การค้นพบเอพิโสดที่เกิดบ่อยในข้อมูลอนุกรมเวลา

นายสุระ รอดพงษ์พันธ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2558

DISCOVERING FREQUENT EPISODES IN TIME SERIES

Mr. Sura Rodpongpun

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 2015

Thesis Title	DISCOVERING FREQUENT EPISODES IN TIME SERIES
By	Mr. Sura Rodpongpun
Field of Study	Computer Engineering
Thesis Advisor	Assistant Professor Chotirat 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 Supot Teachavorasinskun, Ph.D.)

THESIS COMMITTEE

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

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

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

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

. . . . . External Examiner
(Songpol Ongwattanakul, Ph.D.)

สุระ รอดพงษ์พันธ์: การค้นพบเอพิโสดที่เกิดบ่อยในข้อมูลอนุกรมเวลา. (DISCOVERING FREQUENT EPISODES IN TIME SERIES) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ. ดร.โชติรัตน์ รัตนามหัทธนะ, 252 หน้า.

การค้นพบเอพิโสดที่เกิดบ่อยนั้นนับเป็นโจทย์ที่ท้าทายเป็นอย่างมาก ซึ่งเอพิโสดนั้น คือ กลุ่มของเหตุการณ์ที่เกิดขึ้นโดยมีอันดับบางส่วนในช่วงระยะเวลาหนึ่ง อย่างไรก็ตามในโจทย์ทางด้านการหาเอพิโสดในข้อมูลอนุกรมเวลาที่เป็นจำนวนจริงนั้น ยังไม่มีการแก้ปัญหาได้ดีเท่าที่ควร ซึ่งส่วนที่ยากที่สุดนั้น คือการที่ข้อมูลอนุกรมเวลานั้นเป็นลำดับของจำนวนจริง แทนที่จะเป็นลำดับของข้อมูลวิยุต ทำให้แบบรูปที่เหมือนกันนั้นมีความแตกต่างกันทางด้านโครงร่างได้ จึงทำให้เป็นการยากในการบ่งชี้แบบรูป การใช้การจัดกลุ่มลำดับย่อยสำหรับข้อมูลอนุกรมเวลานั้นก็สามารถนำมาใช้เพื่อบ่งชี้แบบรูปได้ อย่างไรก็ตามการจัดกลุ่มลำดับย่อยสำหรับข้อมูลอนุกรมเวลานั้น อาจทำให้เกิดปัญาตามมาได้ด้วยสองสาเหตุหลัก คืออันดับแรก การจัดกลุ่มลำดับย่อยสำหรับข้อมูลอนุกรมเวลานั้น ได้มีการถูกอ้างว่าให้ผลลัพธ์ที่ไม่มีความหมาย กล่าวคือการจัดกลุ่มลำดับย่อยสำหรับข้อมูลอนุกรมเวลานั้นจะให้ผลลัพธ์ในรูปของคลื่นรูปไซน์เสมอ ไม่ว่าข้อมูลขาเข้าจะเป็นอย่างไร และอันดับที่สอง งานวิจัยที่พยายายามทำให้การจัดกลุ่มลำดับย่อยสำหรับข้อมูลอนุกรมเวลานั้นมีความหมาย กลับไม่สามารถบ่งชี้เฉพาะแบบรูปที่สำคัญและละทิ้งข้อมูลที่ไม่มีประโยชน์ได้ จึงทำให้เกิดแบบรูปมากเกินความจำเป็น วิทยานิพนธ์นี้จึงได้แก้ปัญหาเหล่านี้โดยเสนอการจัดกลุ่มลำดับย่อยสำหรับข้อมูลอนุกรมเวลาแบบใหม่ เพื่อการบ่งชี้แบบรูปที่มีประสิทธิผลมากขึ้น โดยเฉพาะอย่างยิ่งวิธีการที่นำเสนอนั้น สามารถเลือกบ่งชี้แบบรูปที่มีความสำคัญ ในขณะที่ละทิ้งแบบรูปที่ไม่สำคัญได้ ผลลัพธ์ ที่ได้ทำให้สามารถแปลงข้อมูลอนุกรมเวลาแบบจำนวนจริงไปเป็นลำดับของเหตุการณ์ได้อย่างมีประสิทธิผล จึงทำให้สามารถนำวิธีการค้นพบเอพิโสดที่เกิดบ่อยสำหรับข้อมูลวิยุติที่มีอยู่ มาประยุกต์ใช้ได้อย่างไม่ยาก มากไปกว่านั้นวิทยานิพนธ์นี้ได้ประยุกต์การใช้ไดนามิกไทม์วอร์ปปิง และการหาค่าเฉลี่ยรูปร่างเพื่อเพิ่มประสิทธิผลของงานที่นำเสนอ โดยวิทยานิพนธ์นี้ได้ยืนยันผลลัพธ์โดยการทำการทดลองบนชุดข้อมูลที่หลากหลาย ซึ่งผลลัพธ์จากการทดลองนั้นยืนยันความมีประสิทธิภาพและประสิทธิผลที่ดีของวิธีการที่นำเสนอ

ภาควิชา .....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต .....
สาขาวิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อ อ.ที่ปรึกษาหลัก .....
ปีการศึกษา ..... 2558 .....

5371816121: MAJOR COMPUTER ENGINEERING

KEYWORDS: DATA MINING / SUBSEQUENCE CLUSTERING / TIME SERIES / FREQUENT EPISODES

SURA RODPONGPUN : DISCOVERING FREQUENT EPISODES IN TIME SERIES.

ADVISOR : ASST. PROF. CHOTIRAT RATANAMAHATANA, PH.D., 252 pp.

Frequent episode discovery is one of the most challenging tasks. An episode is a set of partially ordered occurrences of events in a period of time. However, in real-valued time series mining community, frequent episode discovery has not been addressed well. One of the most difficult problems is that rather than a sequence of discrete events, time series is a sequence of real-valued data. A pattern can be varied in shape of consecutive data points, so that events of the same type are difficult to identify. One can utilize Subsequence Time Series (STS) clustering technique to identify events in the time series, so that frequent episode discovery algorithm can be applied. However, the problem is that output of current STS clustering algorithms cannot be used for the frequent episode discovery because of two main reasons. First, the previously proposed STS clustering algorithms were claimed to be meaningless because the outputs of STS algorithms will always converge to sinusoidal form. Second, some recent STS clustering algorithms that claim to produce meaningful results fail to dispose of trivial subsequences. This leads to inflation and redundancy of the patterns. This thesis approaches the problems by proposing a new STS clustering algorithm to effectively identify interesting events in time series. More importantly, the proposed algorithm also discard trivial or inessential subsequences. As a result of STS clustering, the patterns can be considered as discrete events, similar to those used in general episode discovery algorithms. Moreover, this thesis extends the proposed method to dynamic time warping (DTW) distance, shape-based averaging, and proposes an optimization over the usage of DTW. Experiments show that the proposed framework can perform the episode discovery from time series effectively and efficiently.

Department: ..... Computer Engineering ..... Student's Signature .....

Field of Study: .... Computer Engineering .... Advisor's Signature .....

Academic Year: .....2015.....

Acknowledgements

I would like to express my deep and sincere gratitude to my great thesis advisor, Asst. Prof. Dr. Chotirat Ann Ratanamahatana, for her invaluable guidance, patience, and support during my doctoral graduation. She is always a true example of greatness and excellence. She inspires me to have motivation and courage to be better in so many ways. All of her support has made me who I am today and I will forever be grateful.

I am deeply grateful to Prof. Dr. Boonserm Kijsirikul, Prof. Dr. Prabhas Chongstitvatana, and Asst. Prof. Dr. Sukree Sinthupinyo for being my dissertation committee, and giving all valuable comments and suggestions. I also greatly appreciate Dr. Songpol Ongwattanakul for being my advisor since my undergraduate study, and being my dissertation committee.

