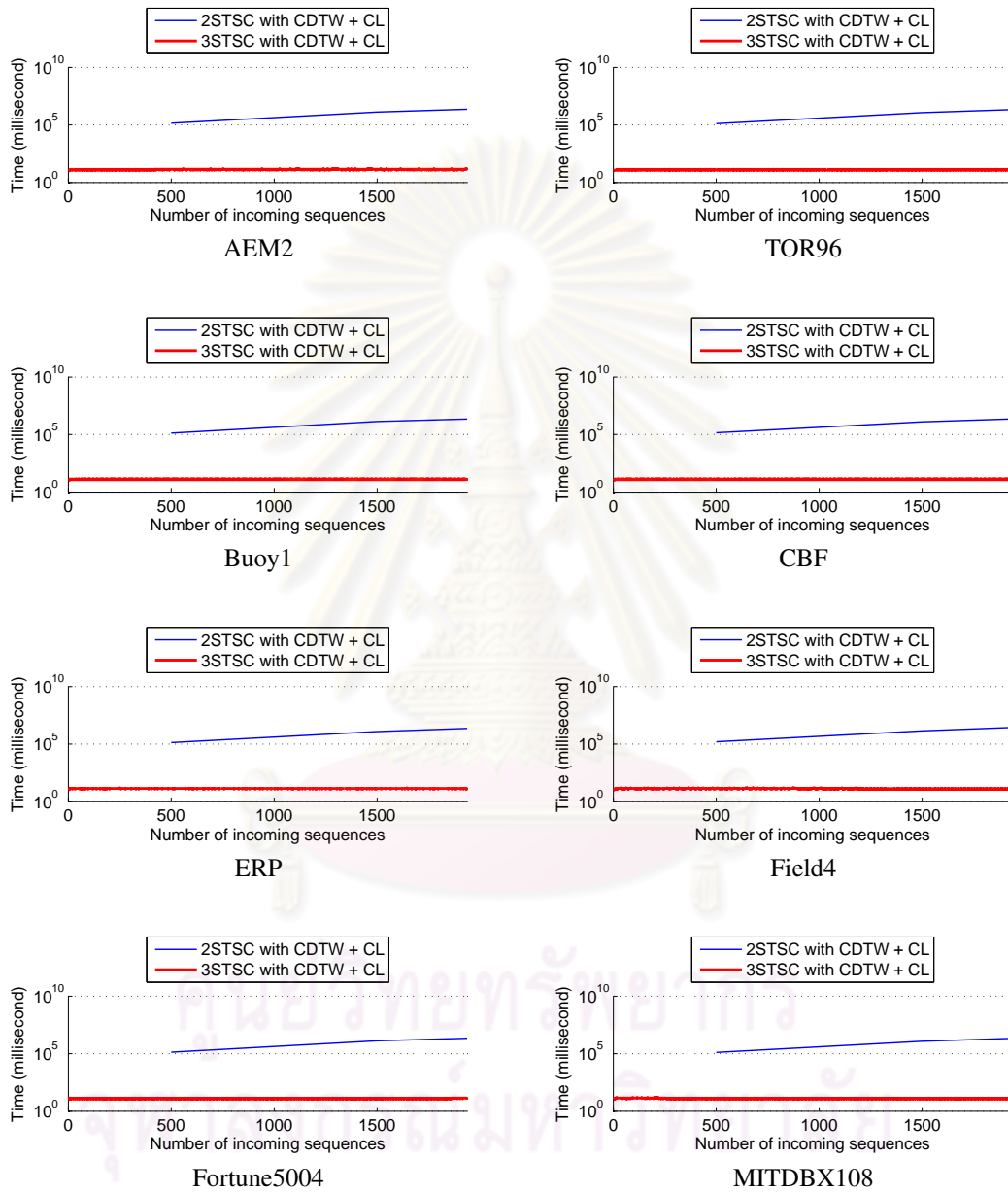


Figure G.1: Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where  $k = 3$  and  $w = 64$ .

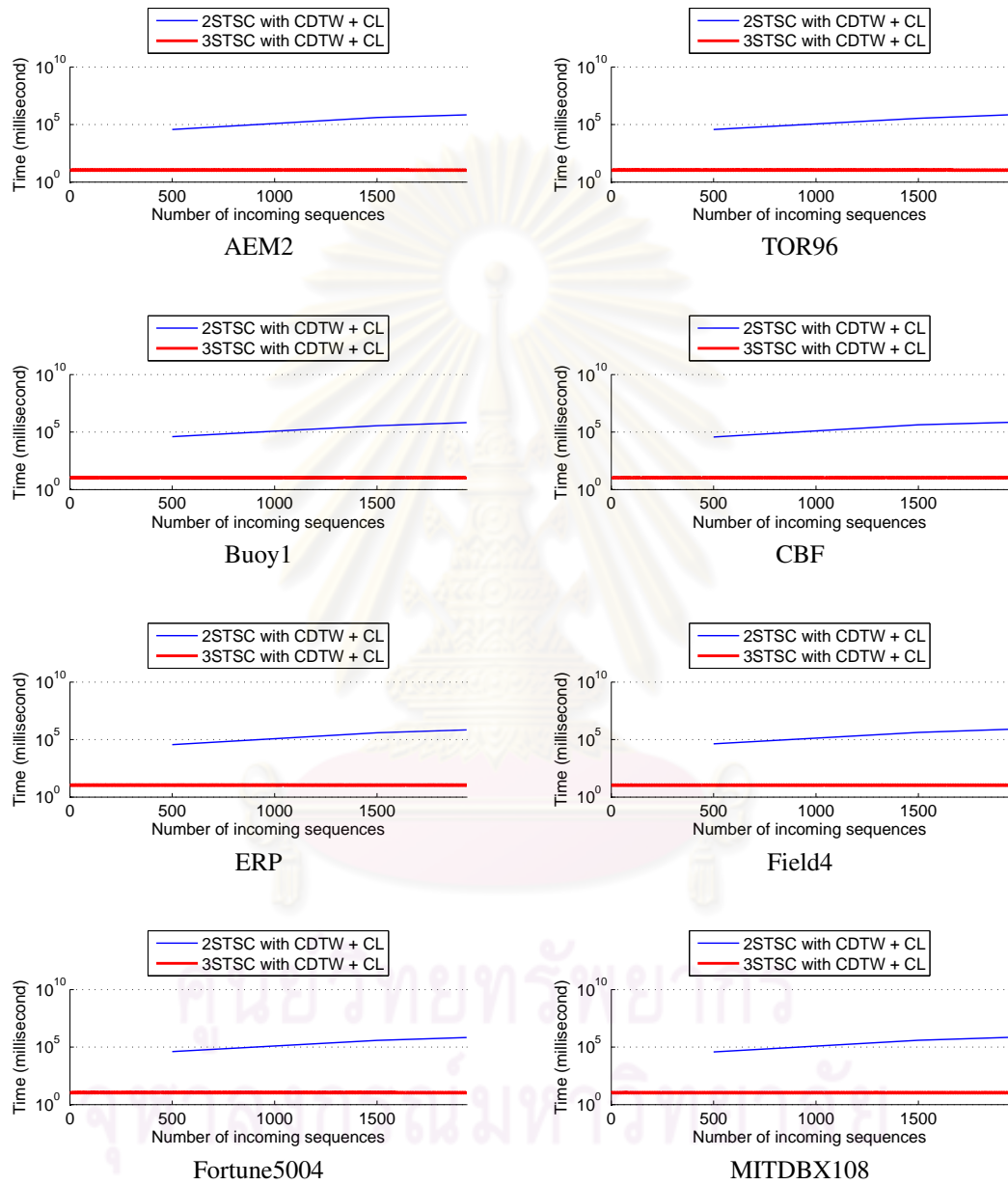


Figure G.2: Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where  $k = 3$  and  $w = 32$

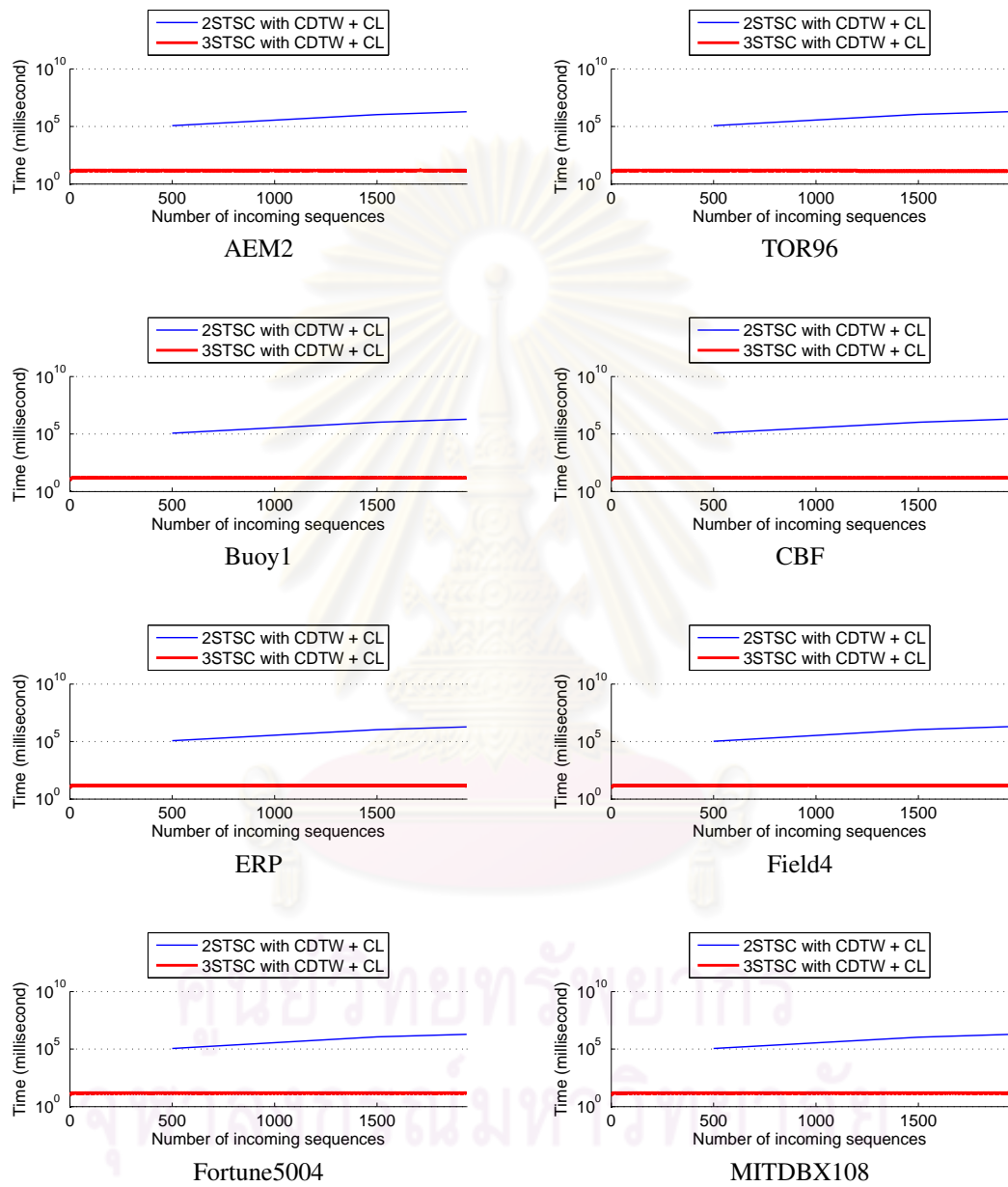


Figure G.3: Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where  $k = 5$  and  $w = 64$ .

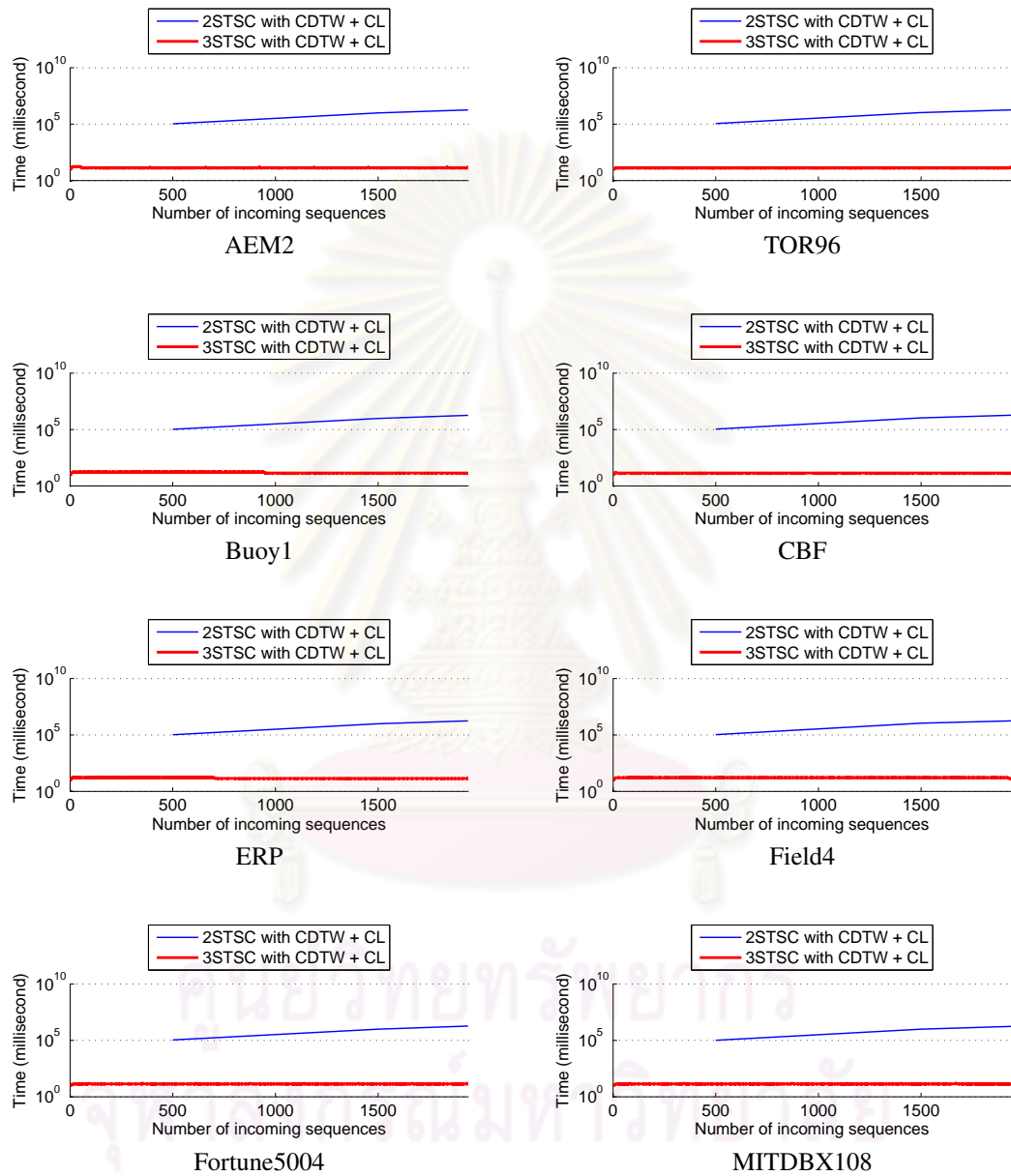


Figure G.4: Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where  $k = 7$  and  $w = 64$ .

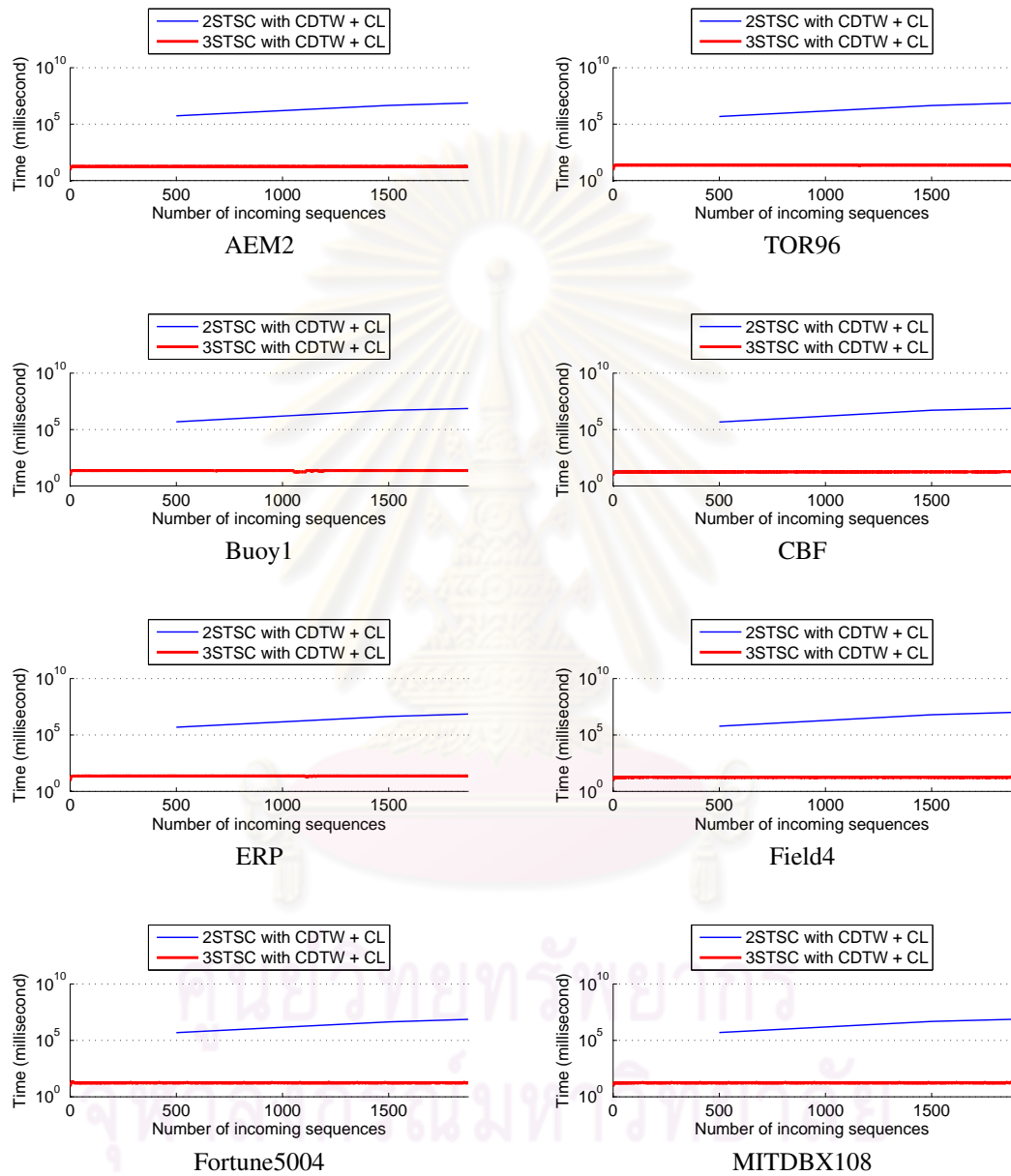


Figure G.5: Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where  $k = 3$  and  $w = 128$ .

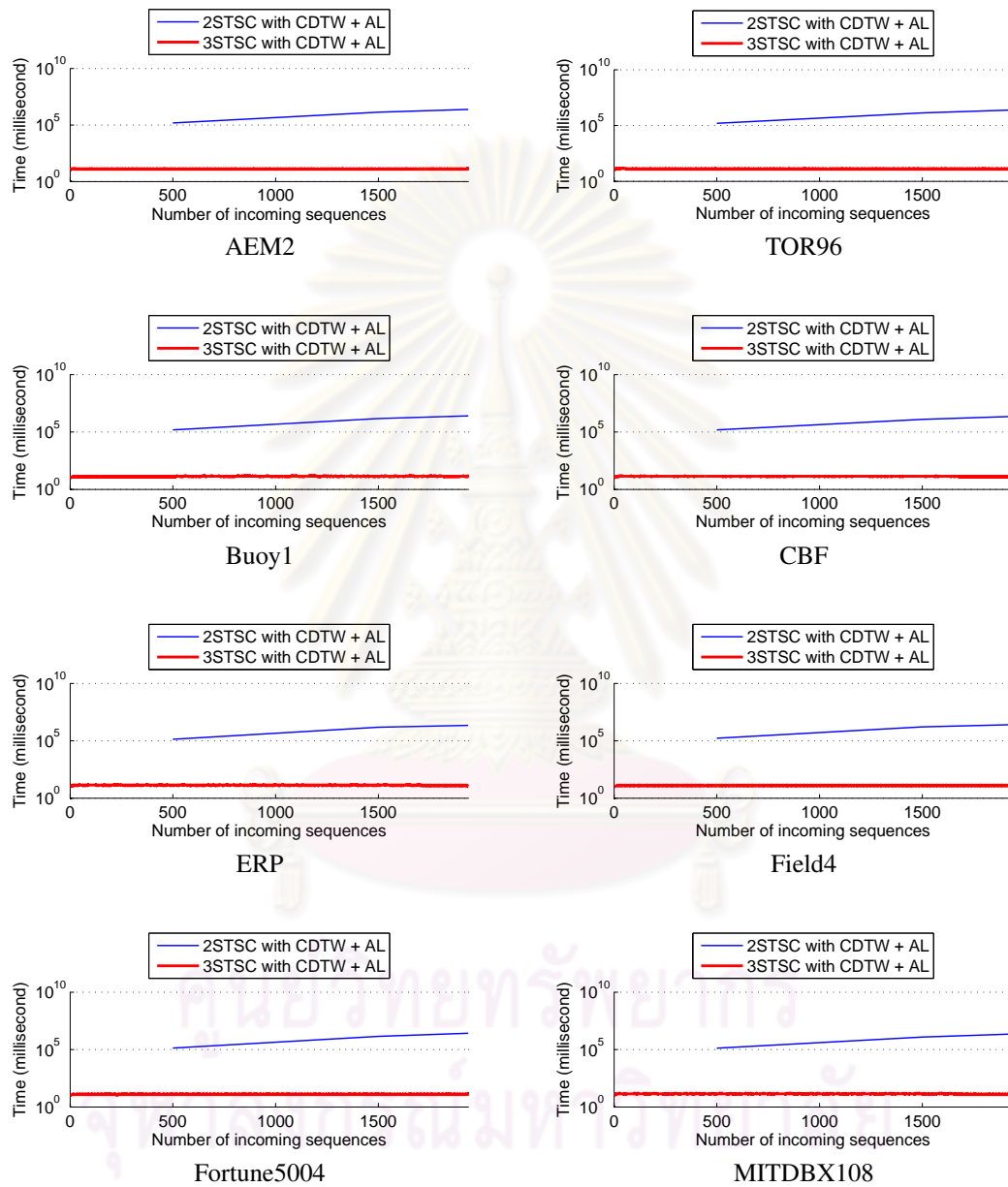


Figure G.6: Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where  $k = 3$  and  $w = 64$ .

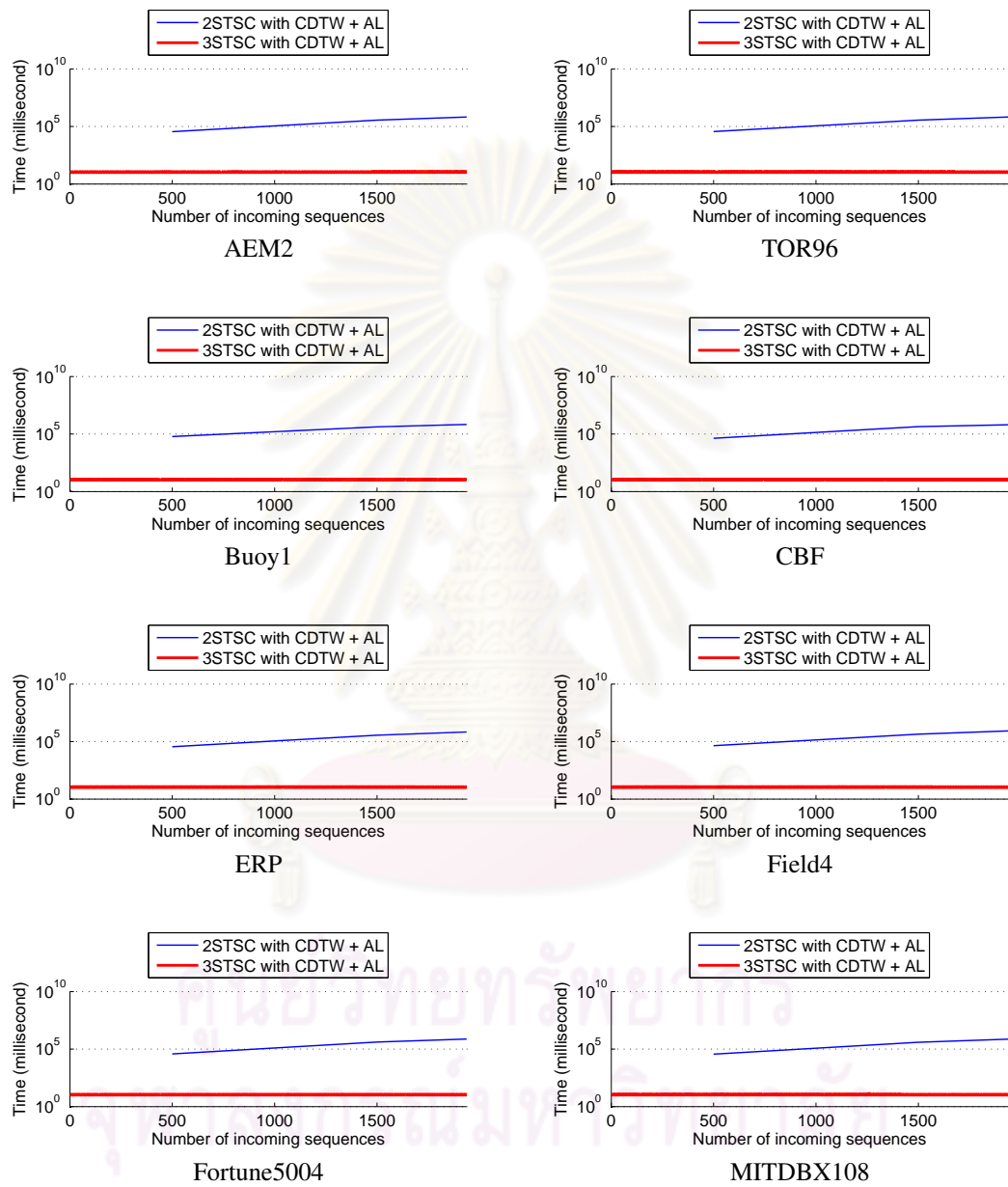


Figure G.7: Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where  $k = 3$  and  $w = 32$ .



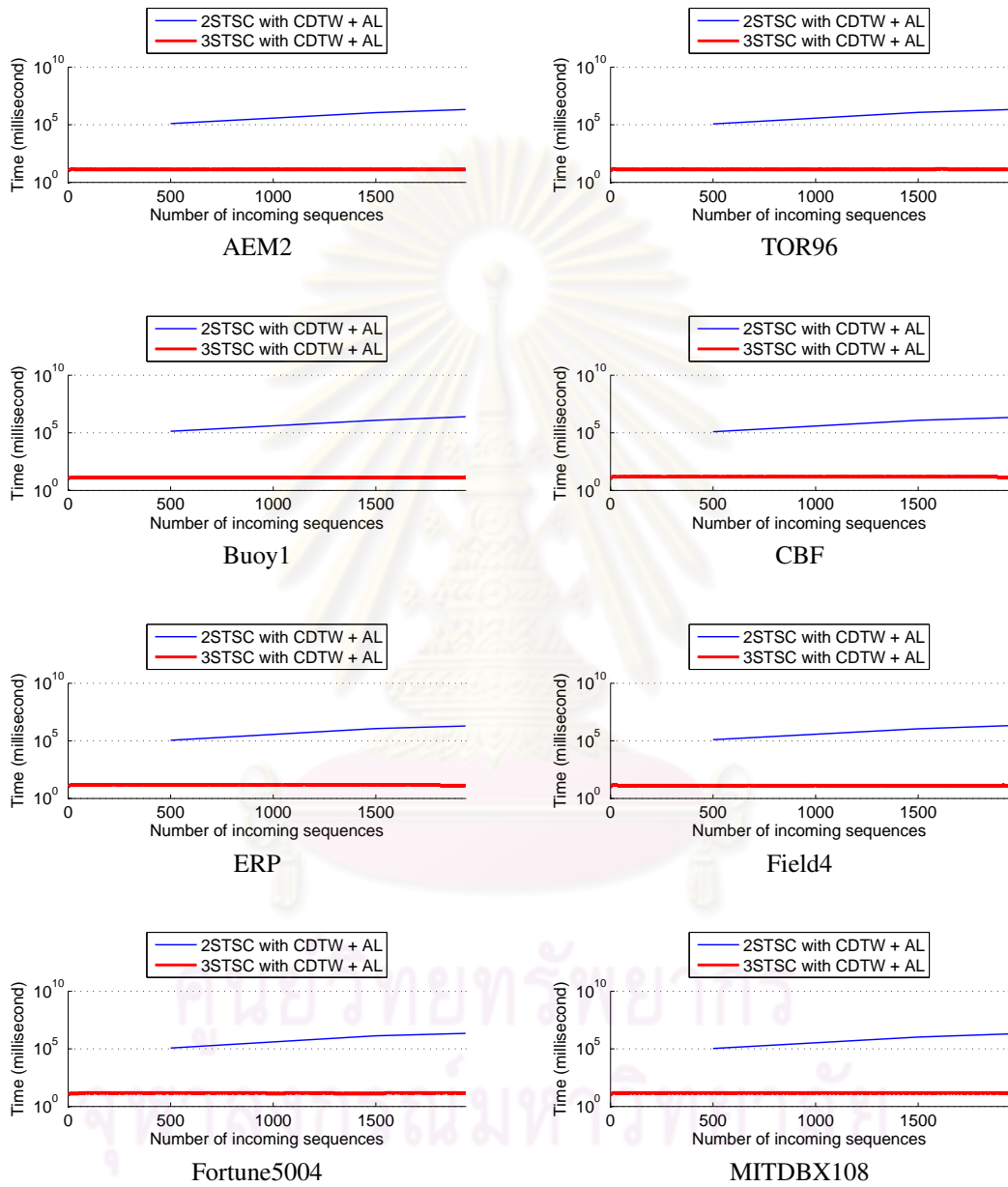


Figure G.8: Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where  $k = 5$  and  $w = 64$ .

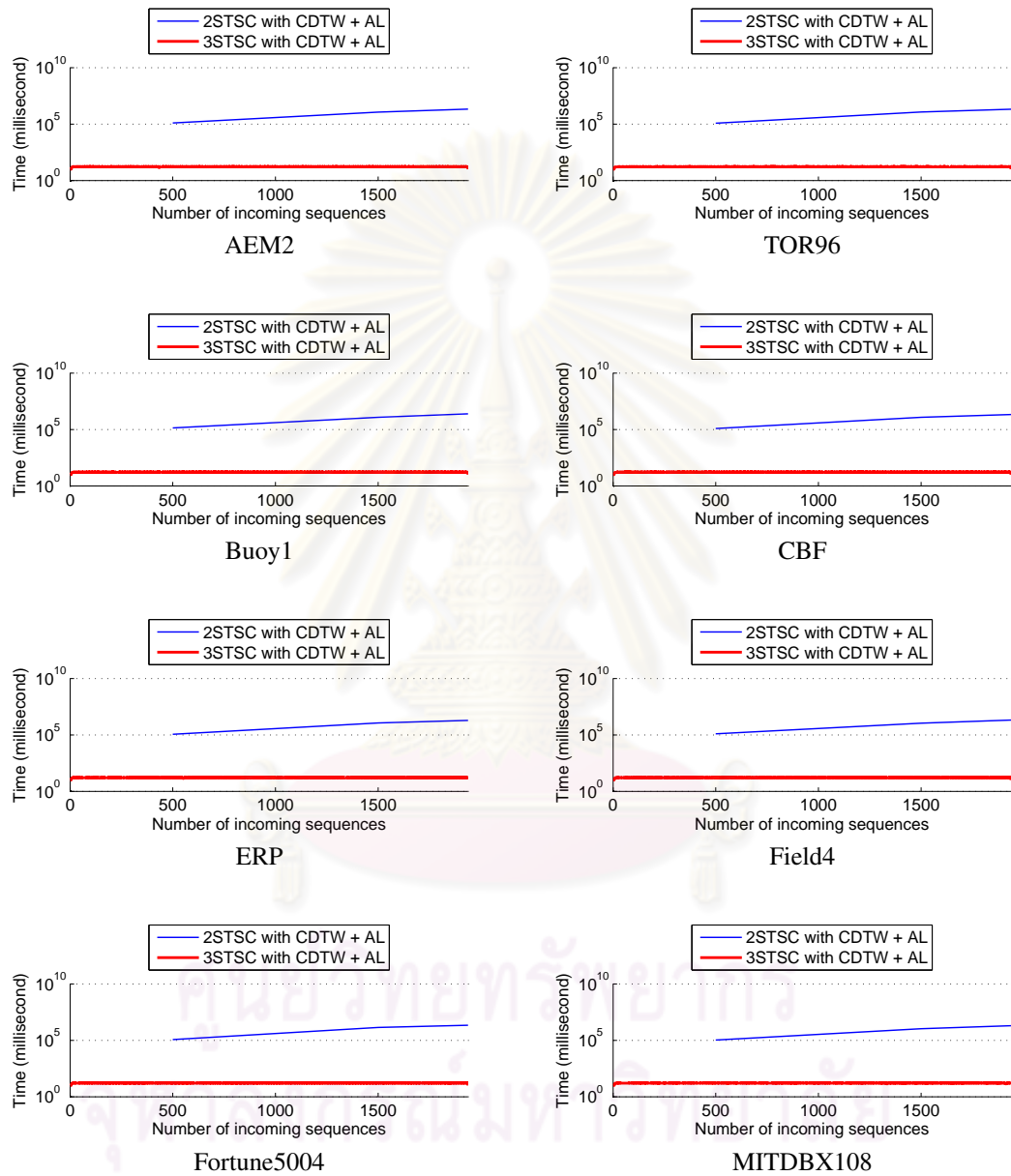


Figure G.9: Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where  $k = 7$  and  $w = 64$ .

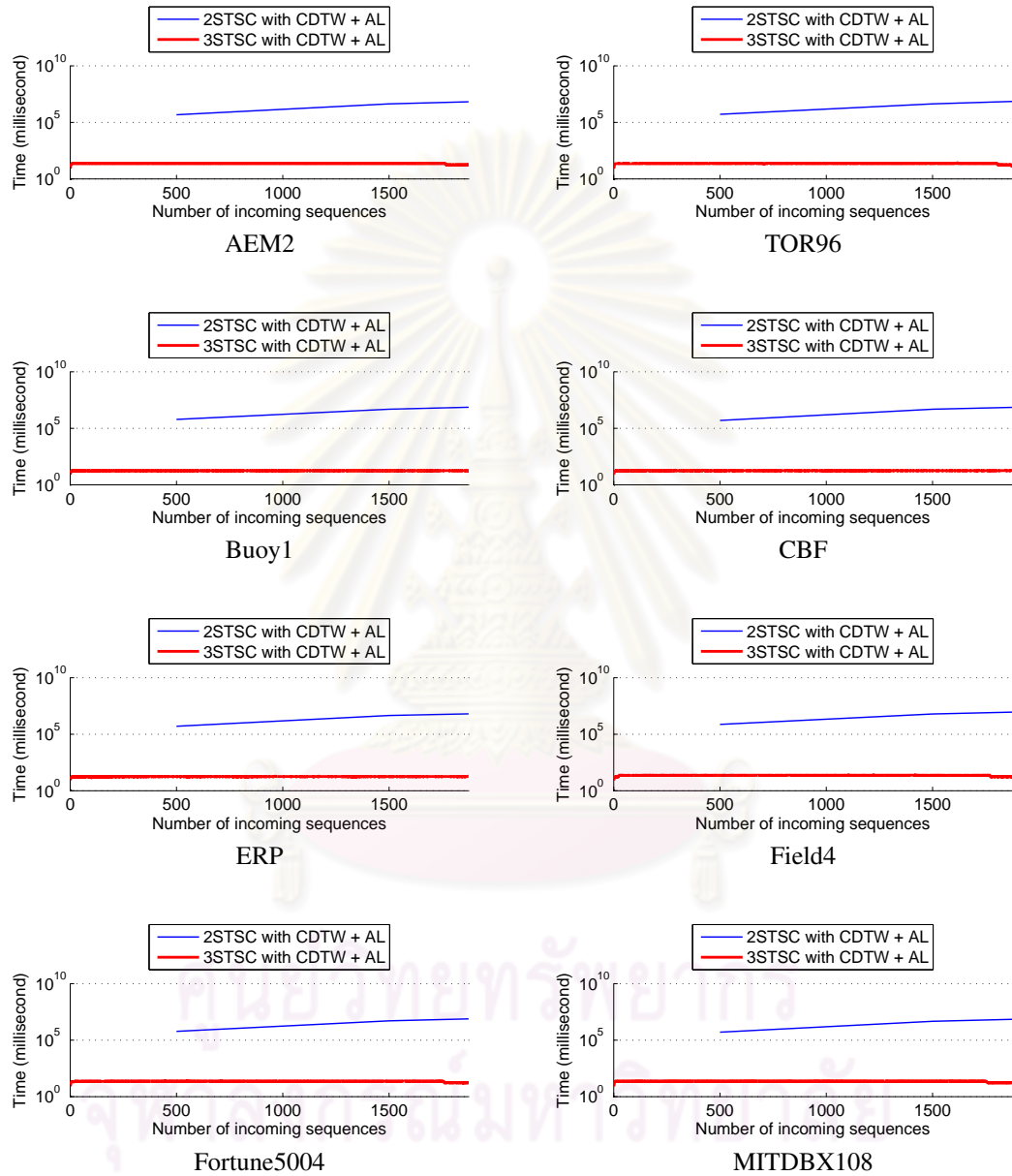


Figure G.10: Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where  $k = 3$  and  $w = 128$ .