I also would like to thank Associate Professor Stephen James Redmond at Graduate School of Biomedical Engineering, University of New South Wales, for giving me the opportunity to earn priceless experience when I was a visiting researcher for a year in Australia.

I am in debt to all my teachers in my every school, especially Chulalongkorn University. I really appreciate Dr. Vit Niennattrakul for giving me valuable support and guidance on how to do great research and being a good graduate student. He is always my inspiration by his commitment to excellence. I thank my friend Dararat for motivating me to be a graduate student. I also thank my lovely friends Haemwaan, Thapanan, Pawan, Warissara, Navin, Sorrachai, Phongsakorn, Thanapong, Supasate, Nareeporn, Warisa, Komate, Patoomsiri, Pittipol, Sirinoot, Warawoot, Tanapoom, Kittipat, Kulit and other friends at the the Department of Computer Engineering for valuable friendship and support.

I appreciate the financial support from the Thailand Research Fund and Chulalongkorn University given through the Royal Golden Jubilee Ph.D. Program (PHD/0319/2551 to S. Rodpongpun) for giving me opportunity to contribute this dissertation to the research community.

Additionally, I thank my dearest wife, Chaviwan Rodpongpun, who is always by my side and support me in every way both physically and mentally.

Lastly, with my deepest gratitude, this dissertation is dedicated to my beloved parents for their endless love and support. This dissertation could not have been completed without them.

Contents

List of Tables

List of Figures

CHAPTER I

INTRODUCTION

In the age of information, time series data is one of the most ubiquitous. Time series are the data that are recorded over time. More specifically, they are in a sequence manner. Time series can be from many sources, such as stock market data recorded over a period of time, or heart rate data that come from a wearable heart rate monitor. Figure 1.1 shows the Stock Exchange of Thailand (SET) index from Google Finance (Google Inc., 2016). Time series data mining is a research area that is actively interesting. Many attempts have been made in trying to solve time series mining problems, such as classification (Mueen et al., 2011), clustering (Zakaria et al., 2012), indexing (Camerra et al., 2010), anomaly detection (Kumar et al., 2005), rules discovery (Das et al., 1998), and trend analysis (Drusinsky, 2003; Li-ping and Mei, 2009).

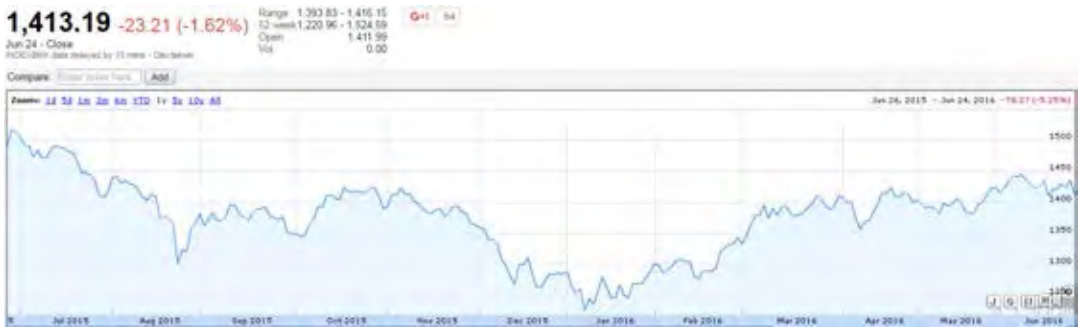

Figure 1.1: SET index from June 26th 2015 to June 26th 2016

One of the most challenging tasks in time series data mining is to discover frequent patterns in data. Frequent patterns is essential for other tasks such as association rule discovery (Das et al., 1998) and prediction (Alvarez et al., 2011; Yaik et al., 2005). Frequent episode discovery is a task to find frequent occurrences of events from sequential data (Achar et al., 2012). More specifically, an episode is partially ordered occurrences of events in a certain time period, and the episode that occurs often is a frequent episode. The first proposed frequent episode discovery algorithm was from (Mannila et al., 1997). After that many new definitions of frequent episodes were proposed by many authors (Achar et al., 2012; Casas-Garriga, 2003; Huang and Chang, 2008; Laxman et al., 2007; Mannila et al., 1997; Meger and Rigotti, 2004). Many approaches use frequent episodes discovery for other applications such as

forecasting and prediction (Martínez-Álvarez et al., 2009), manufacturing (Unnikrishnan et al., 2009), telecommunication (Mannila et al., 1997), network security (Wang et al., 2008), biology (Bouqata et al., 2006; Patnaik et al., 2008), finance (Ng and Fu, 2003), chiller management (Patnaik et al., 2011), and etc.

Most of the proposed frequent episode discovery algorithm are based on an event of single value of discrete data, such as in transactional or temporal database (Achar et al., 2012; Mannila et al., 1997). On the other hand, instead of focusing on a single value, shape of subsequences in the stream can be patterns of interest. In most cases, local patterns or shapes in time series can be very useful. For example in sign language recognition (Yang et al., 2010; Ong and Ranganath, 2005), patterns to be recognized have to be a segment of hand gestures instead of a single point in the recorded data. In that case, it is reasonable to recognize shape occurrences as events. Surprisingly, there was no attempt that applied frequent episode discovery based on shapes of real-valued time series effectively. An attempt in (Sang Hyun et al., 2001) approaches the problem to predict data trend. However, the trend prediction is very limited and cannot be applied to complex shapes. For this reason, general frequent episode discovery algorithms are not capable to find frequent patterns from real-valued time series effectively.

As an illustration, ECG morphology (Sivaraks, 2014) consists of patterns of *P-wave, QRS complex, and T-wave*. Each pattern has its own characteristic or shape. In this case, a single data point in ECG cannot tell anything. If we take this simple example, ECG, as an input for the frequent episode discovery algorithm. A meaningful answer can be the frequent patterns which are *P-wave, QRS complex, and T-wave,* as shown in Figure 1.2.

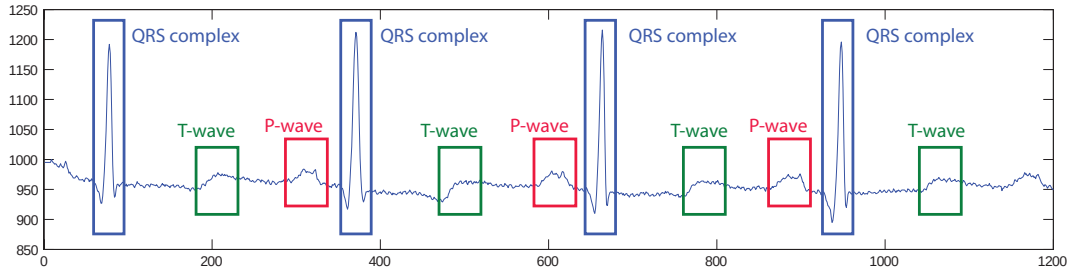

Figure 1.2: ECG morphology consists of *P-wave, QRS complex, and T-wave.*

However, it is not obvious to know the unknown patterns in a real-valued time series. The frequent episodes in this ECG example can be performed more easily if the template of patterns has already been known before running the algorithm. For unknown time series without

predefined patterns, identifying the interesting patterns in the data is very challenging. For example, given a motion data recorded from an equipment for elderly people (Wu et al., 2008), it would be better if frequent complex patterns can be identified to further help understand the people's behavior. Figure 1.3 shows data from SmartCane (Wu et al., 2008), a monitoring device to record walking behavior of elderly people.

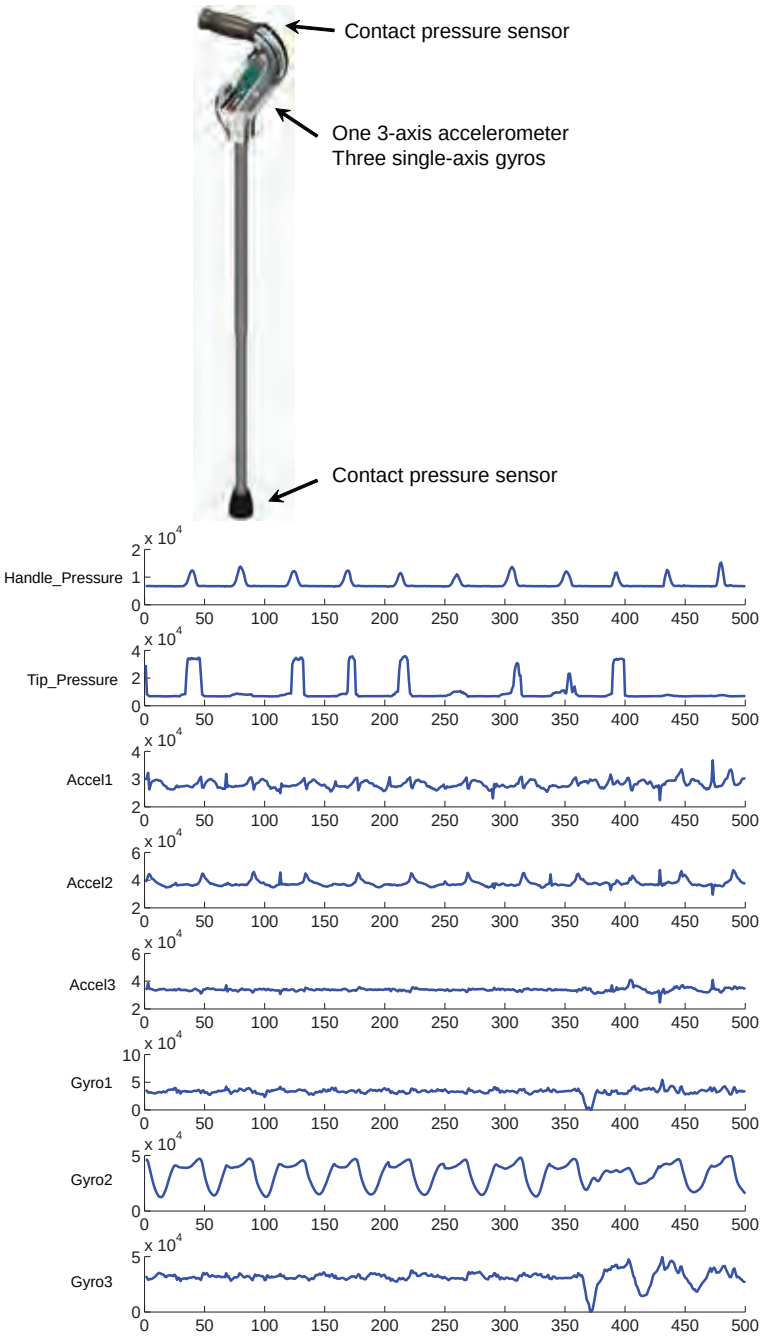

Figure 1.3: Time series recorded from SmartCane equipment (Wu et al., 2008; Niennattrakul, 2010).

There are many attempts proposed to solve this problem of finding unknown interesting patterns. The most well known method is applying Time Series Subsequence (STS) Clustering to extract sub-patterns in a time series. Some research works attempt to use other data discretization techniques (Camerra et al., 2010) to make time series discrete and then perform the clustering. However, there are studies stated that most STS clustering algorithms are meaningless (Keogh and Lin, 2005) due to the fact that most STS Clustering algorithms return output that are not related to the input. More specifically, despite different characteristics of the time series input, the output cluster centers will always be in a form of sine waves. Therefore, the algorithms that use STS clustering as a subroutine (Yairi et al., 2001; Jin et al., 2002; Das et al., 1998; Li et al., 1998; Cotofrei and Stoffel, 2002) will also fail to produce their correct outputs, especially in rule discovery algorithms. On the other hand, frequent patterns or rule discovery algorithms with other discretization techniques (Sarker et al., 2005; Pradhan and Prabhakaran, 2009b; Li et al., 2006; Wan et al., 2007; Lutsiv, 2007) cannot perform well enough due to the limitation of the discretization itself. To emphasize, many parameters have to be chosen. Also, some characteristics of the data would be lost after discretization, and sometimes, similar subsequences can be different after the discretization.

The latest work by Niennattrakul (Niennattrakul, 2010) proposed an STS clustering algorithm called 2STSC. The 2STSC algorithm shows that meaningful STS clustering results can be achieved. Niennattrakul explained that the reason of meaningless results are from trivial match subsequences. For that reason, he proposed an STS clustering algorithm using Dynamic Time Warping (DTW) as a distance measure to better group the same patterns together. Moreover, shape-based averaging methods are proposed and used in his STS clustering algorithm to average the trivial subsequences while keeping the shape of the patterns.

However, one of the problems is that the algorithm clusters all of the subsequences from a sliding window method. Because all of the trivial patterns are clustered, the patterns will be inflated and redundant. In fact, some subsequences such as noises, outliers or some trivial patterns should not be clustered. For instance, in sign language recognition, transitions between consecutive signs, called movement epenthesis (Yang et al., 2010; Ong and Ranganath, 2005), have to be discarded. As a result, it is impossible to identify interesting events in time series due to the fact that every single subsequence is assigned to a cluster. Figure 1.4 illustrates that the trivial patterns are clustered by 2STSC (Niennattrakul, 2010).

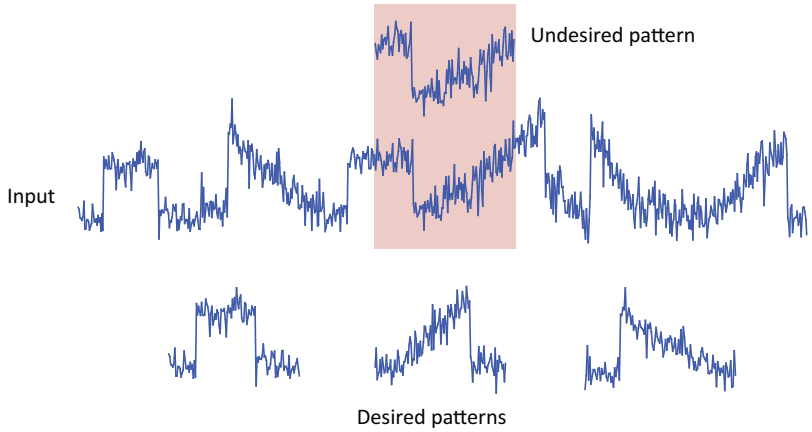

Figure 1.4: Trivial patterns are clustered by 2STSC

The challenges of discovering frequent episodes in real-valued time series can be summarized as follows.

1. Frequent episode discovery from real-valued time series based on shape of the patterns is not well addressed. The reason is that instead of considering shape of the patterns, most of current frequent episode discovery algorithms consider every single discrete value to be an event.

2. STS clustering algorithms can be used to identify interesting patterns in time series. However, most of the approaches fail to produce meaningful results.

3. Despite producing meaningful results by STS clustering algorithms, subsequences in time series can be trivial. Identifying interesting patterns while discarding the trivial ones is challenging.