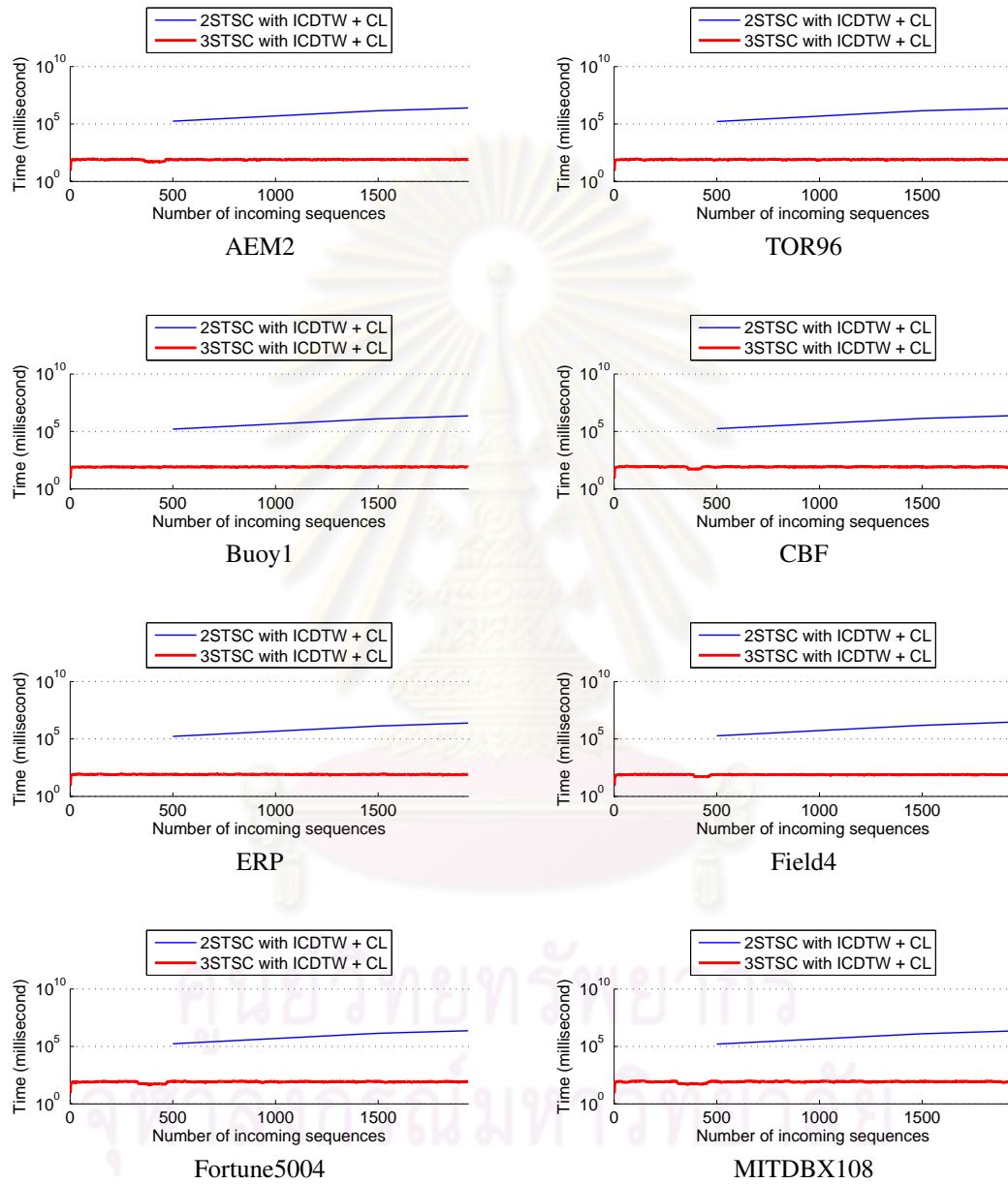


Figure G.11: Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where  $k = 3$  and  $w = 64$ .

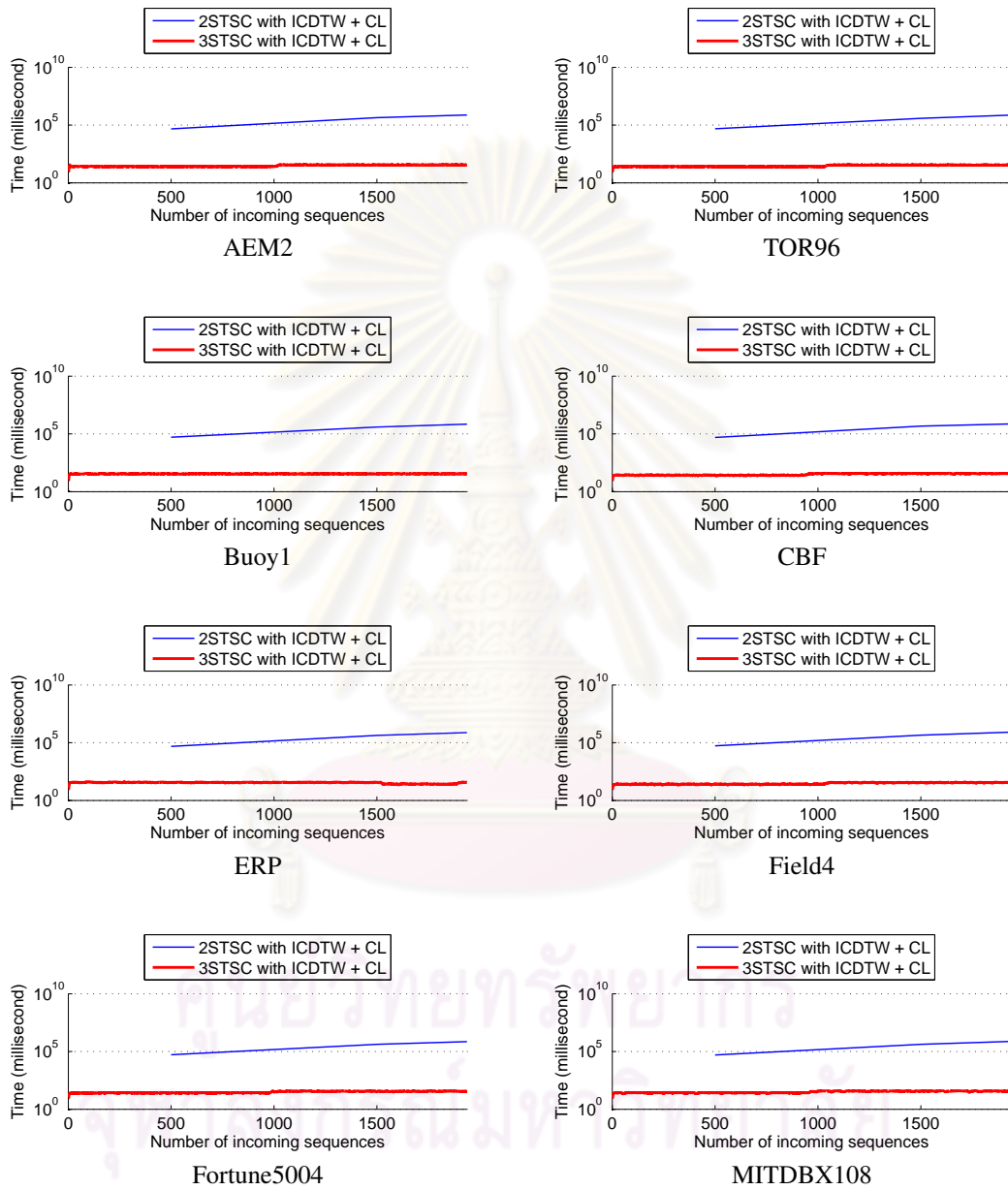


Figure G.12: Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where  $k = 3$  and  $w = 32$ .

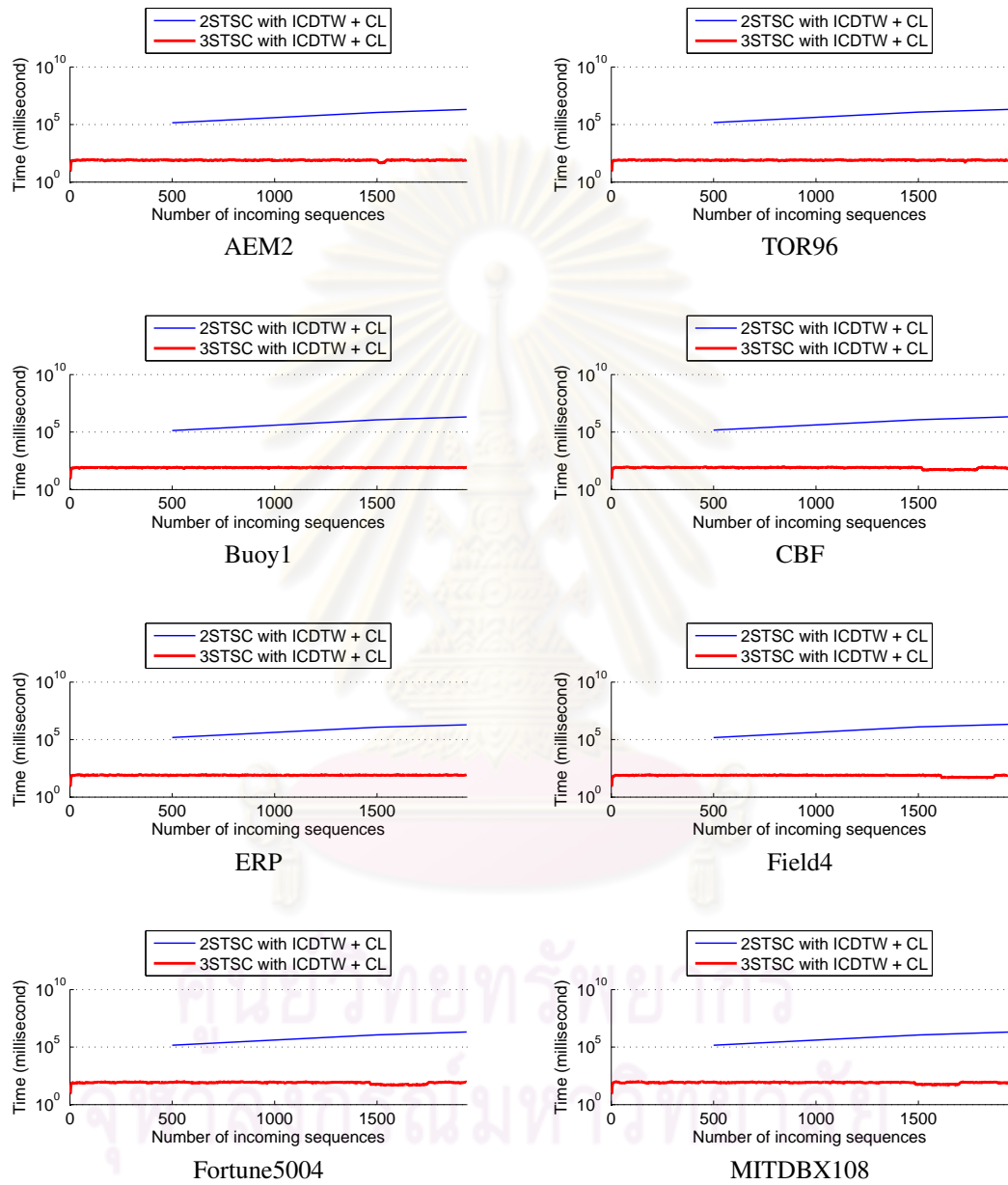


Figure G.13: Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where  $k = 5$  and  $w = 64$ .

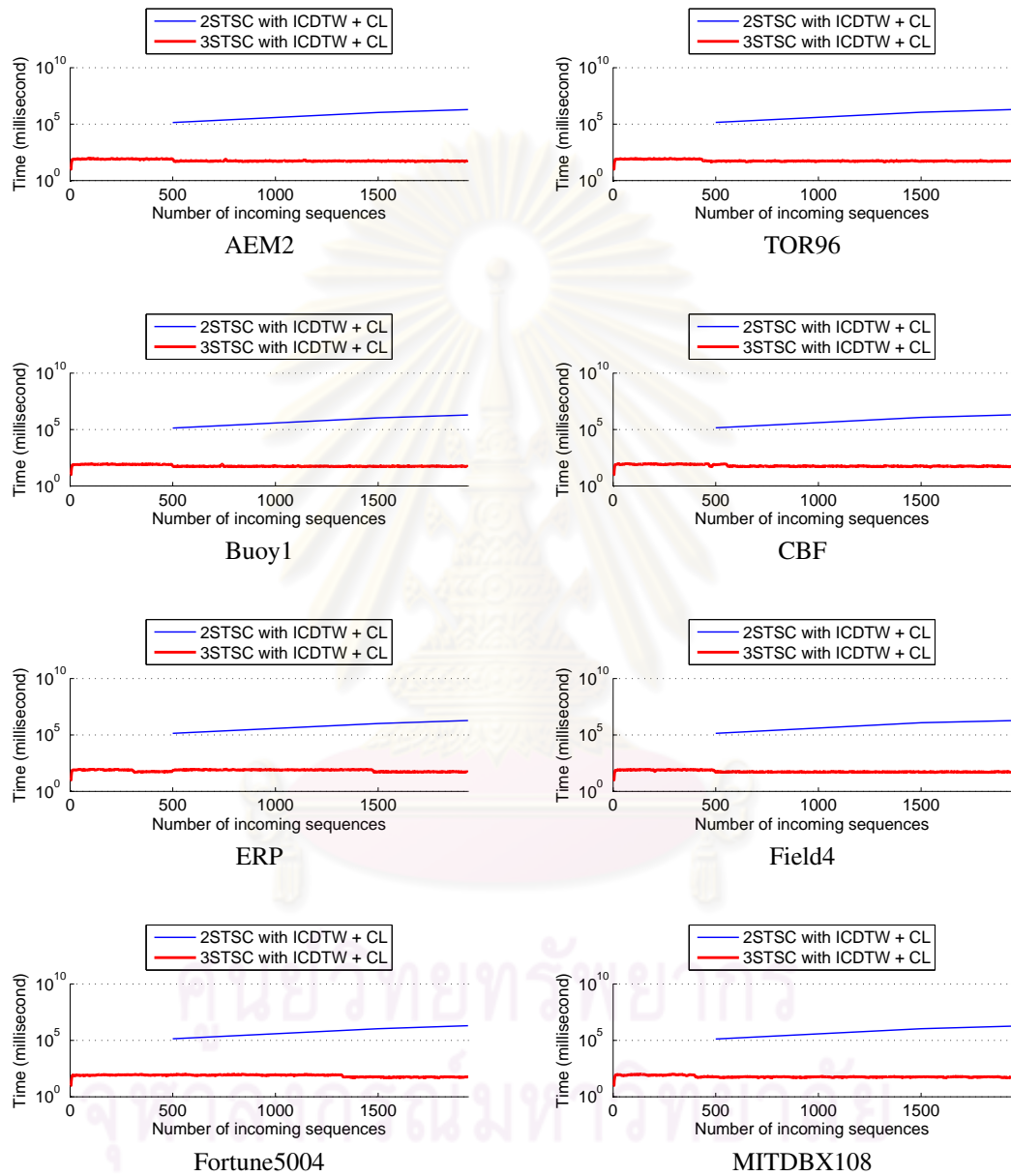


Figure G.14: Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where  $k = 7$  and  $w = 64$ .

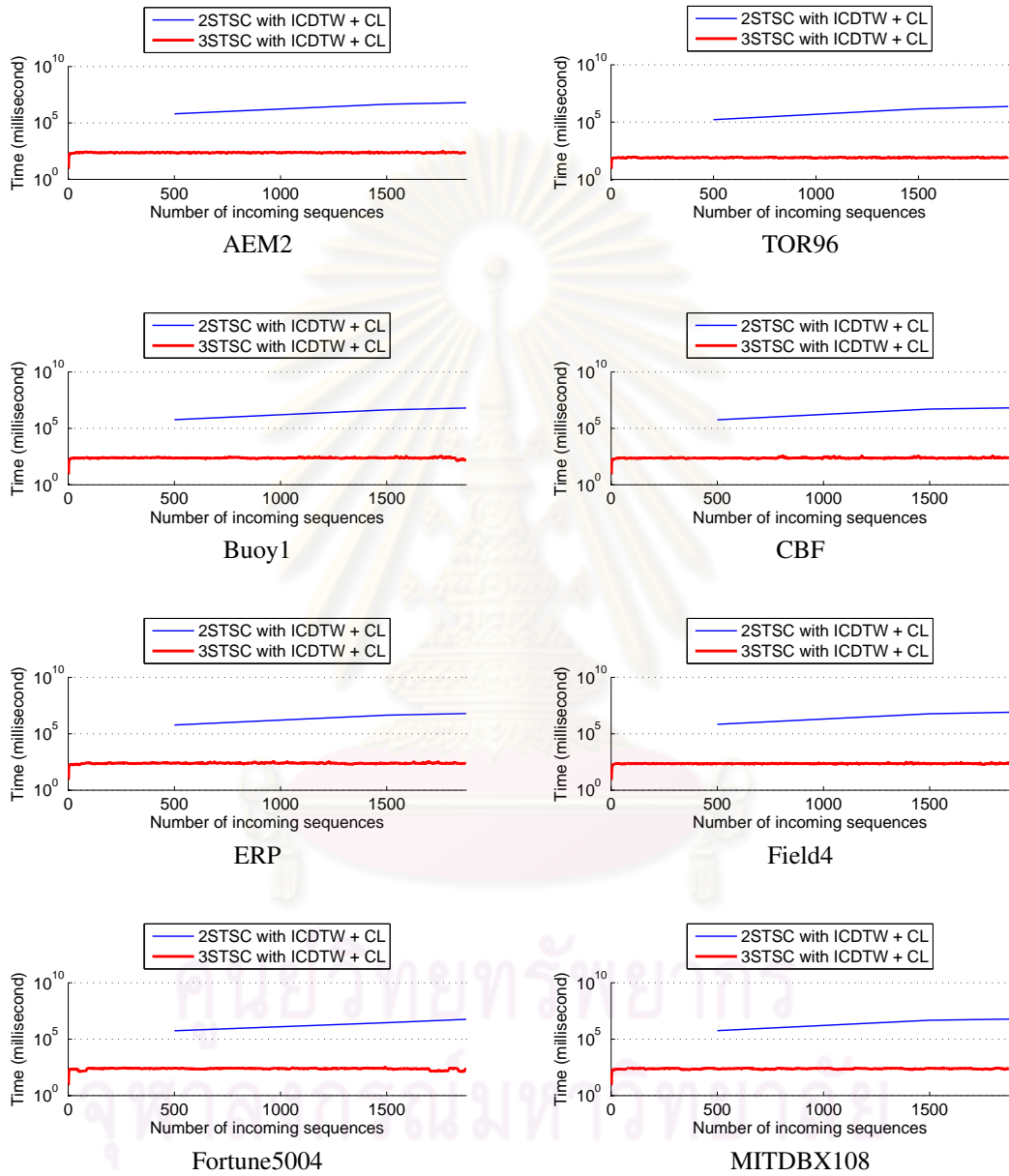


Figure G.15: Computational time of 3STSC and 2STSC with CDTW function and complete linkage when a new incoming sequence arrives, where  $k = 3$  and  $w = 128$ .