This thesis proposes a new frequent episode discovery framework that uses a new STS clustering algorithm, called Selective Subsequence Time Series Clustering (SSTSC), to identify interesting patterns while ignoring trivial ones. More importantly, the proposed SSTSC can identify patterns while maintaining meaningful results. The proposed SSTSC also allows flexibility in terms of input parameters, such as allowing pattern length to be variable, and automatically suggesting total number of clusters. Moreover, the proposed framework is intentionally designed to be divided into common subtasks, so that it is easy for further optimization. As a result of identifying interesting patterns based on their shapes, it can effortlessly adopt the existing frequent episode discovery algorithms (Mannila et al., 1997; Laxman et al., 2007; Achar et al.,

2012) to find the occurrences of frequent time series patterns. The proposed framework is evaluated by experimenting on both real world data and annotated semi-synthetic data to evaluate the effectiveness of the algorithm.

Beside the proposed framework, this thesis explores possibilities of using Dynamic Time Warping (DTW) distance and a shape-based averaging technique to improve effectiveness of the framework. Also, this thesis proposes an optimization technique to the application of DTW distance in the framework by designing a bounding technique to reduce execution time of subsequence search using DTW distance.

Next is the summary of the objectives, scopes, research methodology, and contributions.

1.1 Objectives of the thesis

This thesis aims to design a frequent episode discovery framework for real-valued time series data. The following are objectives of this thesis.

- To design a new Subsequence Time Series (STS) clustering algorithm to identify interesting patterns in the real-valued time series, and using the patterns as events for the frequent episode discovery.
- To avoid overly-identified patterns in STS clustering algorithm, because identifying trivial patterns can cause inflation and redundancy of the patterns.
- To explore the use of Dynamic Time Warping (DTW) Distance and shape-based averaging scheme to improve effectiveness of the proposed STS clustering algorithm.
- To optimize the performance of using DTW distance in the framework.

1.2 Scopes of the thesis

The following are the scopes of this thesis.

- This thesis focuses on frequent episode discovery algorithm for time series data, whose frequent events are determined based on interesting patterns discovered from the new STS clustering algorithm.

- The following performance measurements are used to evaluate the performance of the algorithms.
  - Shape-based Meaningfulness Measurement (SMM) (Niennattrakul, 2010)
  - Rand index
  - Precision, Recall, and F-Measure
  - Frequent patterns visualization
- Datasets used in this thesis are from the UCR time series data archive (Keogh et al., 2011), and some medical datasets are from PhysioNet (Goldberger et al., 2000).

1.3 Research methodology

- Study and research on general topics related to time series data mining.
- Survey on potential topics of the thesis including classification, clustering, indexing, subsequence matching, and motif discovery.
- Review literatures related to frequent episode discovery on real-valued time series.
- Research and identify problems along with their causes related to frequent episode discovery algorithm on real-valued time series.
- Design a framework for identifying interesting patterns for frequent episode discovery in real-valued time series.
- Explore the usage of Dynamic Time Warping distance and shape-based averaging to improve effectiveness of the designed framework.
- Design a new subsequence matching algorithm to improve efficiency of the use of DTW in the framework.
- Evaluate proposed framework and algorithm performance.
- Compose the thesis.

1.4 Contributions of the thesis

- This is the first work to address frequent episode discovery problem on real-valued time series effectively.

- The proposed framework is shown to be able to be applied in real world applications.
- This thesis proposes a new frequent episode discovery framework. The following are details of the proposed framework.
  - The framework includes a new STS clustering algorithm for identification of interesting patterns, while trivial patterns are discarded.
  - The framework uses a compression based objective function to perform clustering.
  - The framework allows variability in length of the patterns.
  - The framework provides the best cluster number parameter suggestion.
  - The framework is suitable for frequent episode discovery and other applications such as rule discovery and prediction.
- The proposed framework is intentionally designed to be divided into common subtasks, so that it is designed to be manageable for further optimization.
- This thesis proposes effectiveness improvements to the previously proposed STS clustering method by utilizing DTW distance, and shape-based averaging technique.
- This thesis proposes efficiency improvements on using DTW in a subsequence search subtask.

CHAPTER II

BACKGROUND

In this section, background, definition and notation of frequent episode discovery and time series subsequence clustering algorithms along with their relevant topics used in this thesis are explained.

2.1 Frequent episode discovery

Frequent episode discovery is a task to find frequent occurrences of events from sequential data (Achar et al., 2012). More specifically, an episode is partially ordered occurrences of events in a certain time period, and the episode that occurs often is a frequent episode. The frequent episode discovery algorithms have first been proposed by H. Mannila, et al. (Mannila et al., 1997). Variation of the episode discovery later proposed by many authors (Achar et al., 2012; Laxman et al., 2007), mostly concerns about different definitions of frequency of episode. There is a study (Achar et al., 2012) that simplifies various frequent episode discovery algorithms to a unified apriori-based algorithm. Next is the detail of general frequent episode discovery algorithm (Mannila et al., 1997; Achar et al., 2012).

An *event sequence* of length n can be written as $D = \langle (E_1, t_1), (E_2, t_2), ..., (E_n, t_n) \rangle$, where E_i is the i^{th} *event* occurring in the sequence, each E_i is a symbol from a finite set of alphabets Σ, and t_i is the occurrence time of E_i. The sequence is ordered so that, $t_i \leq t_{i+1}$for all i.

For example, the following sequence is an event sequence with 10 events:

$$\langle (A,1), (A,2), (C,3), (B,3), (A,6), (A,7), (C,8), (B,9), (D,11), (C,12) \rangle \tag{2.1}$$

Definition 2.1 (Episode) *An N-node episode α, can be defined as a triplet, $(V_\alpha, \leq_\alpha, g_\alpha)$, where $V_\alpha = \{v_1, v_2, ...v_N\}$ is a set of N nodes, $\leq_\alpha$is a partial order on V_α, and $g_\alpha : V_\alpha \rightarrow \Sigma$ is a mapping that associates each node in α with an event-type from Σ.*

When the $\leq_\alpha$ is a total order, α is a *serial episode*, while α can be a *parallel episode* if the order

is empty. In this thesis, because the patterns in the time series cannot overlap with each other, only serial episode will be considered.

For example, consider a 3-node episode $V_\alpha = \{v_1, v_2, v_3\}$, $g_\alpha(v_1) = A, g_\alpha(v_2) = B, g_\alpha(v_3) = C$, with $v_1 \leq_\alpha v_2 \leq_\alpha v_3$. The episode can be denoted by $(A \rightarrow B \rightarrow C)$.

Definition 2.2 (Occurrence) *An occurrence of episode α in the event sequence D is a map $h : V_\alpha \rightarrow \{1, ..., n\}$ such that $g_\alpha(v) = E_{h(v)}$ for all $v \in V_\alpha$, and for all $v, w \in V_\alpha$ with $v \leq_\alpha w, t_{h(v)} < t_{h(w)}$.*

In the example event sequence (2.1), the events $(A, 2), (B, 3)$, and $(C, 8)$ construct an occurrence of $(A \rightarrow B \rightarrow C)$. Note that the occurrence of the events in the sequence need not be contiguous.

Definition 2.3 (Subepisode) *An episode $\beta = (V_\beta, <_\beta, g_\beta)$ is a* subepisode *of $\alpha = (V_\alpha, <_\alpha, g_\alpha)$, denoted by $\beta \preceq \alpha$ if there exists a $1 - 1$ mapping $f_{\beta\alpha} : V_\beta \rightarrow V_\alpha$ such that (i) $g_\beta(v) = g_\alpha(f_{\beta\alpha}(v))$ for all $v \in V_\beta$, and (ii) for all $v, w \in V_\beta$ with $v <_\beta w$, we have $f_{\beta\alpha}(v) <_\alpha f_{\beta\alpha}(w)$ in V_α.*

In other words, an episode β is a *subepisode* of α if every event of the same type in β appears in α in the same order. For example, given an episode $\alpha = (A \rightarrow B \rightarrow C)$**,** $(A \rightarrow C)$ is a subepisode of α, while $(C \rightarrow B)$ is not. If β is a subepisode of α, every occurrence of α contains an occurrence of β (Mannila et al., 1997).

The key objective of frequent episode discovery is to find all frequent episodes. The frequent episode is an episode that has *frequency* of occurrence higher than a user-defined threshold. There are many attempts to define different episode frequencies (Casas-Garriga, 2003; Huang and Chang, 2008; Iwanuma et al., 2004; Mannila and Toivonen, 1996; Mannila et al., 1997; Meger and Rigotti, 2004) along with algorithms to discover the frequent episode efficiently.

A *span* is a general term used in literatures. Given an occurrence h of an *N*-node episode α and $V_\alpha = \{v_1, v_2, ...v_N\}$, the *span* is equal to $(t_{h(v_N)} - t_{h(v_1)})$. In other words, the *span* is a constraint that every event in an episode has to appear within a specific period of time. Consequently, the *frequency* of an episode is obtained by counting only the episode's occurrences

that have their *span* lower than a user-specified limit (Achar et al., 2012).

An apriori-style level-wise method is the most popular method of frequent episode discovery (Achar et al., 2012). Considering the fact that an episode can be frequent if its subepisode is frequent, the method uses a "candidate generation" step to create candidates (potential frequent episodes) of size k by combining frequent episode of size $k-1$. The next step is "frequency counting" that counts the frequencies of all candidates by a chosen definition of frequency, and then determines frequent candidates.

2.2 Rule discovery from frequent episodes

Rule discovery is one of the well-known data mining tasks. In general association rule mining (Schluter and Conrad, 2011), rules can be obtained from a transactional database. Let X and Y be two itemsets, the basic association rule is in a form of $X \Rightarrow Y$, which means when X occurs in a transaction, therefore, with high probability, Y will occur in the same transaction. For example, in the market basket analysis, customers who buy a keyboard are most likely to buy a mouse too. When time involves with the data, the mentioned basic association rule above is unnecessary. Temporal rule mining (Das et al., 1998) is then introduced. The simple temporal rule is written as $X \Rightarrow^{T} Y$, which means when event X happens, then it is most likely to see Y happens within time T.

The more complex temporal rule is called sequential rules or sequential patterns proposed in (Agrawal and Srikant, 1995; Schluter and Conrad, 2011). This form of rules describes sequentially occurrence of events. For example, customers who buy Harry Potter Vol. 1-6 will most likely buy Harry Potter Vol.7. This kind of rules can be described as $X_1, X_2, X_3, ..., X_n \Rightarrow^{V,T} Y$. In other words, if events $X_1, X_2, X_3, ..., X_n$ happen within time window V, then Y should happen thereafter within time T.

Other variations of basic or temporal association rules is proposed according to some specific applications. The quantitative association rule (Martinez-BallesterosF et al., 2011) is rules in a form of $X_1(a) \wedge X_2(b) \Rightarrow Y$. This type of rules is used when attributes X_1 and X_2 reach the threshold a and b, then Y will happen sequentially.

Another type of rules is Cyclic/calendar-based association rules (Li et al., 2001), which is used when cyclical events occur. For instance, sandwiches and milk will be bought together every morning. This type of rules can be written as $X \Rightarrow_{l,o} Y$, which means X and Y will

happen together every l unit of time in the o unit of time.

To be more sophisticated, when rules can explain events from different transactions, inter-transactional association rules are introduced (Morchen and Ultsch, 2004). Consider a circumstance when stock X_1 is steep rising together with stock X_2 in the same day, stock Y would be rising in the next day with high probability. Note that this type of rule can be written the same as the basic or temporal association rules, while applying for events from different transactions.

For the rules from frequent episodes, episode rules can be obtained from the results of the frequent episode discovery (Mannila et al., 1997). In this case, given an event sequence D, frequency fr of an episode is simply counted by how many *span* windows w in a set of all windows W have an occurrence of the episode. Formally, $fr(\alpha, W) = \frac{|\{w \in W | \alpha \, occurs \, in \, w\}|}{|W|}$.

For example, given an episode $\alpha = (A \rightarrow B \rightarrow C)$ and its subepisode $\beta = (A \rightarrow B)$, a rule $(A \rightarrow B) \Rightarrow (C)$ can be obtained. Suppose $fr(\beta, W) = 4.2\%$, which means β can be found in 4.2% of windows, and $fr(\alpha, W) = 4.0\%$. It can be estimated that after seeing a window with A and B, C will follow in the same window with a chance of 0.95. Formally, an episode rule is $\beta \Rightarrow \alpha$,such that $\beta \preceq \alpha$. The *confidence* of the episode rule is the fraction $\frac{fr(\beta,W)}{fr(\alpha,W)}$. In other words, the *confidence* can be described as the conditional probability of the occurrence of α in a window, given that β occurs in the window.

However, the rule discovery is an application example of this thesis. It is not in the scopes of this thesis, since the scopes only cover frequent episode discovery level.

2.3 Subsequence Time Series (STS) clustering

In this section, explanation of Subsequence Time Series (STS) clustering will be provided. First, definitions of time series and subsequences are defined as follows:

The algorithm takes real-valued *time series* as an input. It will begin with the definition of *time series*.

Definition 2.4 (Time Series) *A time series T of length m is an ordered sequence of real value data, where $T = (t_1, t_2, ..., t_m)$.*

The proposed approach takes a sequence of time series T as an input and extracts it to a set of

subsequences.

Definition 2.5 (Subsequence) *A subsequence of length n of time series T is $S_{i,n} = (t_i, t_{i+1}, ..., t_{i+n-1})$, where $1 \leq i \leq m - n + 1,\ n \leq m$.*

Suppose there is a time series T of length m, STS clustering clusters the subsequences $\tilde{S} = \{S_{i,n} | 1 \leq i \leq m - n + 1,\ n \leq m\}$ of the time series T. In other words, STS clustering is the whole time series clustering that takes every subsequence as individual time series. Therefore, general clustering algorithms such as *k*-means clustering or hierarchical clustering can be applied. However, consecutive subsequences can be trivial. More specifically, for the subsequences that have overlap region, the overlapping part will be of the same value. Applying general clustering algorithms to the STS clustering problem can cause poor or meaningless result (Keogh and Lin, 2005). For this reason, STS clustering algorithms need to take the problem of meaninglessness into account.

2.4 Similarity measure

Most time series mining algorithms require a similarity measure or a distance metric to measure how similar a pair of time series (or subsequences) are.

2.4.1 Euclidean distance measure

To measure distance between two subsequences, the Euclidean distance that has been widely used in time series domain is considered. The distance is shown below.

$$Dist(X_{i,n}, X_{j,n}) = \sqrt{\sum_{k=0}^{n-1} (x_{i+k} - x_{j+k})^2} \tag{2.2}$$

where $X_{p,q}$ is a subsequence of length q started at position p

2.4.2 Dynamic Time Warping distance measure

Dynamic Time Warping (DTW) distance measure (Keogh and Ratanamahatana, 2005) is a well-known shape-based similarity measure for time series data. It uses a dynamic programming technique to find an optimal warping path between two time series. Suppose we have two time series sequences, a sequence X of length n and a sequence Y of length m. The distance is

calculated by the following equation.

$$D(X_{1...n}, Y_{1...m}) = d(x_n, y_m) + min \begin{cases} D(X_{1...n-1}, Y_{1...m-1}) \\ D(X_{1...n}, Y_{1...m-1}) \\ D(X_{1...n-1}, Y_{1...m}) \end{cases} \tag{2.3}$$

where $D(X_{1...n}, \phi) = D(\phi, Y_{1...m}) = \infty$, $D(\phi, \phi) = 0$, and ϕ is an empty sequence. Any distance metric can be used for $d(x_i, y_j)$, including L1-norm, i.e., $d(x_i, y_j) = |x_i - y_j|$.