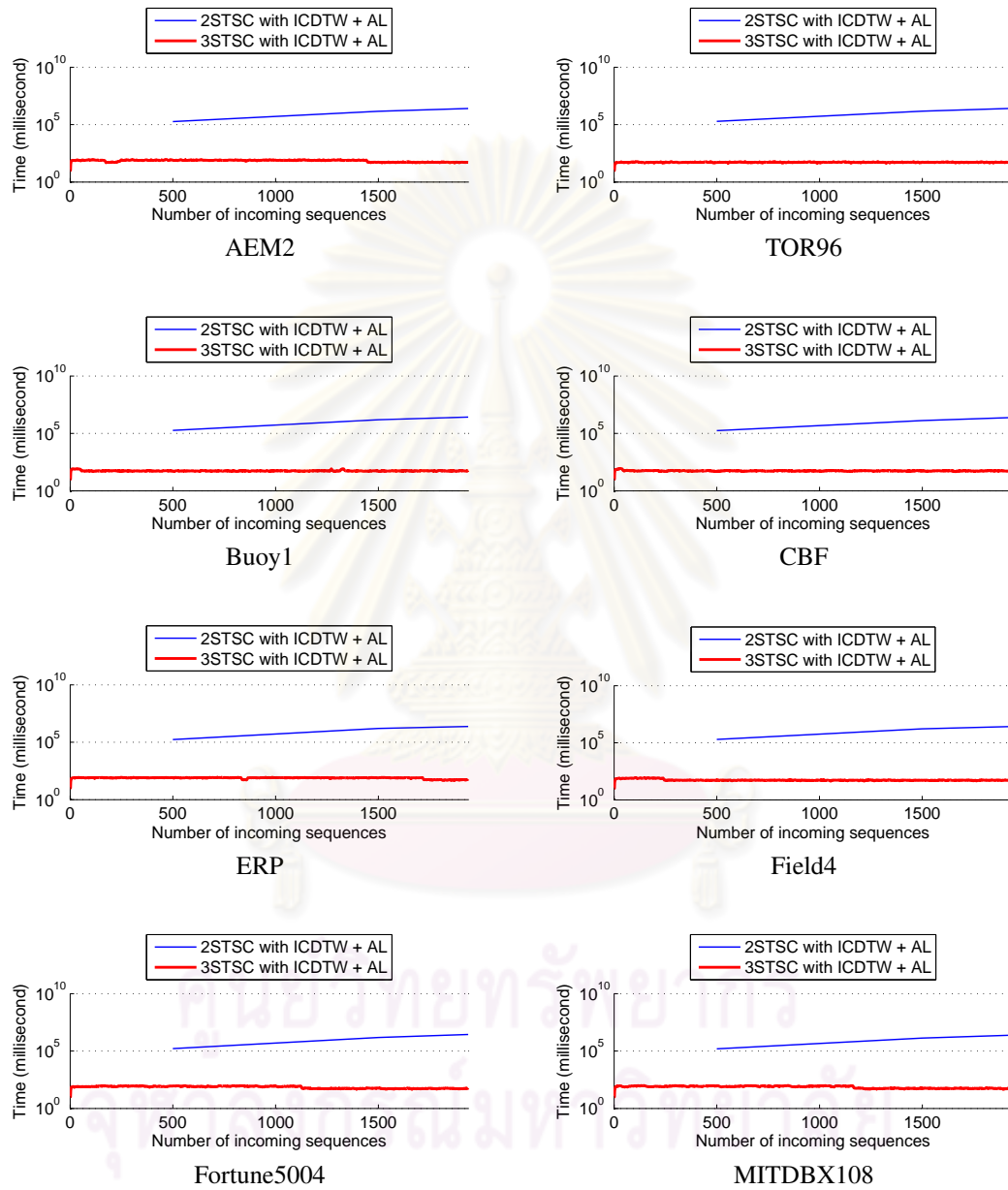


Figure G.16: Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where  $k = 3$  and  $w = 64$

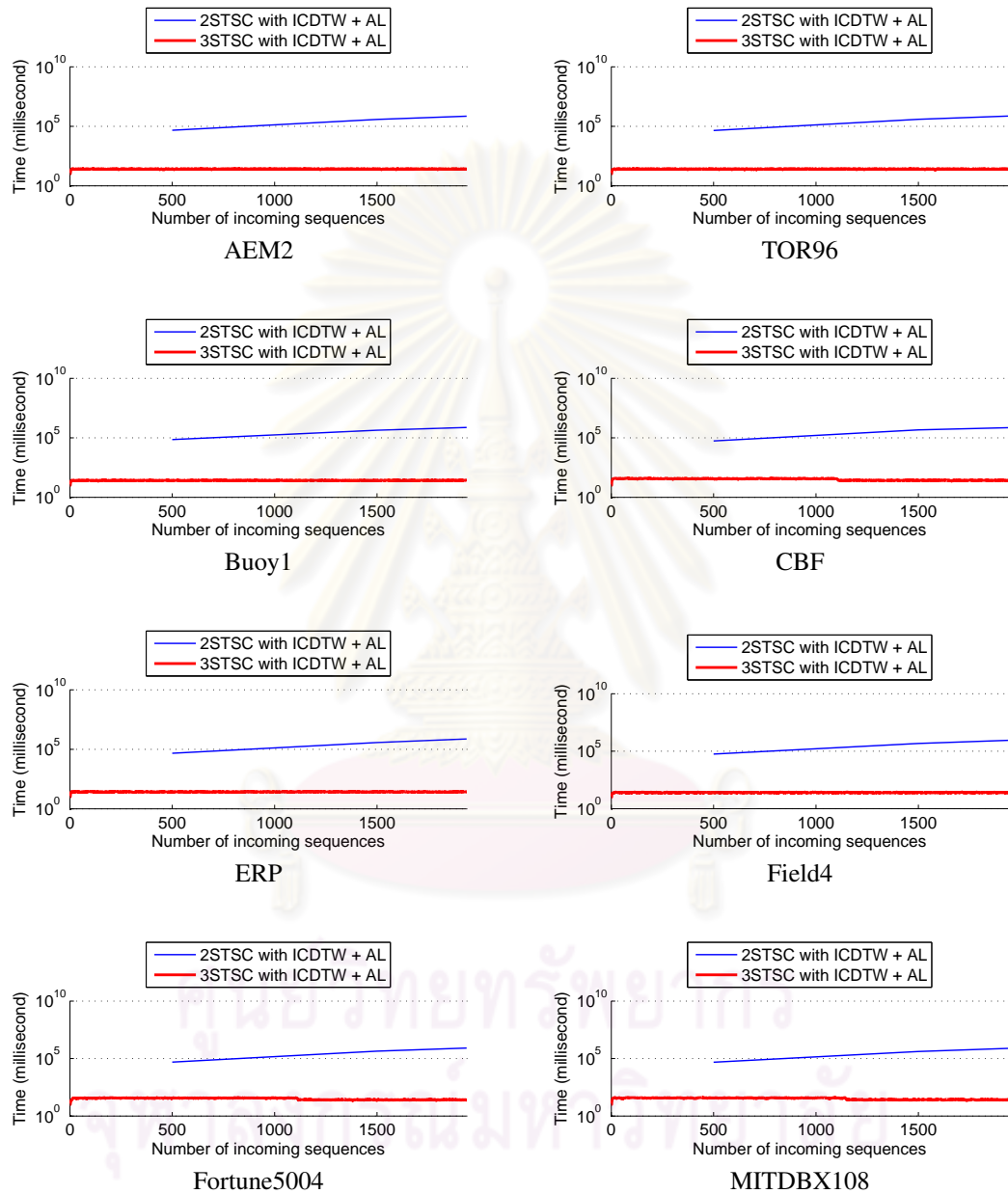


Figure G.17: Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where  $k = 3$  and  $w = 32$ .

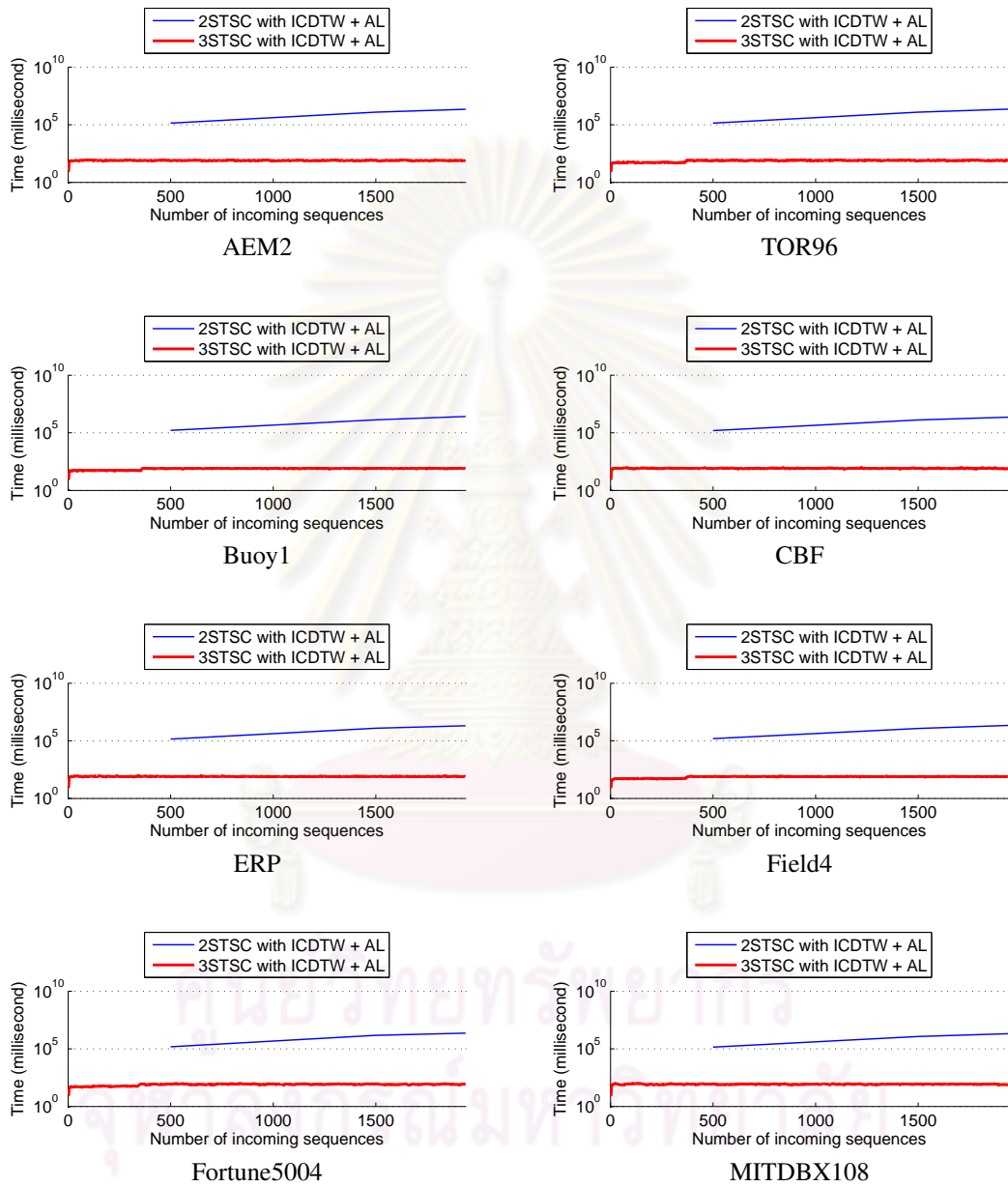


Figure G.18: Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where  $k = 5$  and  $w = 64$ .

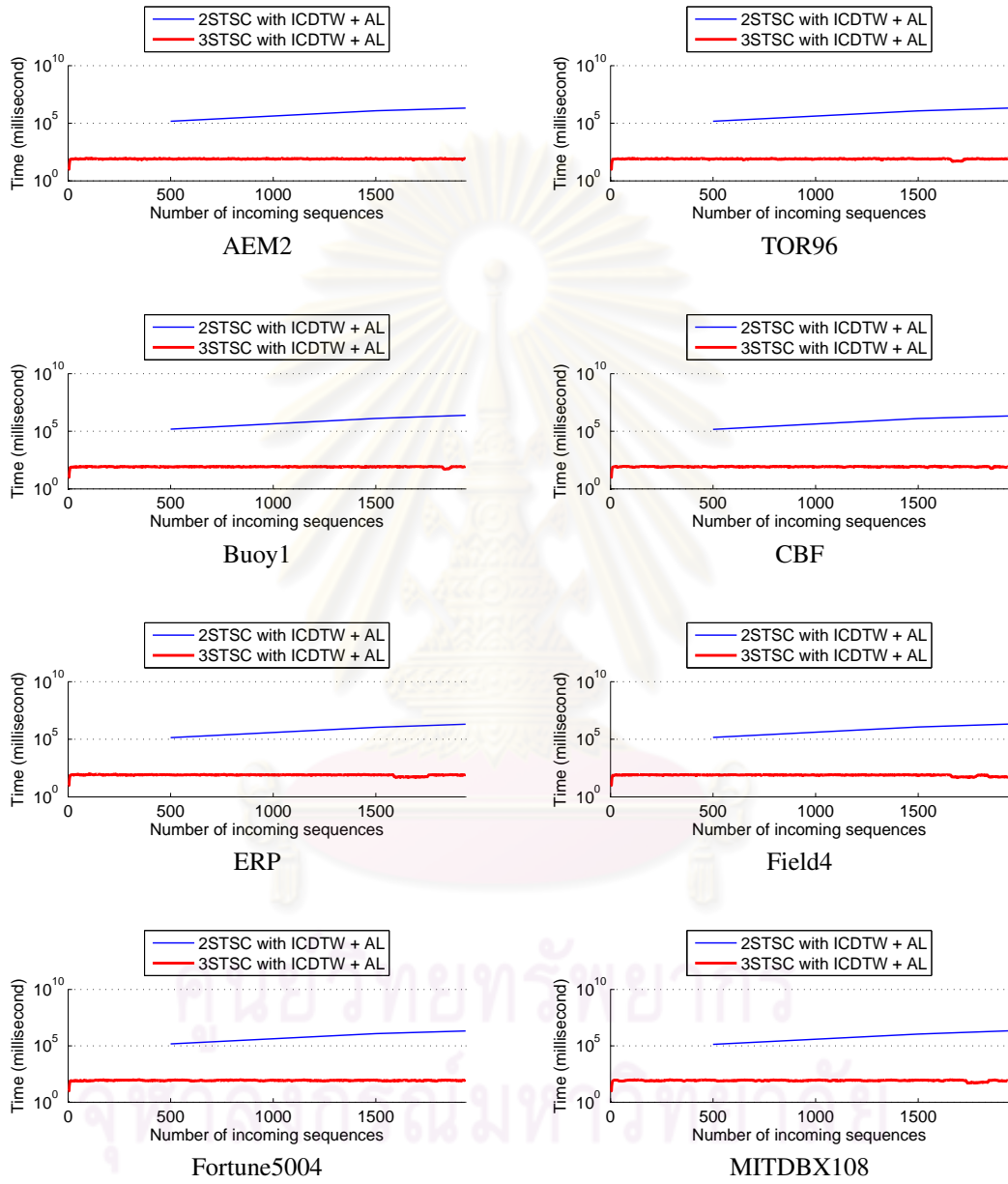


Figure G.19: Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where  $k = 7$  and  $w = 64$ .

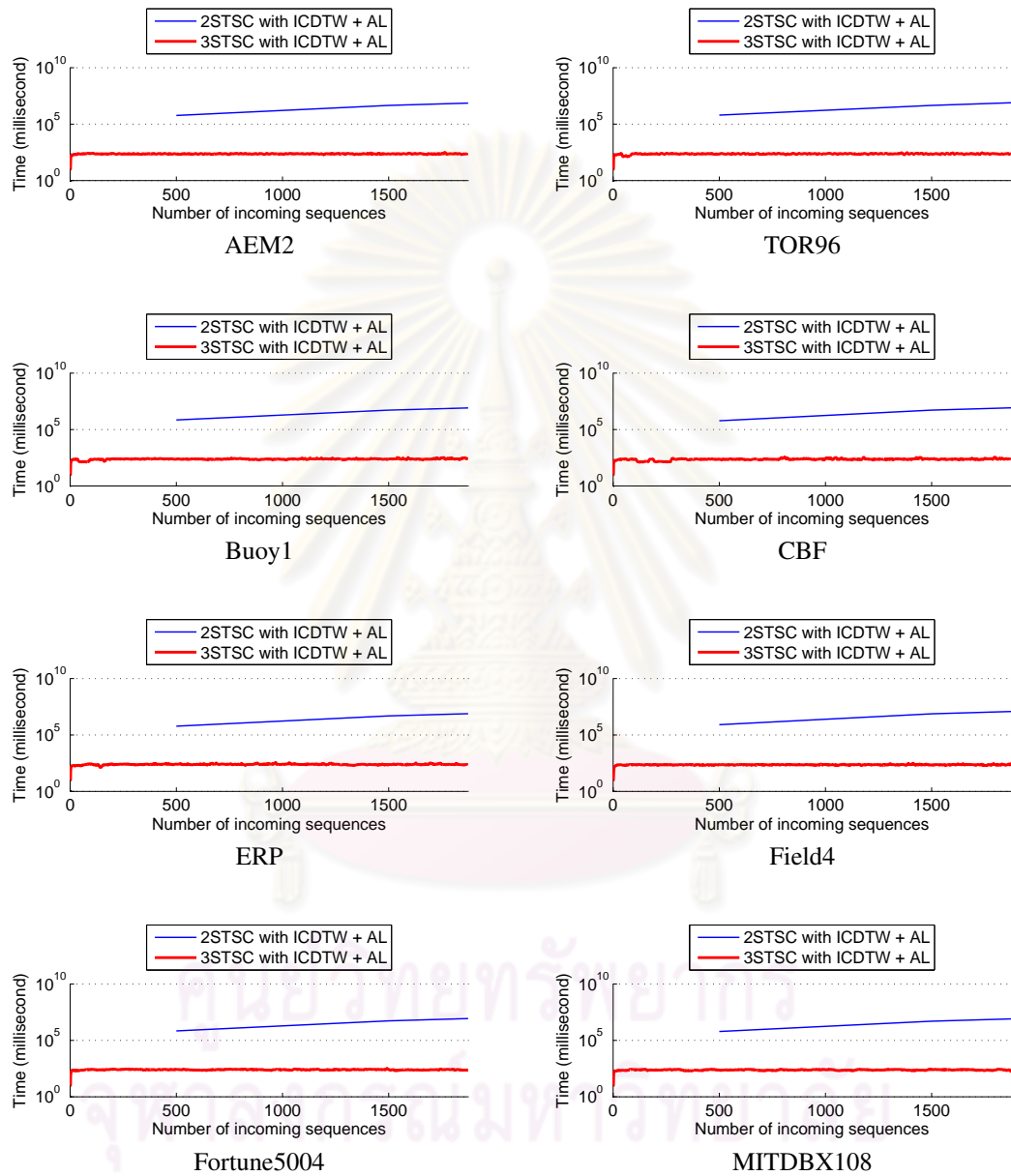


Figure G.20: Computational time of 3STSC and 2STSC with CDTW function and average linkage when a new incoming sequence arrives, where  $k = 3$  and  $w = 128$ .

## APPENDIX H

### COMPLETE EXPERIMENTAL RESULTS OF THE SECOND EXPERIMENT IN CHAPTER 6



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย

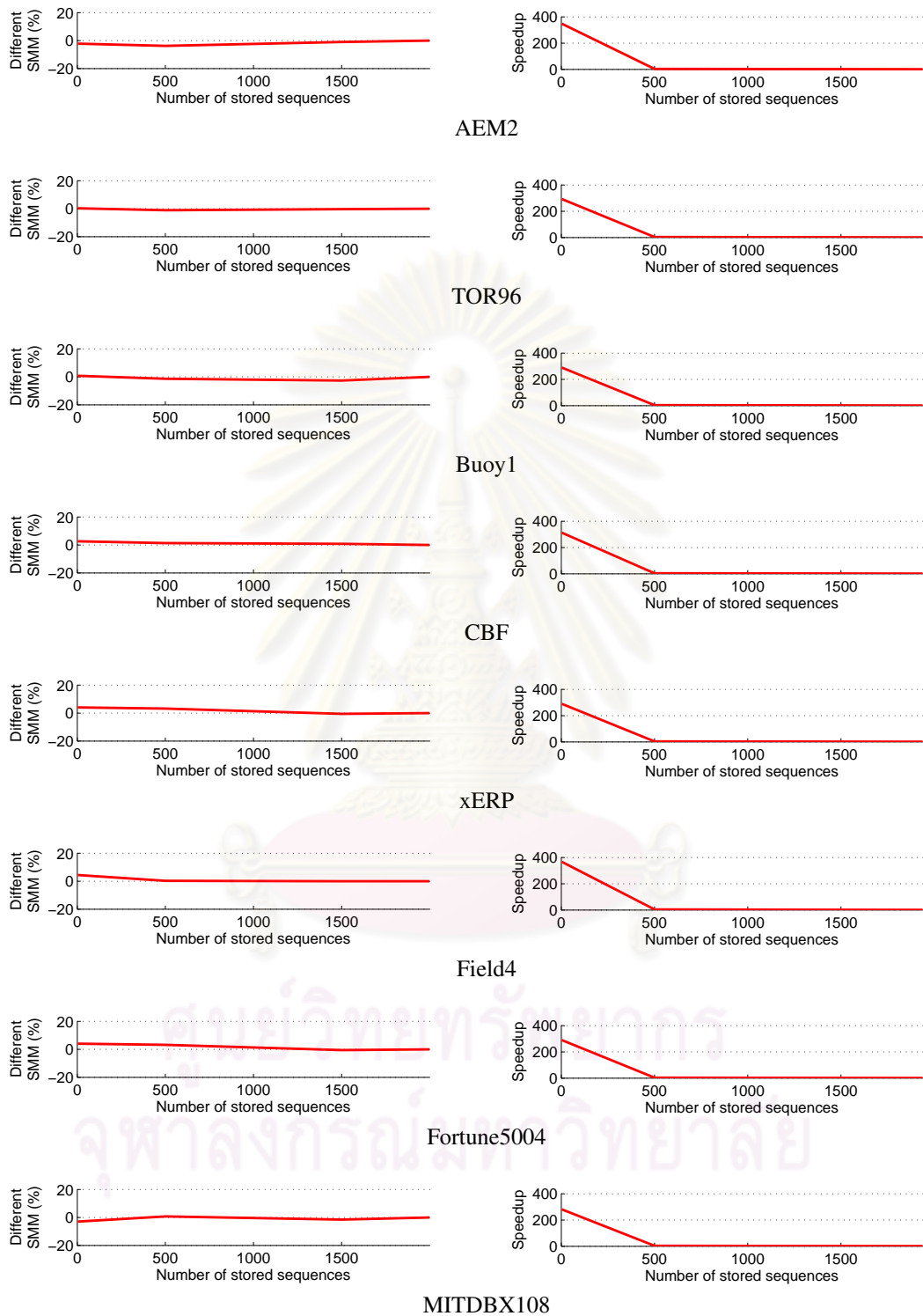


Figure H.1: Percentage difference of SMM and speedup of 3TSC with CDTW function and complete linkage when  $k = 3$ ,  $w = 64$ , and number of stored sequences are varied.

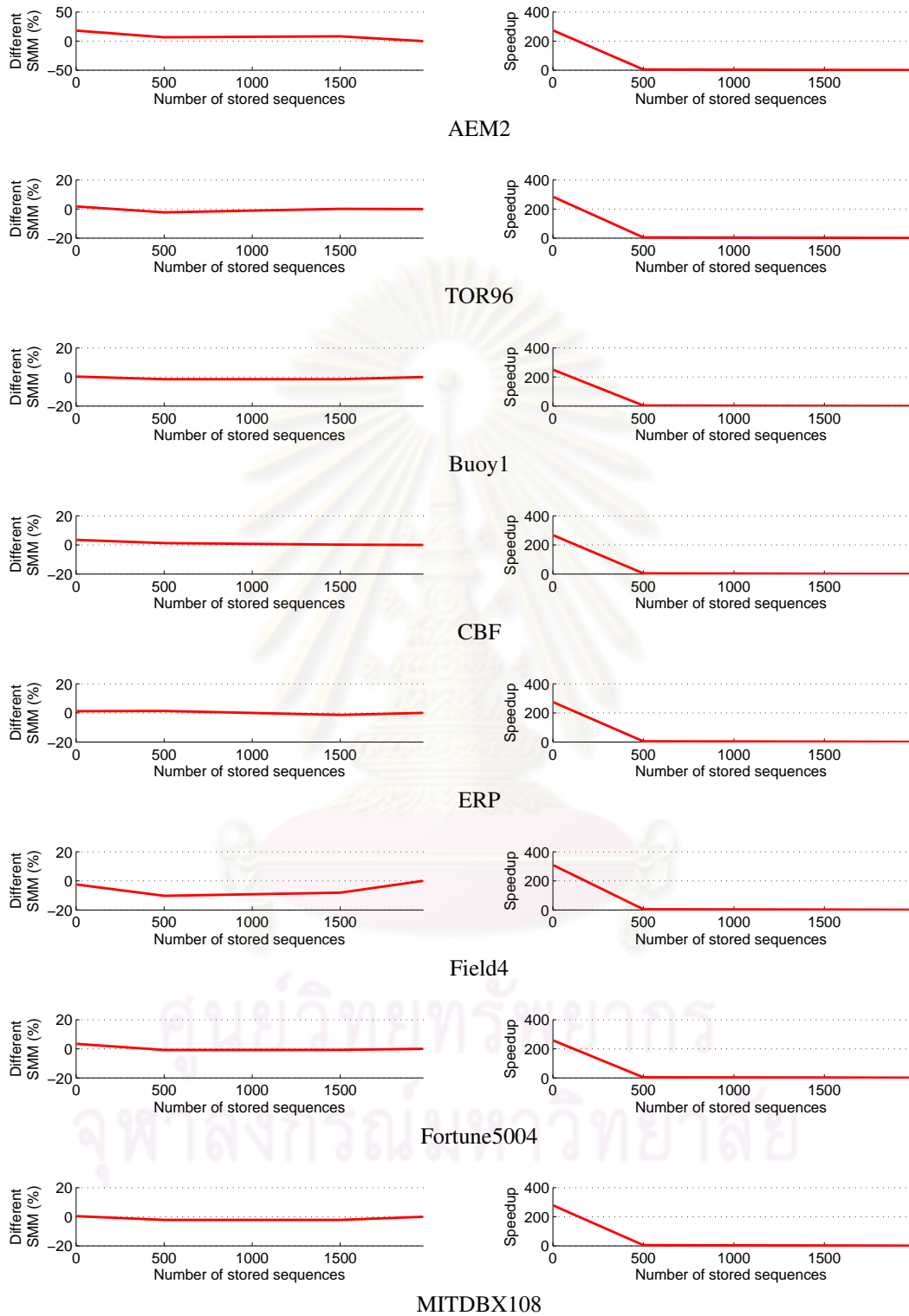


Figure H.2: Percentage difference of SMM and speedup of 3STSC with CDTW function and complete linkage when  $k = 3$ ,  $w = 32$ , and number of stored sequences are varied.



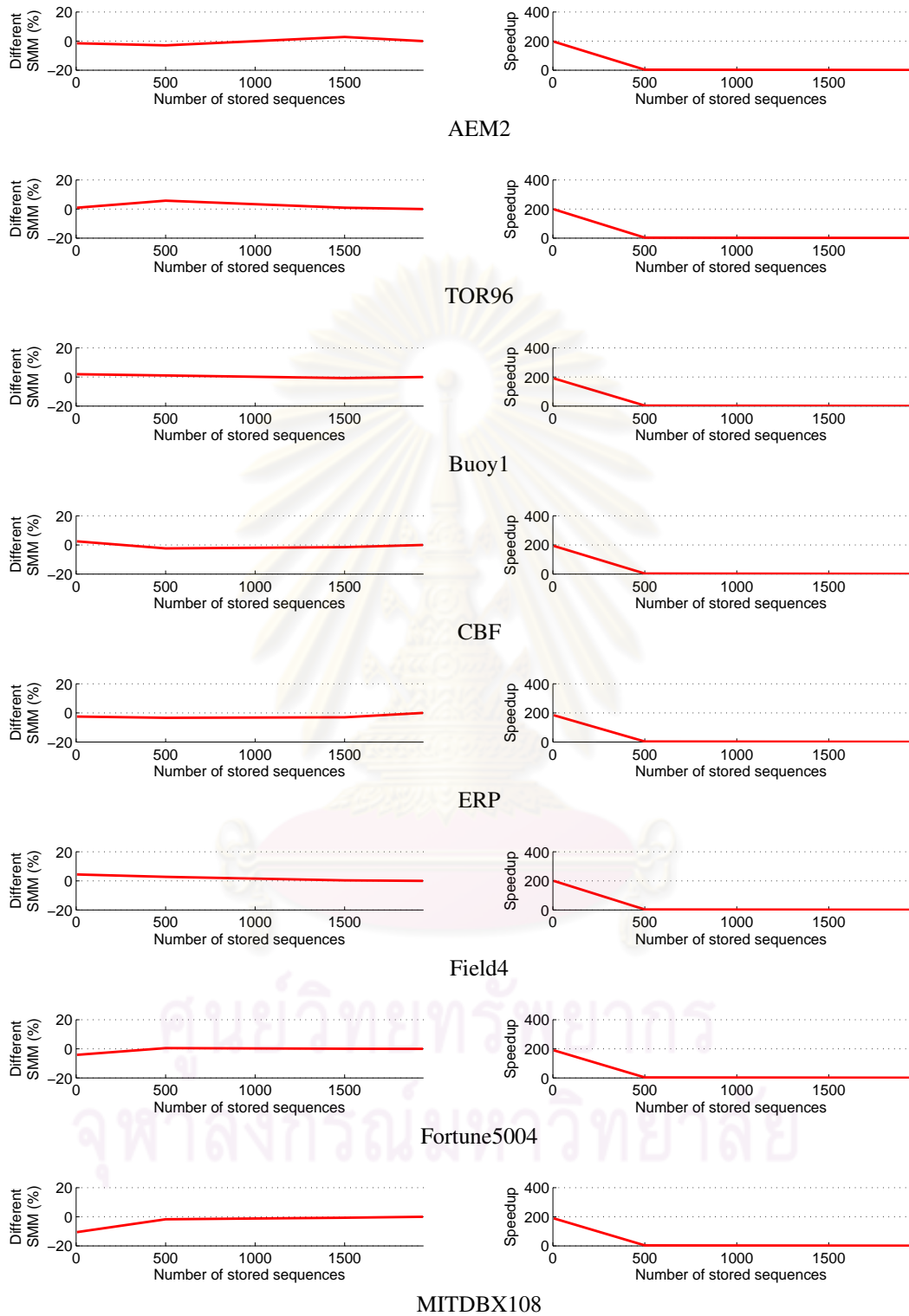


Figure H.3: Percentage difference of SMM and speedup of 3TSC with CDTW function and complete linkage when  $k = 5$ ,  $w = 64$ , and number of stored sequences are varied.

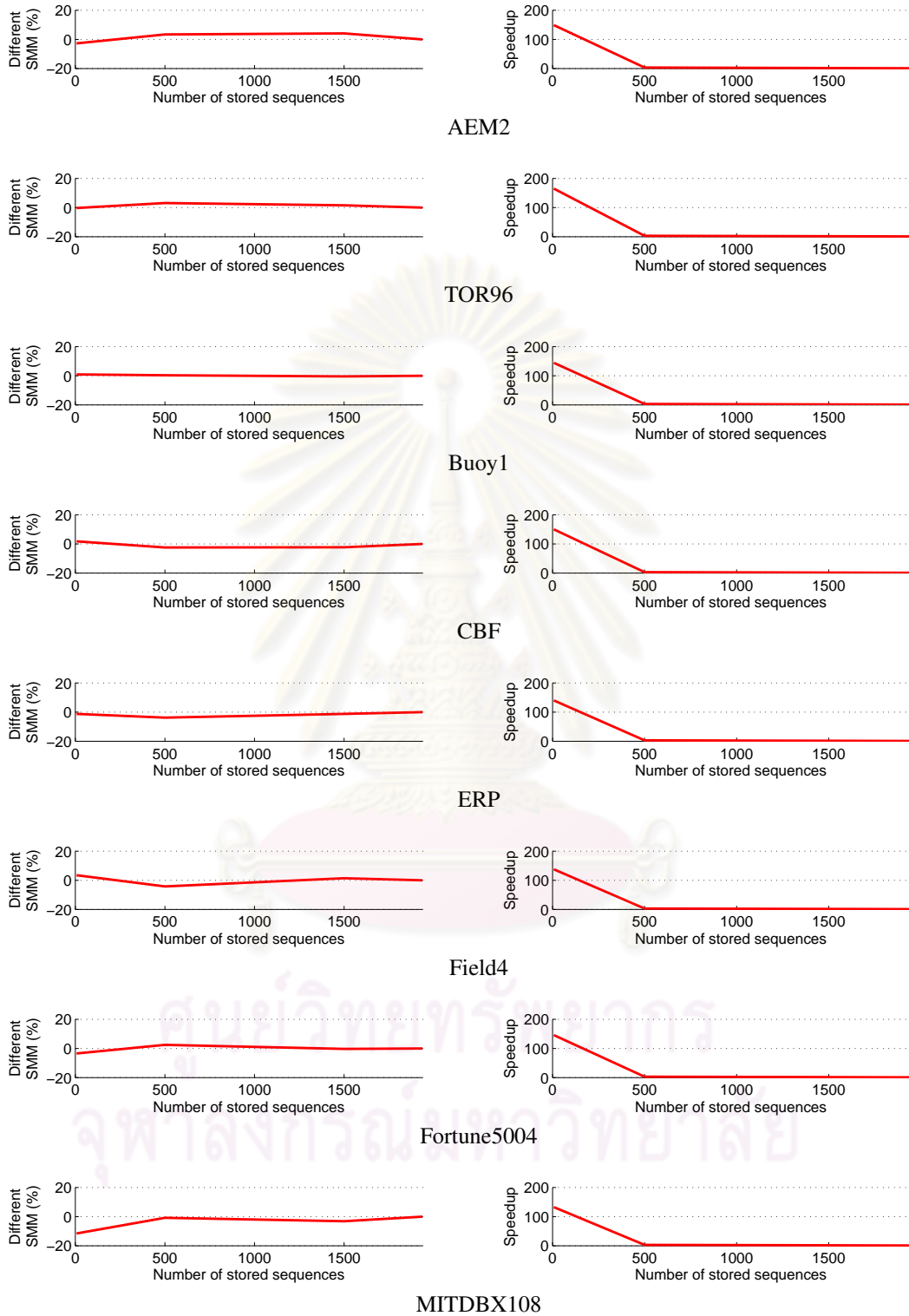


Figure H.4: Percentage difference of SMM and speedup of 3STSC with CDTW function and complete linkage when  $k = 7$ ,  $w = 64$ , and number of stored sequences are varied.

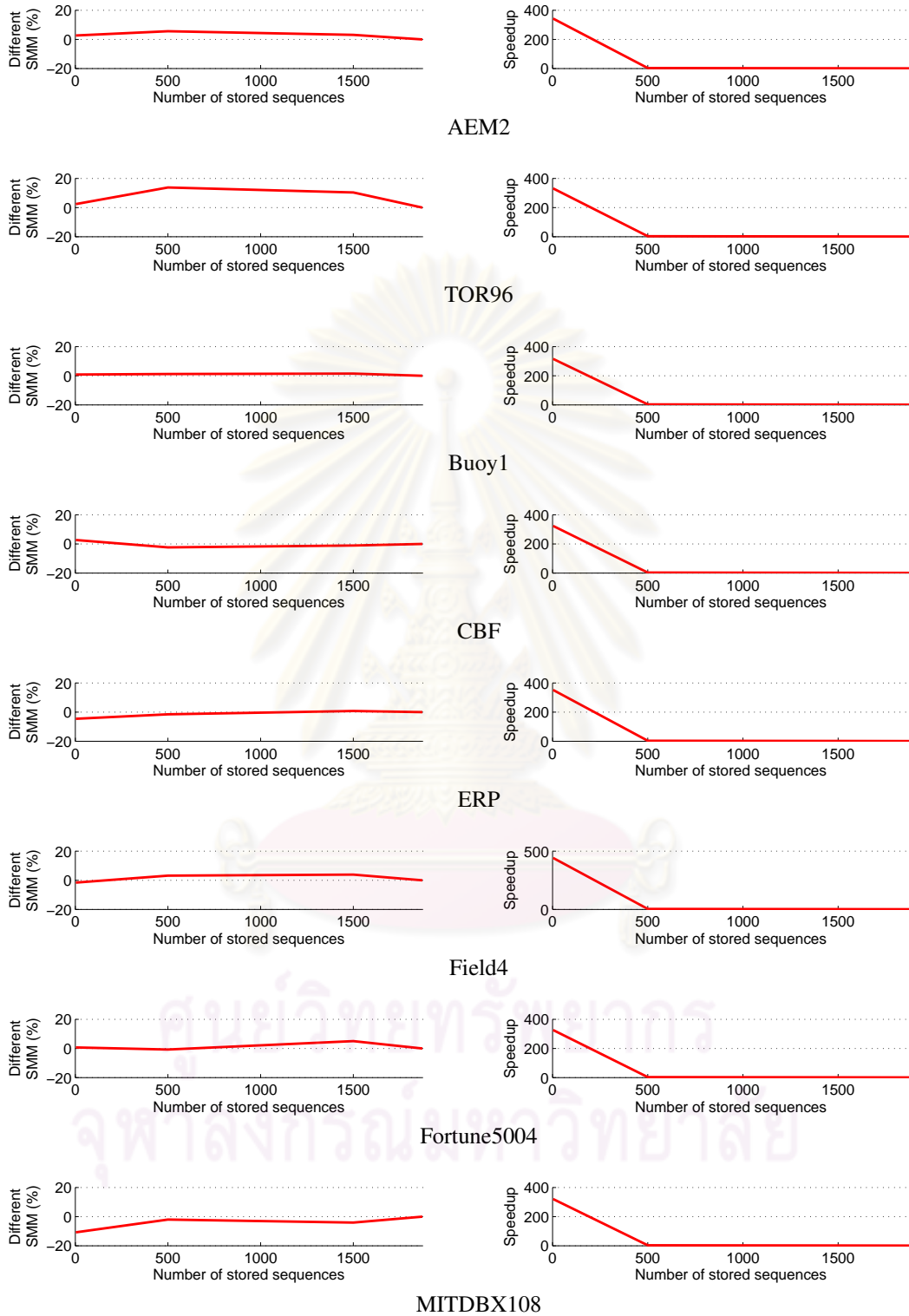


Figure H.5: Percentage difference of SMM and speedup of 3STSC with CDTW function and complete linkage when  $k = 3$ ,  $w = 128$ , and number of stored sequences are varied.