2.4.3 DTW with global constraint

A global constraint efficiently limits the optimal path to give a more suitable alignment. Recently, an *R-K band* (Ratanamahatana and Keogh, 2004), a general model of global constraints, has been proposed. *R-K band* represents a global constraint by a one-dimensional array R, i.e., $R = (r_1, r_2, ..., r_i, ..., r_n)$, where n is the length of time series, and r_i is the height above the diagonal in the *y*-axis and the width to the right of the diagonal in the *x*-axis. Each r_i value is arbitrary, making the *R-K band* an arbitrary-shaped global constraint.

2.4.4 Lower-bounding function for DTW distance

Although DTW outperforms many other distance measures, it is known to require huge computational complexity. Therefore, LB_{Keogh} has been proposed to speed up similarity search. $LB_{Keogh}(Q, C)$ between the query sequence $Q = (q_1, q_2, ..., q_i, ..., q_n)$ and a candidate sequence $C = (c_1, c_2, ..., c_i, ..., c_n)$ can be computed as follows.

$$LB_{Keogh}(Q, C) = \sum_{i=1}^{n} \begin{cases} |c_i - u_i| & ; if\ c_i > u_i \\ |l_i - c_i| & ; if\ c_i < l_i \\ 0 & ; otherwise \end{cases} \tag{2.4}$$

where $u_i = max\{q_{i-r_i}, ..., q_{i+r_i}\}$and $l_i = min\{q_{i-r_i}, ..., q_{i+r_i}\}$are envelope elements calculated from a global constraint $R = (r_1, r_2, ..., r_i, ..., r_n)$.

2.5 Z-normalization

The two time series sequences are compared using any similarity measures; all the data should first be normalized. *Z*-normalization (Han et al., 2006) that makes the value of mean

and standard deviation of a time series to be zero and one will be used. Given a subsequence $T_{i,n} = (t_i, t_{i+1}, ..., t_{i+n-1})$ whose mean is μ and standard deviation is σ. The normalized time series is $T'_{i,n} = \left(t'_i, t'_{i+1}, ..., t'_{i+n-1}\right)$, where $t'_k = \frac{t_k - \mu}{\sigma}$.

2.6 Uniform scaling

Many research works show that uniform scaling technique can improve performance in terms of accuracy (Fu et al., 2008; Yankov et al., 2007). Specifically, a subsequence $T = (t_1, ..., t_i, ... , t_m)$ can be shrunk/stretched, by specifying a scaling factor $f \geq 1$, to a new time series $T' = \left(t'_1, ..., t'_j, ..., t'_n\right)$, where $t'_j = t_{\lceil j \cdot n/m \rceil}$, $\lceil m/f \rceil \leq n \leq \lfloor m \cdot f \rfloor$.

In this thesis, input time series can be extracted to subsequences of different lengths. In detail, STS clustering algorithm in section 3.3 takes 2 parameters, w and f, the window length and the scaling factor, respectively. Subsequences of length from $\lceil w/f \rceil$ to $\lfloor w \cdot f \rfloor$ are extracted, then uniform scaling is used to make them the same length of w before clustering.

2.7 Motif discovery

A subsequence motif (Tang and Liao, 2008; Frith et al., 2003; Li et al., 2012; Mueen et al., 2009; Yingchareonthawornchai et al., 2013; Yankov et al., 2007) is the most similar pair of subsequences in time series data. Many research works have proposed motif discovery algorithms, trying to improve performance in terms of speed and accuracy. In this thesis, MK algorithm in (Mueen et al., 2009), which is considered the fastest algorithm to find a pair of motifs by using the Euclidean distance, will be utilized.

2.8 Subsequence search in time series

Subsequence matching algorithm (Wu et al., 2005; Niennattrakul et al., 2009; Rodpongpun et al., 2011) is usually used as a subroutine in many data mining tasks. By giving a query sequence, we can retrieve a subsequence, which is the most similar to the query, from a longer time series. In this thesis, the Euclidean and DTW distance are used as a distance measure to compare the query sequence with all the extracted subsequences

2.9 Time series averaging

In the clustering part of this thesis, an averaged time series sequence of each cluster needs to be calculated. This section describes averaging methods used in this thesis.

2.9.1 Amplitude averaging

Amplitude averaging method (Niennattrakul, 2010) calculates a mean time series by averaging every pair of points or dimensions of two time series of the same length. Given two time series of length n, $A = (a_1, a_2, ..., a_i, ..., a_n)$ and $B = (b_1, b_2, ..., b_i, ..., b_n)$, a mean time series $C = (c_1, c_2, ..., c_i, ..., c_n)$ is calculated by $c_i = \frac{a_i + b_i}{2}$, for all $i = 1, 2, ..., n$.

In addition, for the input sequences that have different weights, for example A has weight ω_A, and B has weight ω_B, the mean time series C can be calculated by $c_i = \frac{\omega_A \cdot a_i + \omega_B b_i}{\omega_A + \omega_B}$, for all $i = 1, 2, ..., n$.

Therefore, a mean time series $X = (x_1, x_2, ..., x_i, ..., x_n)$ can be computed from a set of m time series of length n, $\mathbb{T} = \{T_1, T_2, ..., T_j, ..., T_m\}$, with different weights ω_{T_j} by $x_i = \frac{\sum_{T \in \mathbb{T}} \omega_{T_j} t_i}{\sum_{T \in \mathbb{T}} \omega_{T_j}}$, for all $i = 1, 2, ..., n$ and $j = 1, 2, ..., m$.

2.9.2 Shape-based averaging

Unlike the amplitude averaging, shape-based averaging method, called CDTW (Niennattrakul et al., 2012), averages pairs of data points according to a warping path of DTW distance between two time series. When calculating DTW distance, a path matrix is created to store an index of the adjacent element that has minimum cumulative distance, and a warping path is traced back from the last element to the first element (Niennattrakul, 2010). An averaged time series is then computed along the warping path.

Given a path $W = (w_1, w_2, \ldots, w_k, \ldots, w_N)$, where $w_k = (i_k, j_k)$ is k^{th} coordinate in the DTW's optimal path of two sequences A and B. Therefore, a position c'_{k_x} of a data point in a new averaged sequence C' is determined by $C'_{k_x} = \frac{\omega_A \cdot i_k + \omega_B \cdot j_k}{\omega_A + \omega_B}$, and an amplitude c'_{k_y} of a data point in a new sequence C' is determined by $C'_{k_y} = \frac{\omega_A \cdot a_{i_k} + \omega_B \cdot b_{j_k}}{\omega_A + \omega_B}$, where ω_A and ω_B are the weights of sequences A and B, respectively.

Consequently, the length of the sequence C' can be equal to or longer than the length of original sequences. For this reason, re-interpolation is applied. In (Niennattrakul et al., 2012), CDTW averaging method uses a cubic-spline interpolation (Burden et al., 1997) to re-sample the averaged time series C' to be the same length of A and B.

CHAPTER III

TIME SERIES FREQUENT EPISODE DISCOVERY FRAMEWORK

This chapter provides a complete detail of the proposed time series frequent episode discovery framework, but before going to the algorithm, related work will be discussed next.

3.1 Related work

The related work section will be divided into two topics. First, related works about frequent episode discovery and other related mining techniques will be explained. This part includes frequent episode discovery algorithms, other related frequent pattern mining algorithms in time series, and related association rule discovery algorithms. Second, a topic specifically about techniques to transform real-valued time series to event sequences, which mainly focus on subsequence time series clustering techniques and some discussion on motif discovery, will be described.