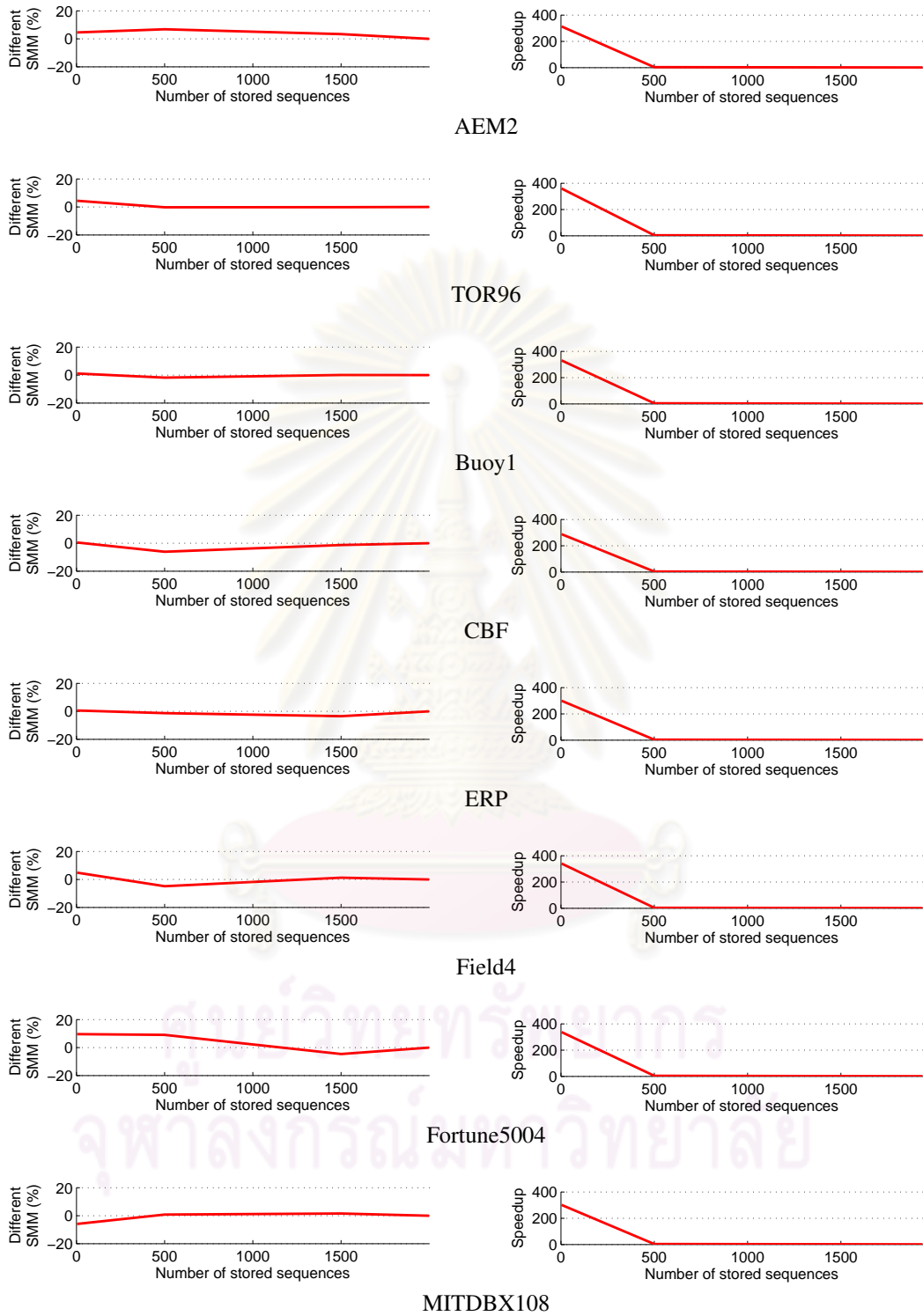


Figure H.6: Percentage difference of SMM and speedup of 3STSC with CDTW function and average linkage when  $k = 3$ ,  $w = 64$ , and number of stored sequences are varied.

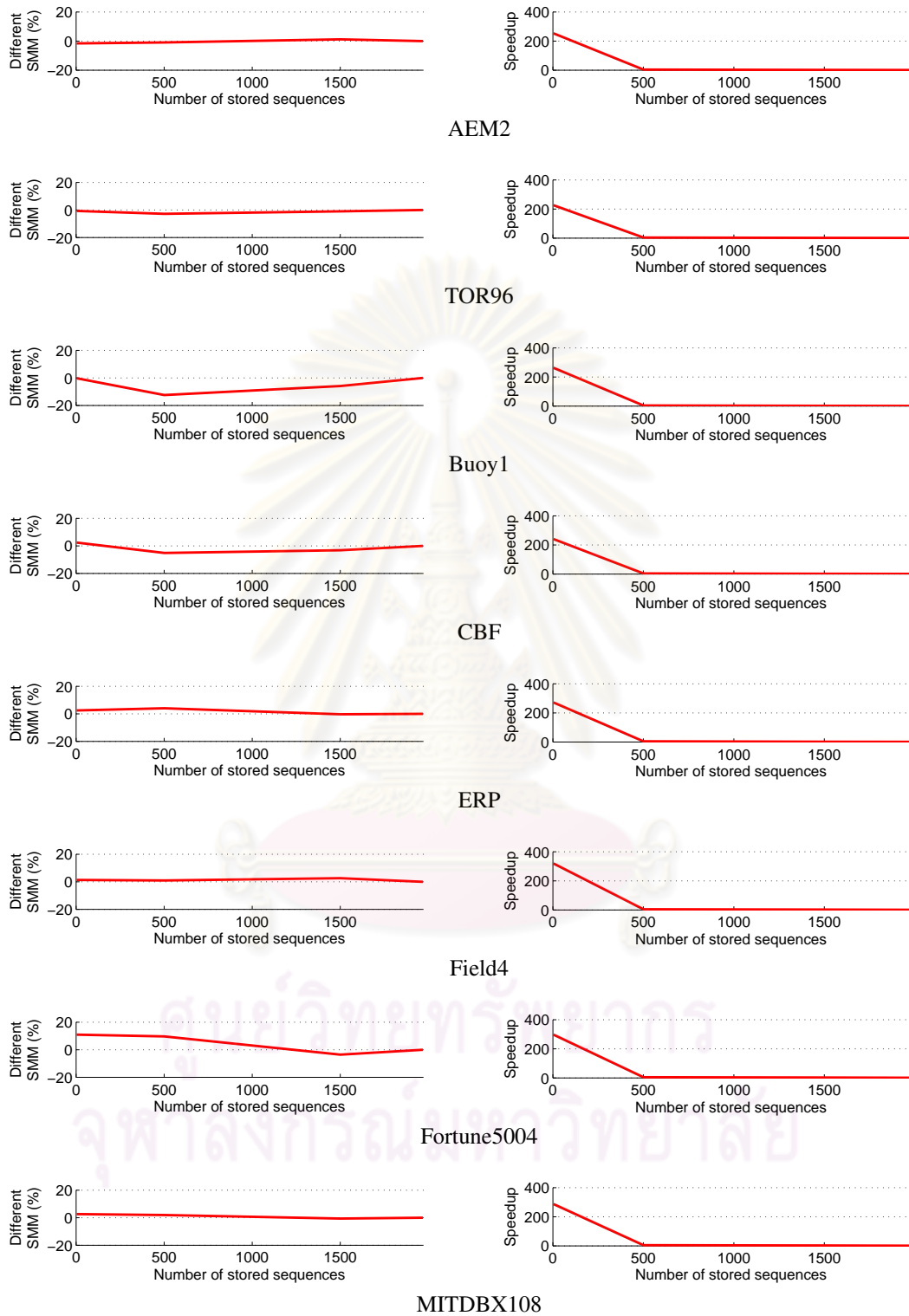


Figure H.7: Percentage difference of SMM and speedup of 3TSC with CDTW function and average linkage when  $k = 3$ ,  $w = 64$ , and number of stored sequences are varied.

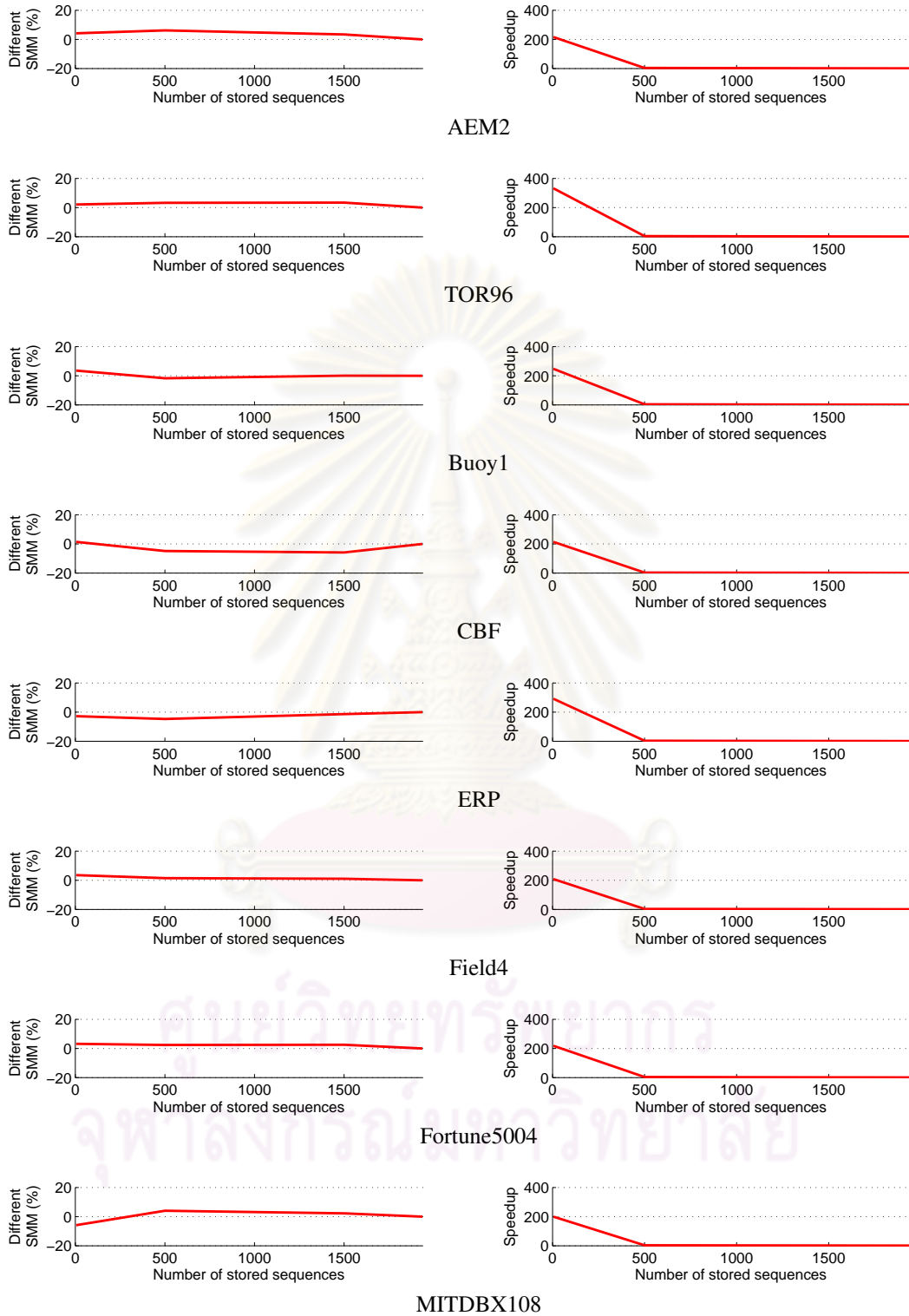


Figure H.8: Percentage difference of SMM and speedup of 3TSC with CDTW function and average linkage when  $k = 5$ ,  $w = 64$ , and number of stored sequences are varied.

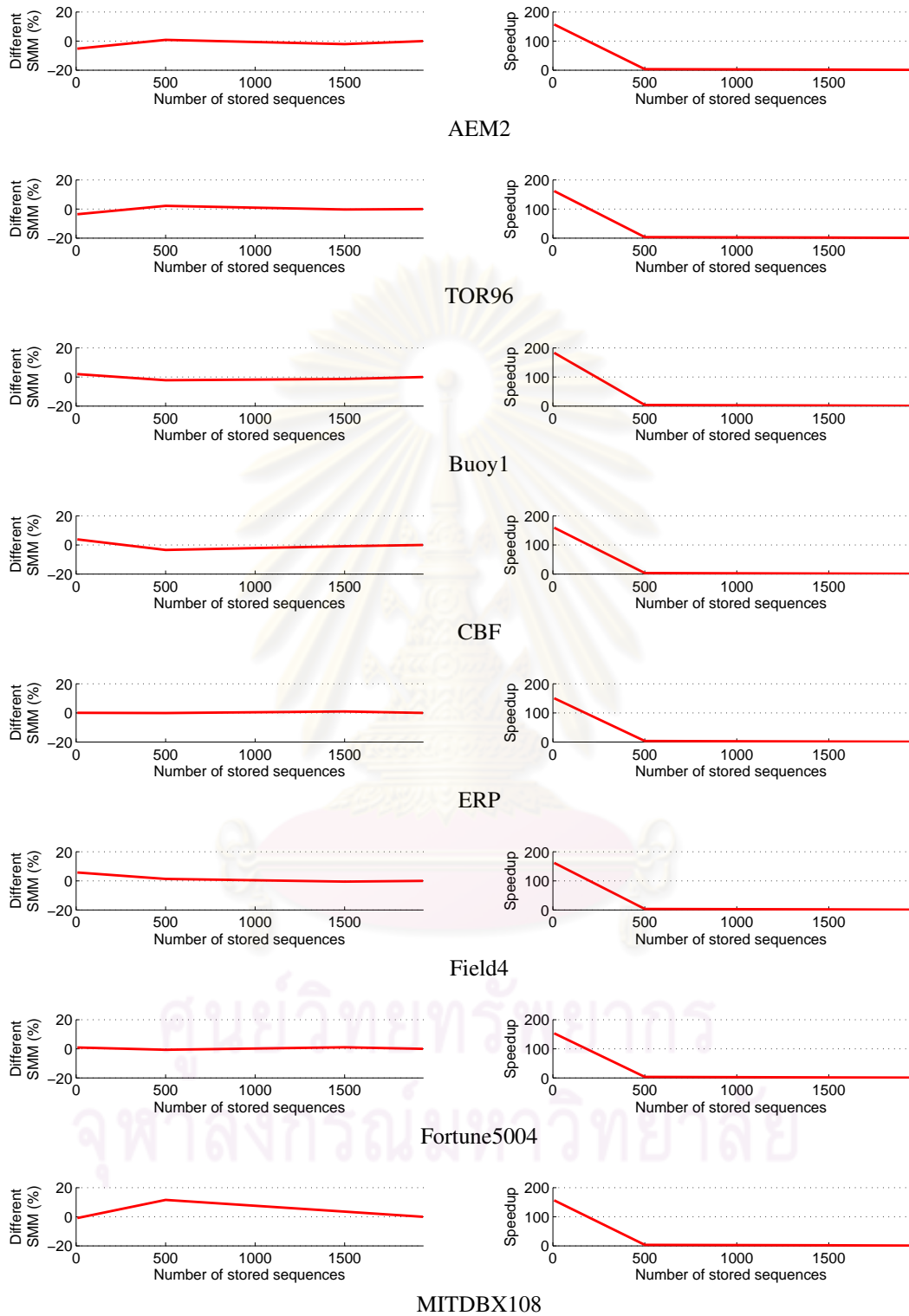


Figure H.9: Percentage difference of SMM and speedup of 3TSC with CDTW function and average linkage when  $k = 7$ ,  $w = 64$ , and number of stored sequences are varied.

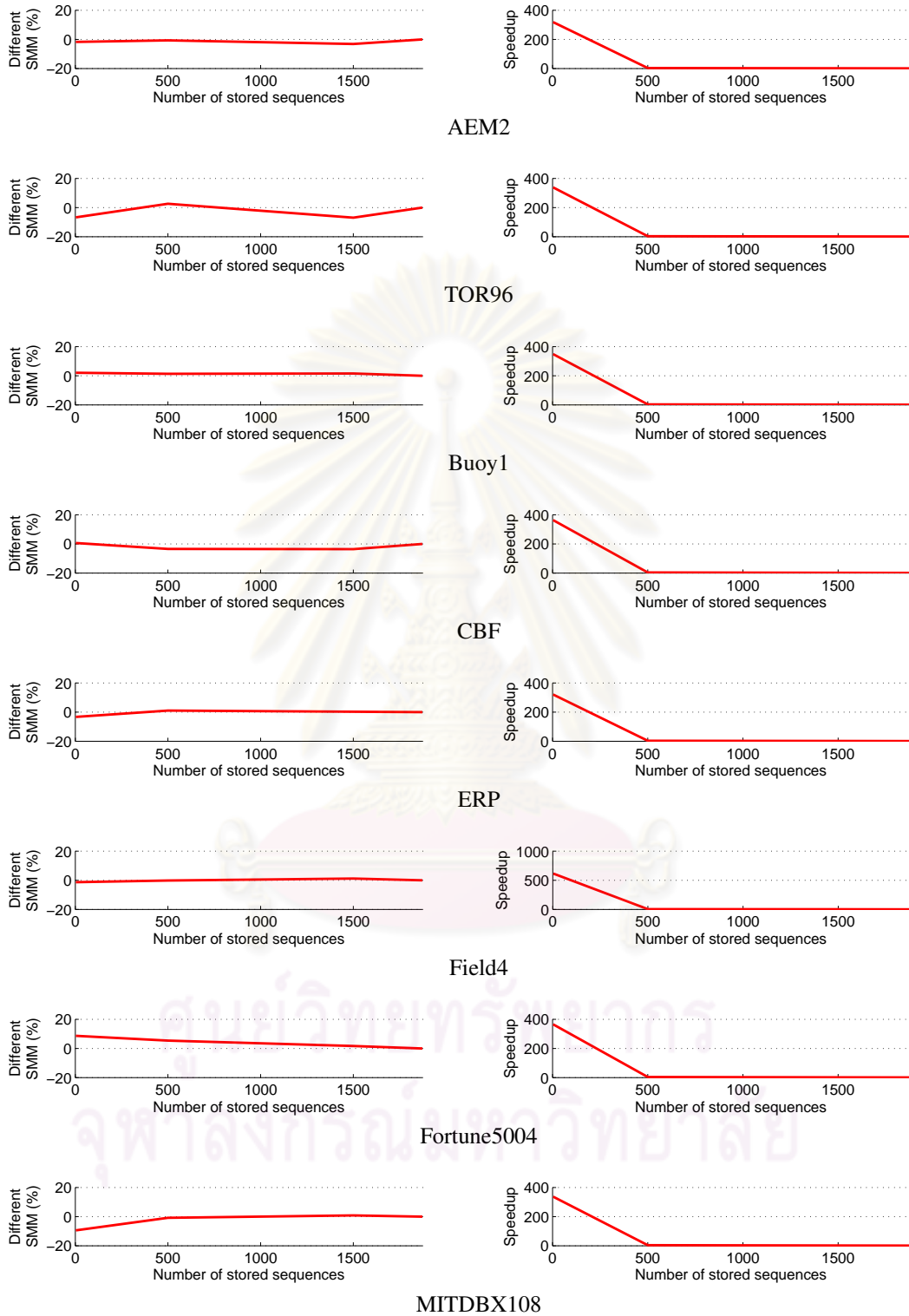


Figure H.10: Percentage difference of SMM and speedup of 3STSC with CDTW function and average linkage when  $k = 3$ ,  $w = 128$ , and number of stored sequences are varied.