3.1.1 Frequent episode discovery and related mining techniques

The frequent episode discovery algorithms have been active since the first work proposed by H. Mannila, et al. (Mannila et al., 1995). Variations of the frequent episode discovery algorithms are proposed by many authors (Achar et al., 2012; Laxman et al., 2007). In (Mannila et al., 1997), window-based, and minimal occurrence-based frequency counting methods are proposed with optimized counting algorithms. Head frequency and total frequency definitions (Iwanuma et al., 2004) are proposed along with counting methods to reduce the redundancy of frequent episodes in (Mannila et al., 1997). There are a non-overlapped, and non-interleaved frequency introduced in (Laxman, 2006) to have constraints that the frequent episodes should not be overlapped or interleaved. Other frequency definition is distinct occurrence-based (Mahesh Joshi and Kumar, 1999) that do not allow an event to be in multiple episodes.

There are many frequent episode discovery algorithms that are proposed for various types of data scheme. A work from (Srivatsan Laxman, 2012) proposes an episode mining algorithm for dynamic event streams. The work provides approximation algorithm to discover frequent

episodes from streaming data. Another type of data is uncertain sequence data, which is the data that has probability of occurrence (Wan et al., 2013b,a). The uncertain data needs to take the probability of event occurrences into account.

In spite of that, there are many researches on frequent episode discovery. Most of the works are based on input event sequence to be discrete. In other words, an event is considered to be only one data point of known type. However, it will be a problem if it needs to consider more complex real-valued patterns or shapes, instead of one discrete data point as events in the sequence. To do so, all of the above mentioned frequent episode discovery algorithms cannot be used. This thesis then proposes to solve this problem.

Some research works use frequent episode discovery for other applications such as prediction or forecasting, such as in (Martínez-Álvarez et al., 2008) and (Martínez-Álvarez et al., 2009), the authors propose algorithms to forecast electricity price time series by discretizing real-valued time series to event sequences, and then discover frequent episodes for prediction. However, these works are proposed to perform segmentation on electricity price to a period of a day, and also reduce the dimension of the price in a day to a specific set of event types. For this reason, the problem can be considered to be discovering frequent episodes on a single value in time, which is beyond the scope of this thesis that the data are real-valued and do not have predefined segments.

Another type of research that is closely related to frequent episode discovery is association rules discovery. Other than mining rules from discrete event sequences (Qin and Shi, 2006), there are works proposed for real-valued time series. The most mentioned technique for discovering association rules from time series is proposed by G. Das, et al. (Das et al., 1998). After extracting an input time series to subsequences, a traditional k-means clustering algorithm is applied to the set of all subsequences. Accordingly, set of temporal association rules are obtained from those clustered subsequences by using Apriori algorithm (Agrawal and Srikant, 1994). Also, variations of the method by G.Das, et al. are proposed (Pradhan and Prabhakaran, 2009b; Lutsiv, 2007). In the same manner, many attempts utilize fuzzy approach to mine rules from time series (Wang and Chen, 2009; Pradhan and Prabhakaran, 2009a). However, symbolization is needed to convert real-valued time series data to discrete subsequences, and many threshold parameters need to be determined to obtain association rules. A well-known discretization technique, SAX (Warasup and Nukoolkit, 2006), is utilized to discretize the input time series to be a set of symbols. Subsequently, an apriori-like algorithm is applied to acquire association rules

from the discretized subsequences. As mentioned above, this type of discretization makes time series lose their characteristics, and also, many parameters are required to achieve an optimal results. The latest work from (Shokoohi-Yekta et al., 2015) proposes an algorithm to discover meaningful rules in time series. The algorithm finds a motif in the time series, and then uses an MDL technique for an objective function to find an optimal split point in the motif pattern that gives an optimal rule. This approach works well with real-valued time series; However, there are limitations when the rule needs to be in a form of $A \Rightarrow B$ only. Also, even though the authors propose *maxlag* value to allow some gap between A and B, the motif that is a baseline to find the rule cannot have the *maxlag* value. For this reason, this work (Shokoohi-Yekta et al., 2015) does not solve the same problem of this thesis, which is to discover frequent episodes that are constructed from important shapes or patterns in time series.

This thesis proposes a new subsequence time series clustering algorithm to identify important patterns in the time series while discarding trivial patterns to avoid pattern inflation and redundancy. In other words, the real-valued time series is transformed to an event sequence, and then any frequent episode discovery algorithms can be applied to find frequent episodes. Next subsection will discuss about related works according to STS clustering method.

3.1.2 Discovering of patterns in real-valued time series

One may consider motif discovery techniques (Frith et al., 2003; Li et al., 2012; Mueen et al., 2009; Tang and Liao, 2008; Yankov et al., 2007) for discovering important patterns in time series. However, in time series mining community, most proposed motif discovery algorithms focus on finding one pattern at a time. More specifically, motif discovery algorithms find a pair of subsequences that are most identical based on a distance measure. Although some methods can find more than one pattern (Mueen et al., 2009; Yingchareonthawornchai et al., 2013), the motif discovery algorithms focus on finding the motif patterns without identifying all the subsequences that are similar to the motif patterns. For this reason, subsequence time series (STS) clustering technique is more relevant to identifying all important patterns and their labels (clusters).

In time series mining communities, clustering task has always been receiving much attention. However, most works focus on clustering individual time series whereas clustering subsequences of a single long time series, the problem considered in this thesis, is not well resolved for using with frequent episode discovery. The most referenced work is the one that uses

STS clustering as a subroutine for rule discovery (Das et al., 1998). The proposed work from Das et al. uses k-means clustering with all subsequences from an input time series to discretize to real-valued time series to discrete events. All of the subsequences are extracted by using a simple sliding window method. After obtaining the discretized sequence, a rule discovery algorithm is applied. Another work (Oates, 1999) proposes an algorithm to identify distinctive subsequences in real-valued time series. However, the method proposed by (Oates, 1999) is inapplicable to single time series problem (Zolhavarieh et al., 2014). Oates also proposes another method called PURUSE (Oates, 2002) to find recurring patterns in time series. The work has two approaches that are a supervised learning and an unsupervised learning. Experiments are based on data from utterances, and data from robotic sensors. A study by Frith et al. (Frith et al., 2003) proposes a model to find clusters of motifs in DNA sequences using maximal log likelihood ratios. They also propose methods that has higher efficiency, which are Cister, Comet, and cluster-buster.

Surprisingly, it has been discovered that the STS clustering methods used in previous works are meaningless (Keogh and Lin, 2005). Because the algorithm tries to cluster every single extracted subsequences, their output turns out to always be in sine waves regardless of what the input sequence looks like. For this reason, the authors claim that the other methods that use STS clustering as a subroutine will fail to produce their meaningful results as well. This work by Lin et al. (Lin et al., 2003) also proposes a new measurement to measure meaningfulness of STS clustering algorithm.