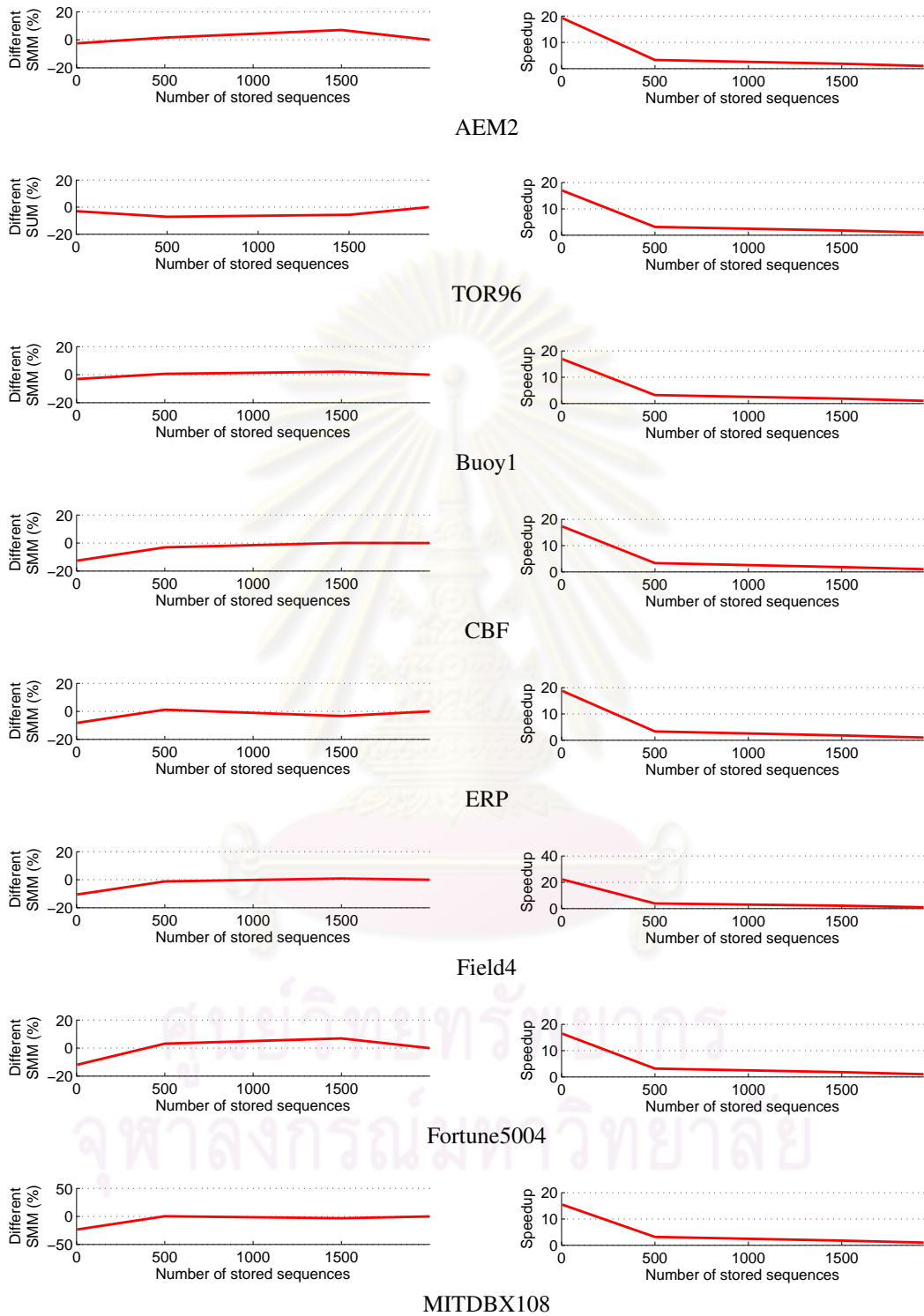


Figure H.11: Percentage difference of SMM and speedup of 3TSC with ICDTW function and complete linkage when  $k = 3$ ,  $w = 64$ , and number of stored sequences are varied.

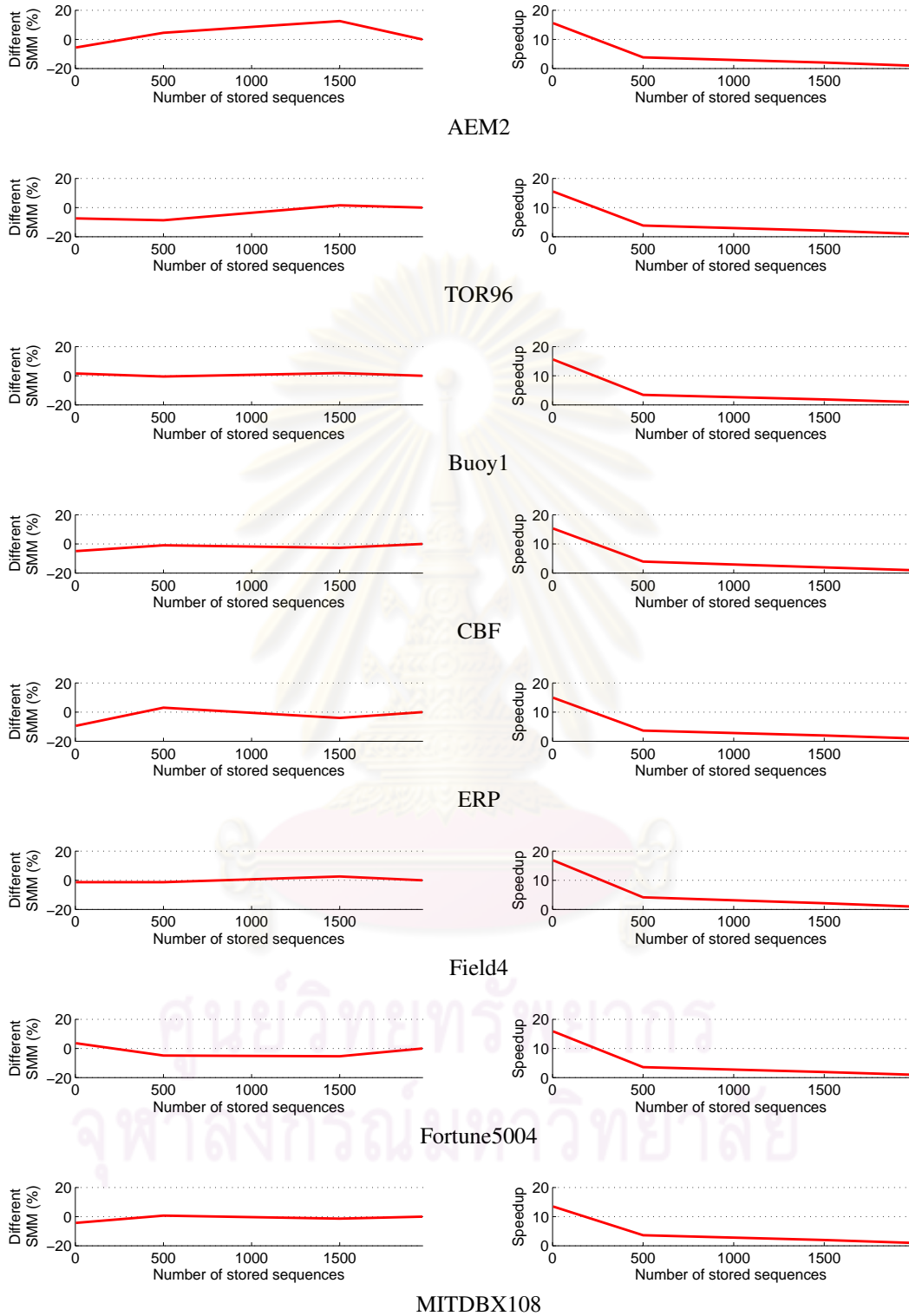


Figure H.12: Percentage difference of SMM and speedup of 3STSC with ICDTW function and complete linkage when  $k = 3$ ,  $w = 32$ , and number of stored sequences are varied.

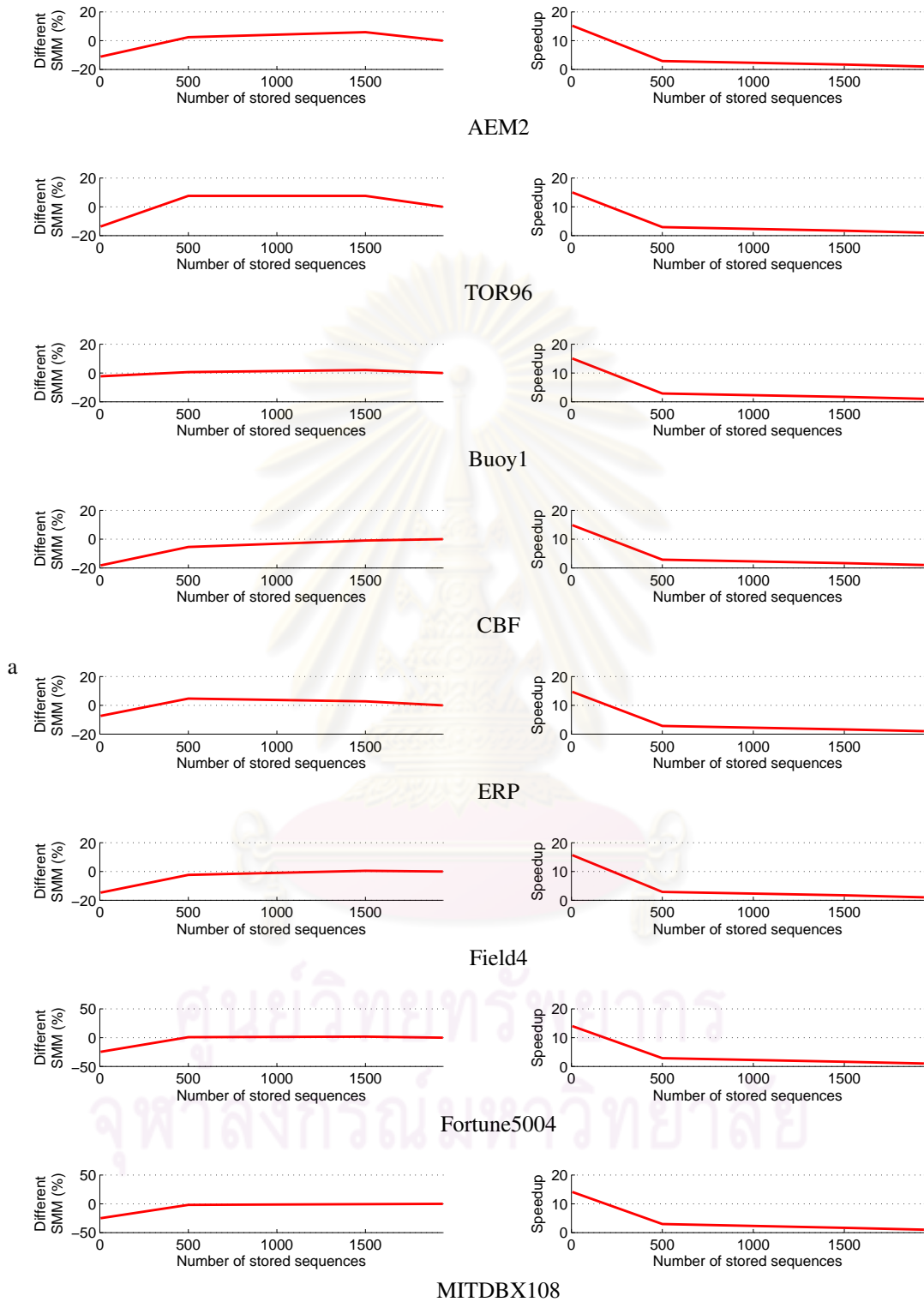


Figure H.13: Percentage difference of SMM and speedup of 3TSC with ICDTW function and complete linkage when  $k = 5$ ,  $w = 64$ , and number of stored sequences are varied.

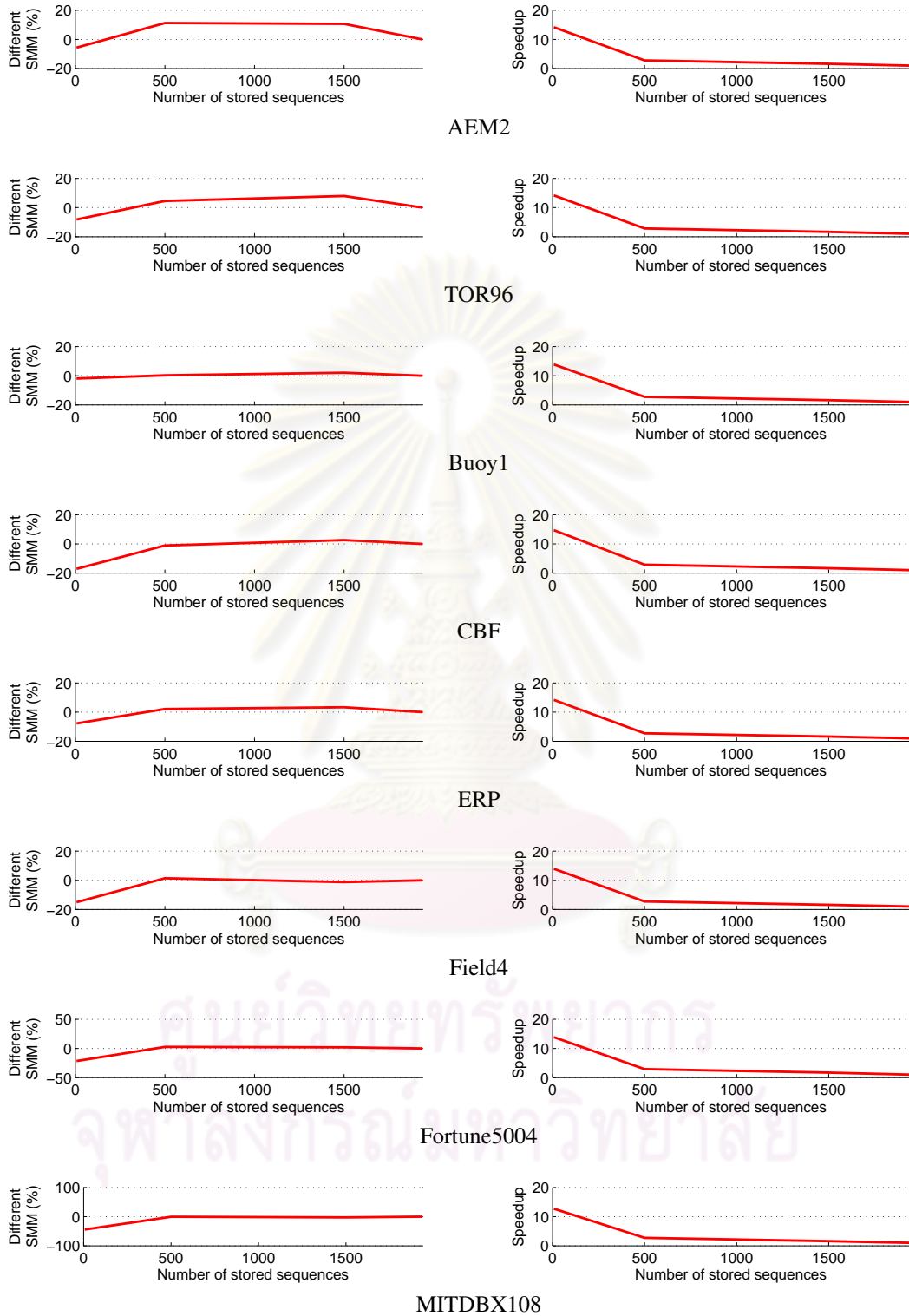


Figure H.14: Percentage difference of SMM and speedup of 3TSC with ICDTW function and complete linkage when  $k = 7$ ,  $w = 64$ , and number of stored sequences are varied.

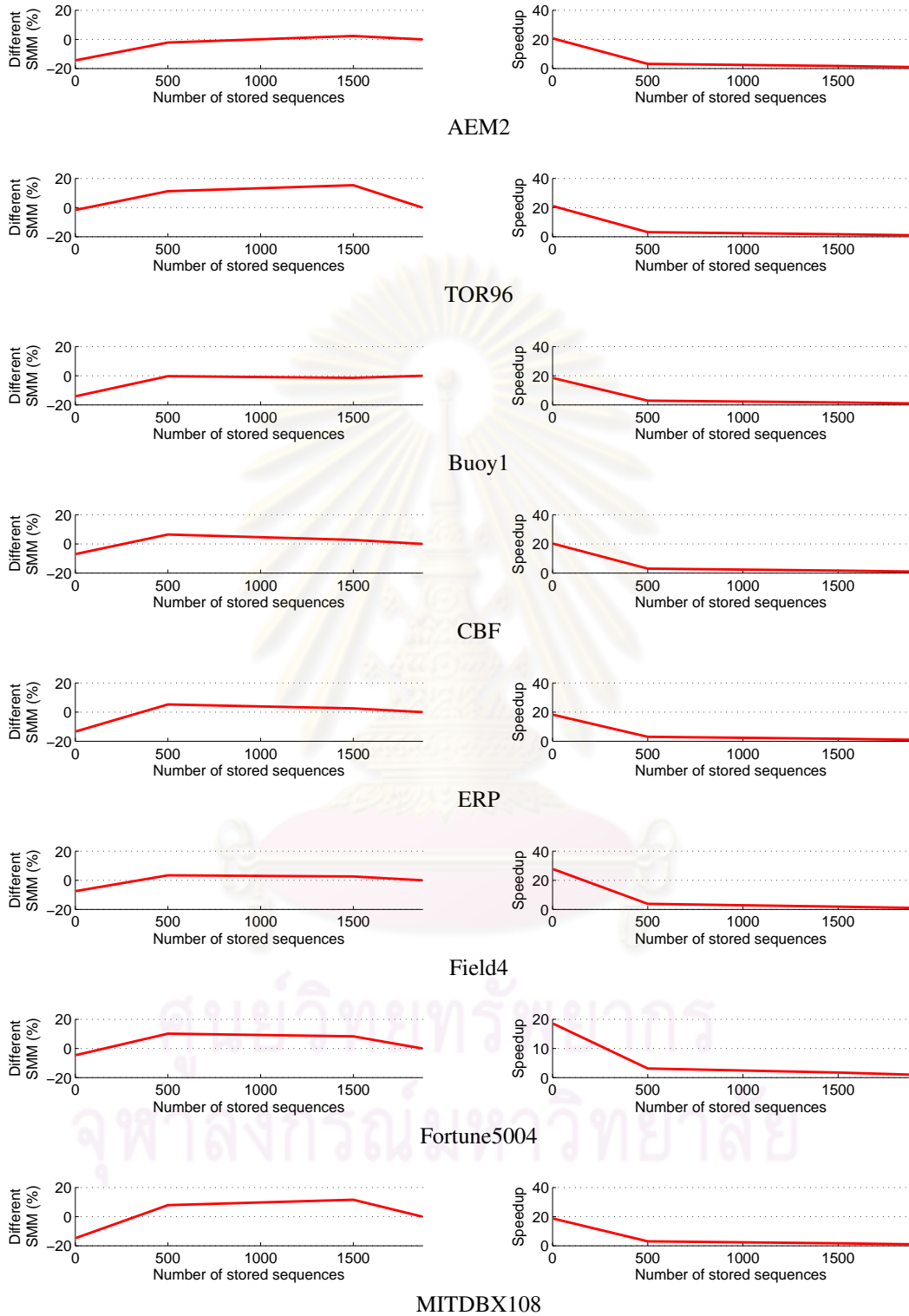


Figure H.15: Percentage difference of SMM and speedup of 3STSC with ICDTW function and complete linkage when  $k = 3$ ,  $w = 128$ , and number of stored sequences are varied.

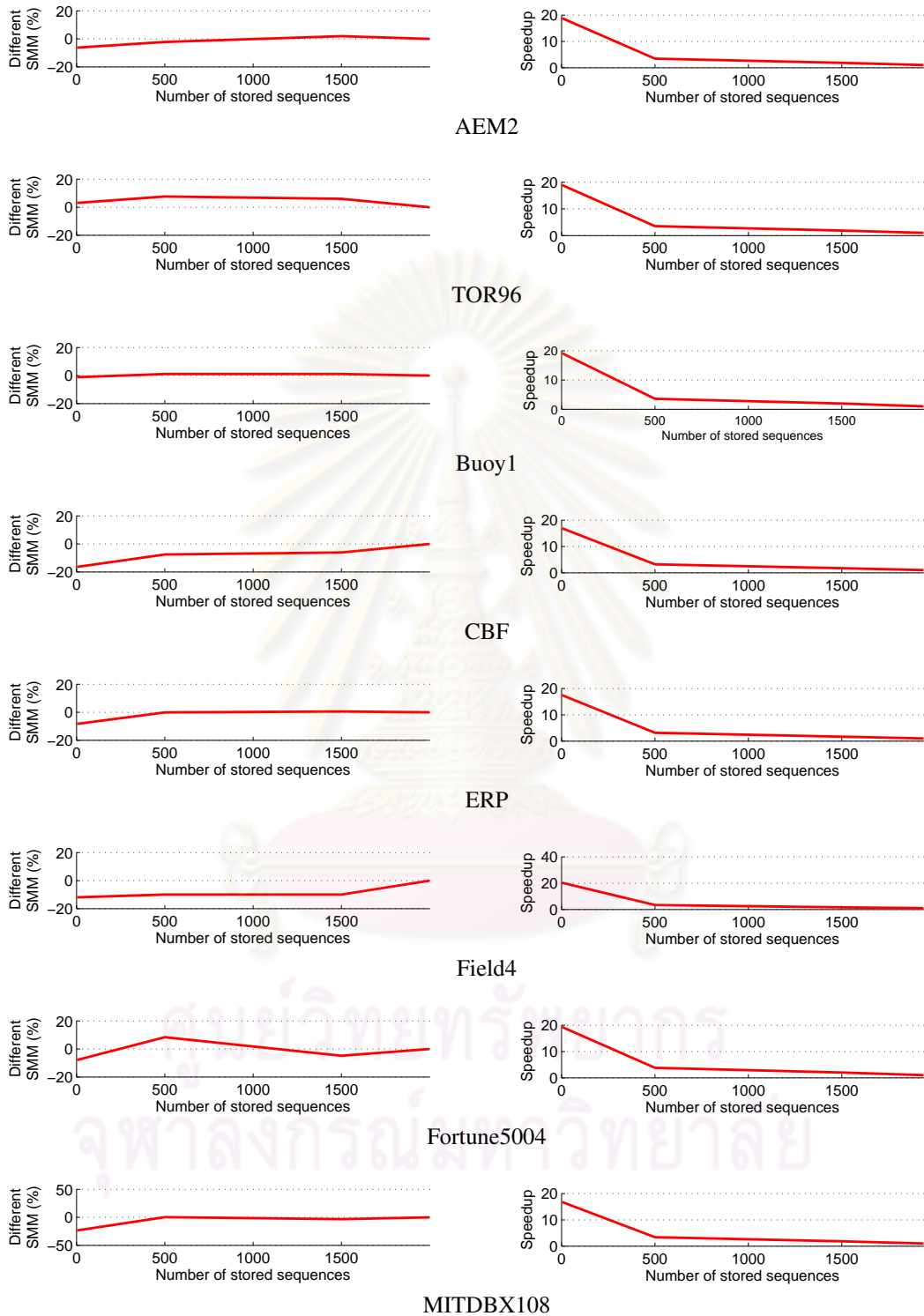


Figure H.16: Percentage difference of SMM and speedup of 3STSC with ICDTW function and average linkage when  $k = 3$ ,  $w = 64$ , and number of stored sequences are varied.

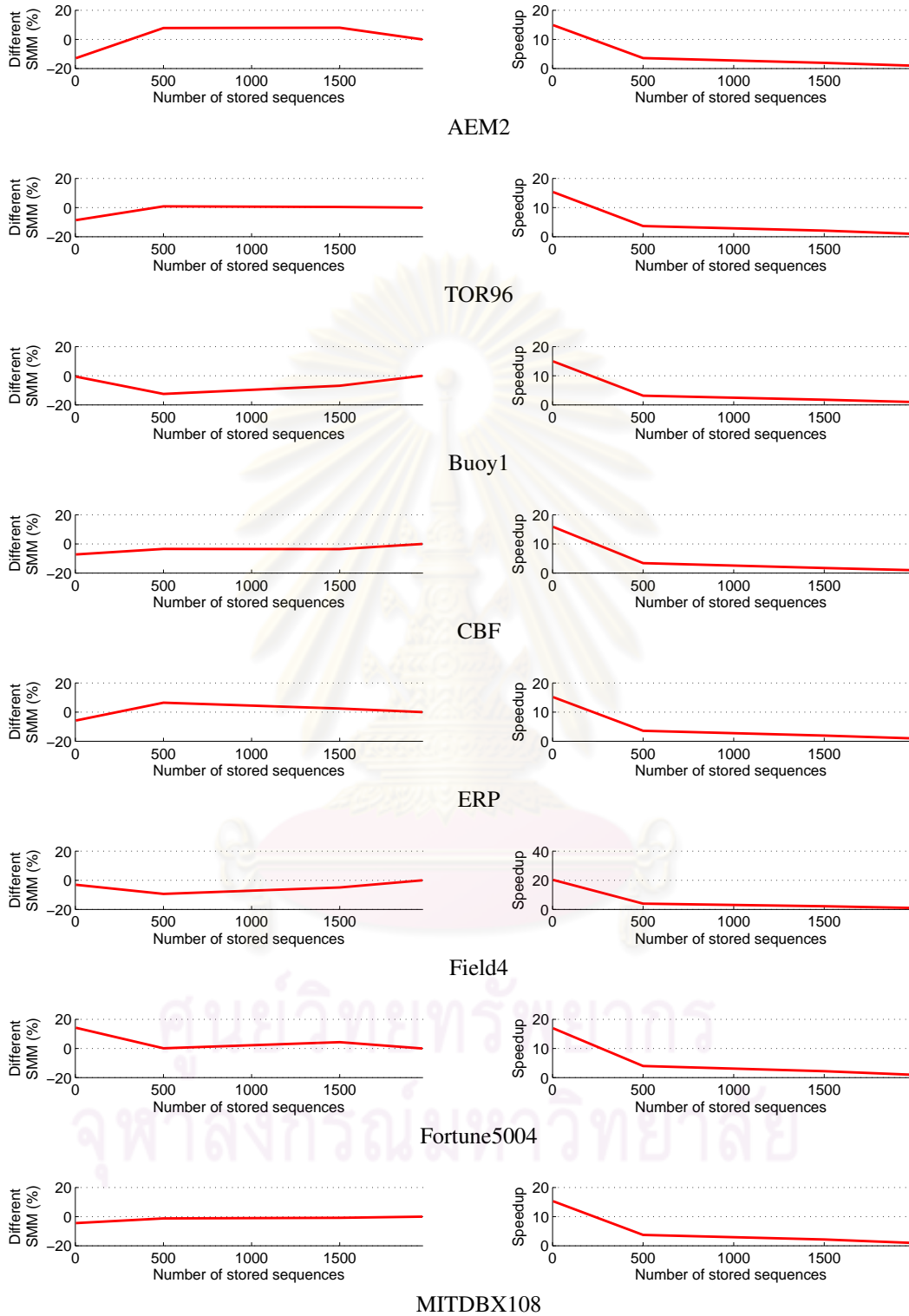


Figure H.17: Percentage difference of SMM and speedup of 3STSC with ICDTW function and average linkage when  $k = 3$ ,  $w = 32$ , and number of stored sequences are varied.

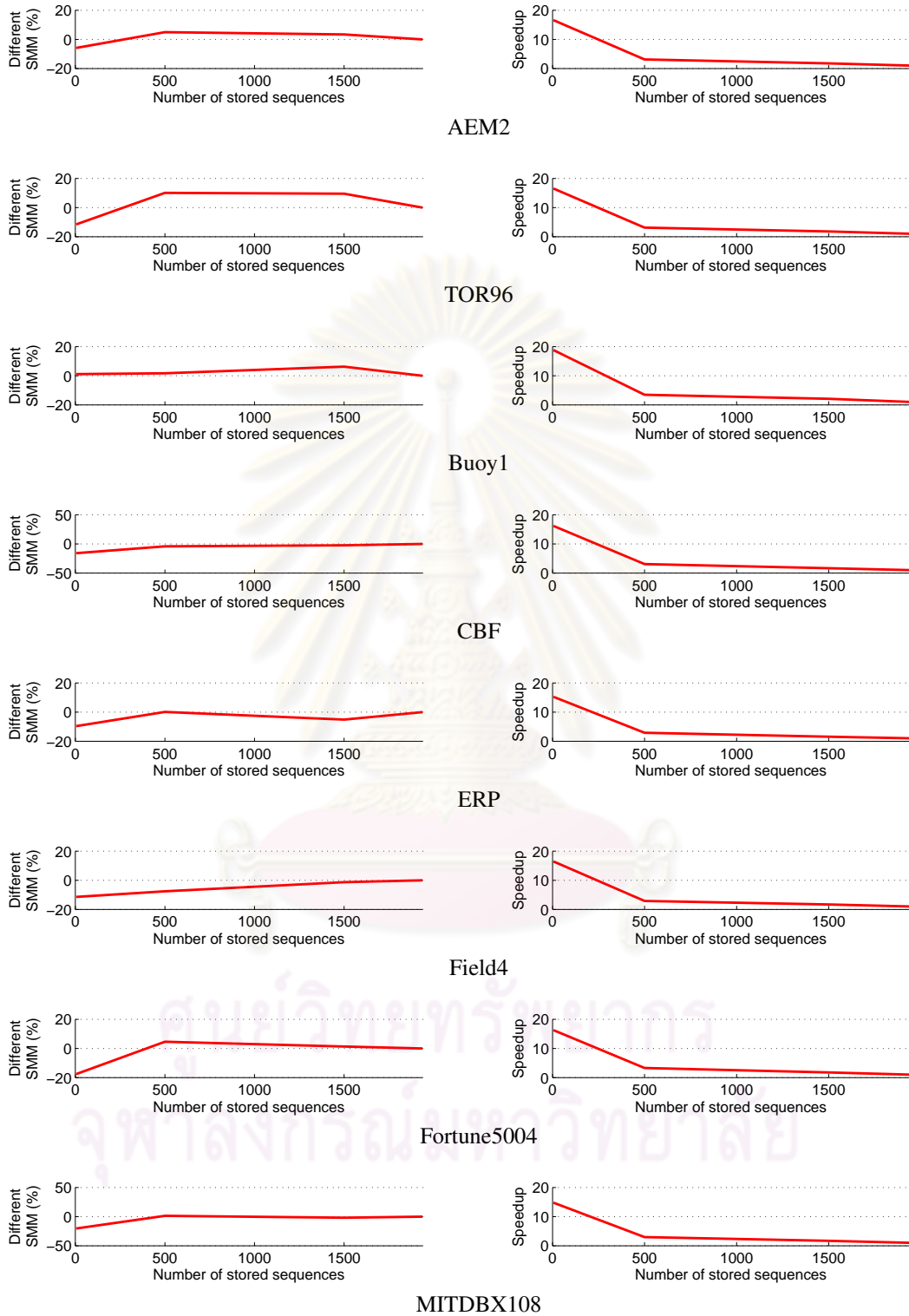


Figure H.18: Percentage difference of SMM and speedup of 3STSC with ICDTW function and average linkage when  $k = 3$ ,  $w = 64$ , and number of stored sequences are varied.



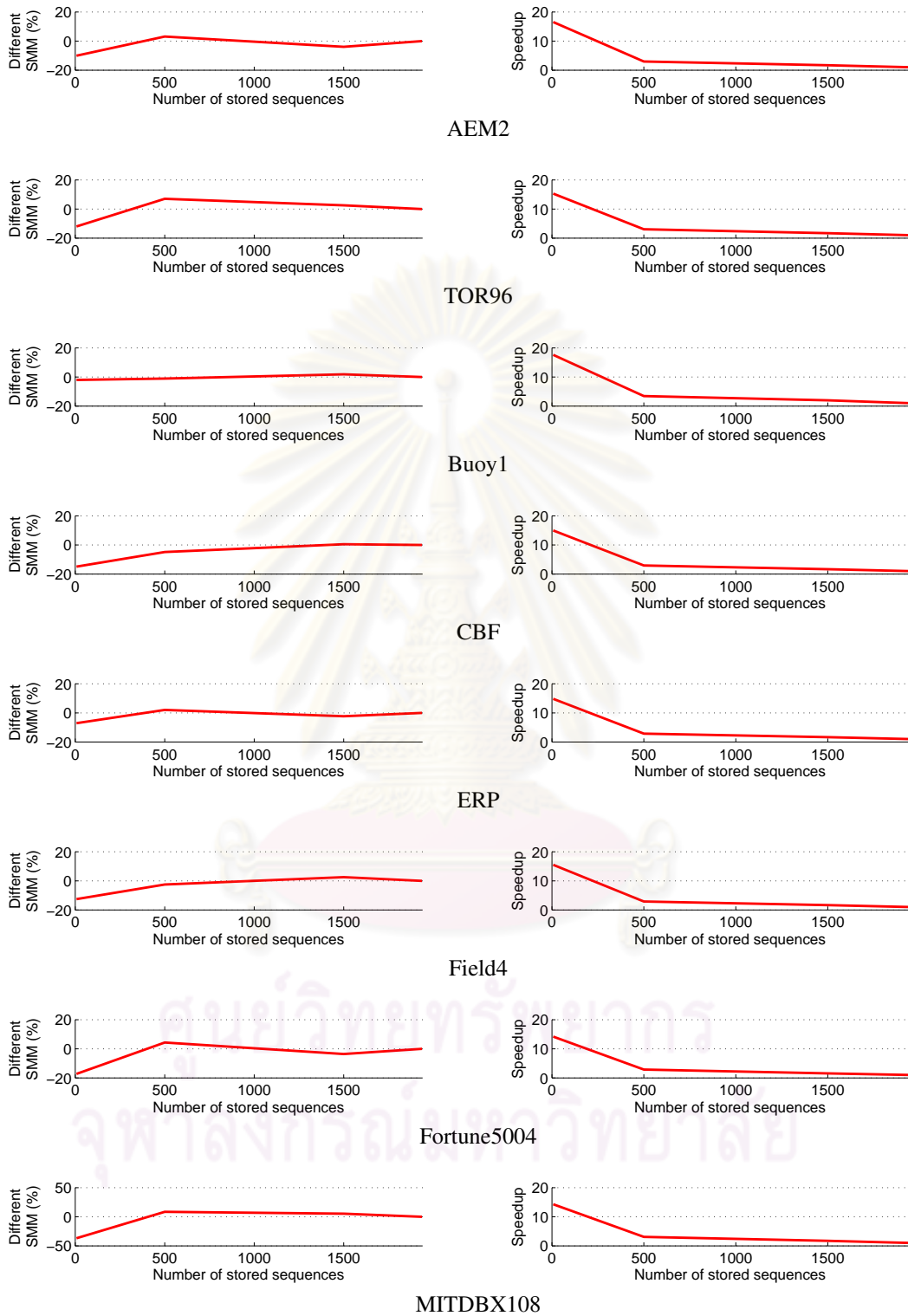


Figure H.19: Percentage difference of SMM and speedup of 3STSC with ICDTW function and average linkage when  $k = 7$ ,  $w = 64$ , and number of stored sequences are varied.

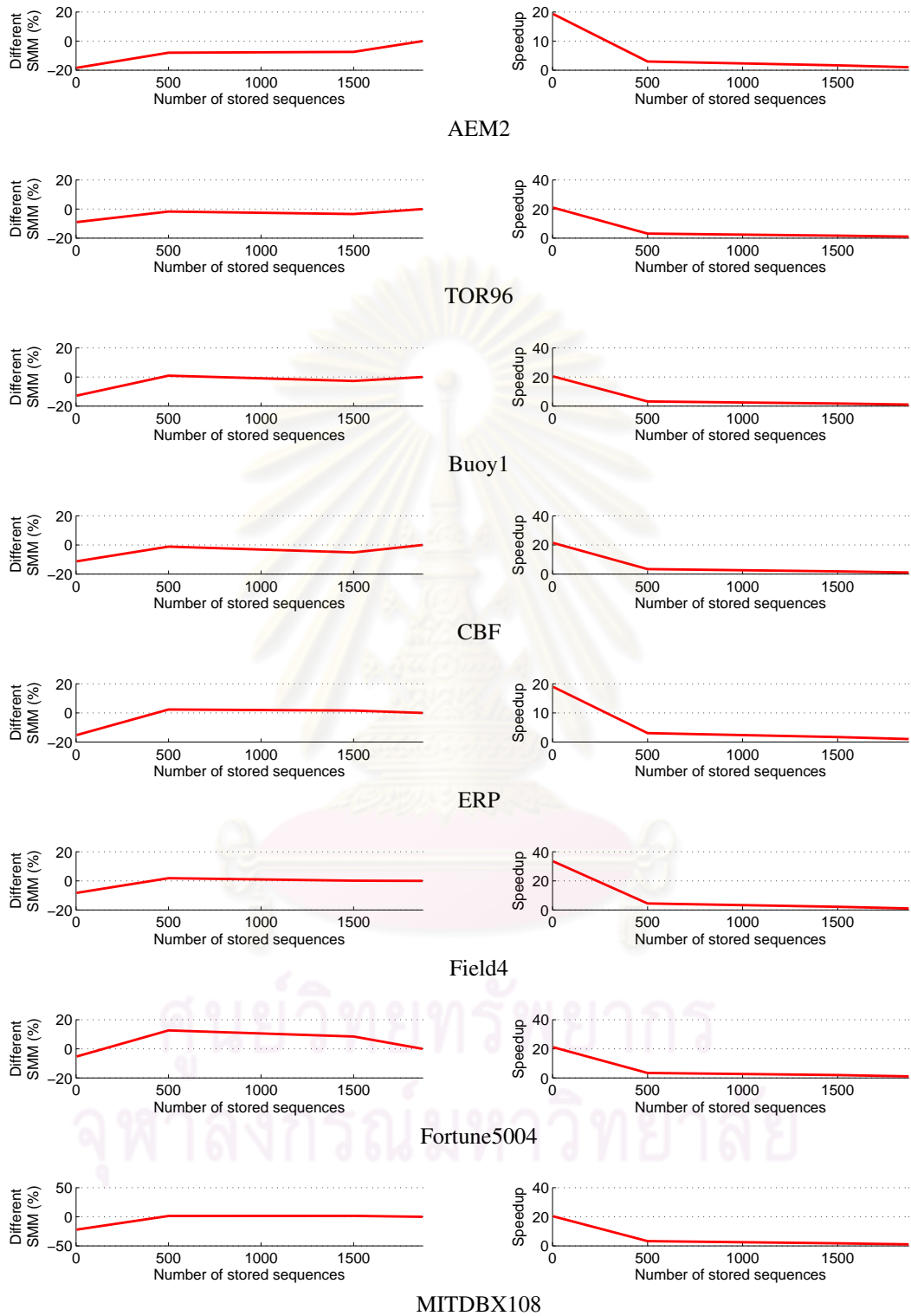


Figure H.20: Percentage difference of SMM and speedup of 3TSC with ICDTW function and average linkage when  $k = 3$ ,  $w = 128$ , and number of stored sequences are varied.

## Biography

Vit Niennattrakul was born in Bangkok, Thailand, on September 8, 1984. He received his B.Eng. in Computer Engineering from Chulalongkorn University in 2006. His doctorate has been under supervision of Asst. Prof. Dr. Chotirat Ann Ratanamahatana. During his Ph.D. study, he was a junior specialist at the University of California, Riverside under supervision of Prof. Eamonn J. Keogh for one year (September 2009 to August 2010), and he was granted scholarships from the Thailand Research Fund through the Royal Golden Jubilee Ph.D. Program (June 2007 to May 2011), Chulalongkorn University Graduate Scholarship to Commemorate the 72<sup>nd</sup> Anniversary of His Majesty King Bhumibol Adulyadej (June 2006 to May 2007), and the 90<sup>th</sup> Anniversary of Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund) (September 2009). He also was reviewers of many well-known journals including Knowledge and Information System (KAIS) and Data Mining and Knowledge Discovery (DMKD). His research interests include but not limited to time series data mining, machine learning, and natural language processing.



ศูนย์วิทยทรัพยากร  
จุฬาลงกรณ์มหาวิทยาลัย