After the meaninglessness claim, there are approaches proposed by many authors. A study by (Chen, 2005) is the first attempt to solve the meaninglessness problem. The study comes to conclusions that the subsequences can be significant, and the distance measure in delay space can lead to meaningful result. A work from (Denton et al., 2009) proposes kernel-density-based method for clustering of time series subsequence. The work shows that noise elimination with the proposed kernel-density-based clustering can be significant in the clustering application (Zolhavarieh et al., 2014). Another work proposes a new cluster shape distance to be used with STS clustering (Goldin et al., 2006). The approach defines a shape by a sorted list of the pairwise Euclidean distance between their centroids. The authors also provide two algorithms related to the proposed new distance measure that matches cluster centroids. First algorithm creates smaller fingerprints while the second approaches with higher accuracy. However, this work only shows the effectiveness with only one dataset of 10 sequences. A work from (Kumar et al., 2006) proposes WaveSim and adaptive WaveSim transform, which are a

perspective of wavelet transform. The algorithm uses a hierarchical tree based method for STS clustering using the proposed WaveSim transform. Chen (Chen, 2007a) proposes another work for the STS clustering to achieve meaningful result. The method restricts the clustering space to extend over the visited region by the time series data in the subsequence vector space. The author claims that the method can achieve the meaninglessness problem. Another work from Chen (Chen, 2007b) provides an alternative distance measure that is Euclidean distance in the delay vector space. The work advises that the STS clustering can be meaningful. However, results of the study are limited given certain barriers (Zolhavarieh et al., 2014). A theoretical analysis of the STS clustering is proposed by (Fujimaki et al., 2008). The analysis is based on a frequency-analysis viewpoint, and provides mathematical background about how STS clustering algorithms generates output as sine waves. The authors also propose a clustering method, Phase Alignment STS clustering, using phase alignment preprocessing. Another interesting work from (Denton et al., 2009) proposes a clustering algorithm that uses frequent occurring subsequences to create clusters. Subsequences can be frequent if their occurrences are more frequent than an expected value of a random chance. The algorithm performs on both pattern-based clustering, and kernel-density-based clustering. An algorithm called CONTOUR proposed by (Wang et al., 2009) improves efficiency over discovering discriminating subsequences. The approach uses pruning techniques to make the algorithm more efficient. Li et al. (Li et al., 2012) propose a method of clustering using numerosity reduction and grammar induction based algorithms. However, for long time series, many iterations may be needed to achieve effective results (Zolhavarieh et al., 2014). Yang approaches STS clustering (Yang and Wang, 2014) by proposing a method using phase shift weighted spherical k-means algorithm. The proposed method only shows output example from only one dataset.

To summarize, the meaninglessness problem of the STS clustering is analyzed by many authors (Fujimaki et al., 2008; Idé, 2006; Ohsaki et al., 2009). It now comes to a conclusion that clustering every subsequence extracted by moving a fixed sliding window leads the cluster centers to converge to a form of sine waves. However, those algorithms (Chen, 2005; Denton et al., 2009) require a fixed value of number of clusters k and length of subsequences w, which are not suitable in real world problems. A comprehensive review of most STS clustering algorithms and their relevant algorithms is also reported by (Zolhavarieh et al., 2014).

Niennattrakul proposes a new STS clustering algorithm based on shape-based averaging technique (Niennattrakul, 2010), called 2STSC, which is proven to be meaningful. However, the drawback of the algorithm is that every single subsequence of a time series is clustered.

Therefore, when doing rule discovery, a lot of consecutive subsequences are inflated. As a result, it is very difficult to find the rules with a lot of trivial patterns. Moreover, parameters also needed to be predefined, i.e., the number of clusters k and length of subsequences w. In this case, it can be seen that results of the algorithm would be very sensitive to the parameters as shown in Figure 3.1. A simple CBF example shows that even the cluster number k is set to 3, which is a desirable class number for the CBF data, some subsequences that lie between two major patterns are forced to be clustered. Hence, those undesirable patterns are assigned to a cluster. This then conducts a cluster center to be incorrect and misleading.

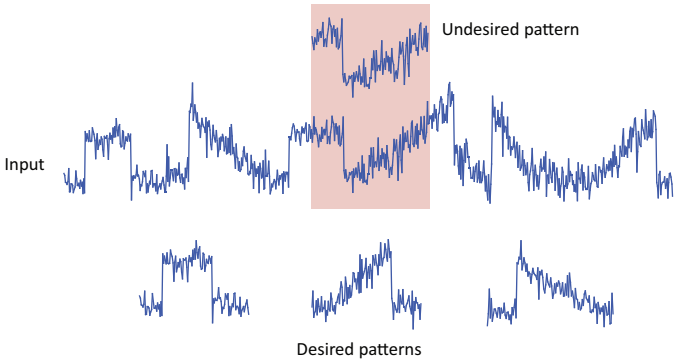

Figure 3.1: Undesired patterns will be forced to be in a cluster

Recently, Rakthanmanon proposes a new STS clustering algorithm (Rakthanmanon et al., 2012) that uses MDL technique to acquire patterns into clusters. It is important to note that some of the similar ideas proposed in (Rakthanmanon et al., 2012) was independently developed in parallel to this thesis work (in section 3.3.3), which is published in (Rodpongpun et al., 2012). This is to declare the originality of this thesis, and to emphasize the impact and contributions of the methods proposed in this thesis.

3.2 Discretization: converting real-valued time series to event sequence

There are a lot of techniques to discretize real-valued time series. Due to the fact that it needs to convert a time series to an event sequence where the events are significant patterns in the time series. Subsequence Time Series Clustering (STSC) algorithm (Zolhavarieh et al., 2014) is the most suitable solution, because the algorithm clusters subsequences into groups where each group consists of similar pattern subsequences. Then, it is easy to mark each clustered subsequence as an event based on its clusters. However, most STSC are claimed to be meaningless. There are some attempts to solve the meaninglessness issue (Niennattrakul, 2010;

Chen, 2007b). The intuition is that we do not need to assign every single subsequence to be events. The consecutive subsequence can be trivial. Especially, when the sampling rate of the data is high, there are a lot of consecutive subsequences that have similar pattern. Therefore, the pattern will be inflated and we cannot get a meaningful result when performing a frequent episode discovery. An algorithm called Selective Subsequence Time Series Clustering (SSTSC) (Rodpongpun et al., 2012) is proposed in this thesis to perform subsequence clustering the way we need. The SSTSC discards some trivial subsequences and clusters only significant ones. For this reason, this thesis proposes this method as a discretization technique to convert a real-valued time series to an event sequence.

Next section will provide details of the proposed algorithm.

3.3 Selective Subsequence Time Series (STS) clustering

Time series clustering (Chen, 2005; Denton et al., 2009; Lai et al., 2010; Keogh and Lin, 2005; Wang et al., 2006) is one of the most popular tasks in time series data mining community (de A. Araújo, 2011; Lee and Tong, 2011; Li and Guo, 2011; Weng and Shen, 2008a,b; Niennattrakul et al., 2012). Most algorithms generally perform whole time series clustering (Wang et al., 2006; Lai et al., 2010). More specifically, those algorithms try to group individual time series instances to a set of clusters. On the other hand, Subsequence Time Series (STS) clustering (Chen, 2005; Keogh and Lin, 2005; Denton et al., 2009), which will be considered in this thesis, has been gaining more popularity. STS clustering algorithm discovers clusters of interesting subsequences within a single time series data stream. This algorithm can be used as a subroutine of other data mining tasks, such as rule discovery (Yairi et al., 2001; Das et al., 1998; Jin et al., 2002), indexing (Li et al., 1998), classification (Cotofrei and Stoffel, 2002), and anomaly detection (Yairi et al., 2001).

Unfortunately, it has been demonstrated that these STS clustering algorithms produce meaningless results (Keogh and Lin, 2005). Because most algorithms use a sliding window to extract subsequences and try to cluster them all, the resulting cluster centers turn out to be some forms of sine waves regardless of the original shape of the patterns in the input data. Therefore, every algorithm that uses this meaningless STS clustering as a subroutine will in turn fail to produce meaningful results as well.

The cause of producing sine waves as outputs has been analyzed by many authors (Fu-

jimaki et al., 2008; Idé, 2006; Ohsaki et al., 2009). They have shown that clustering of every single subsequence leads to meaningless outputs. In fact, some subsequences such as noises or outliers should not be clustered. For instance, consider a speech recognition problem, non-speech segments in a source data has to be determined and removed. Similarly, in sign language recognition, transitions between consecutive signs, called movement epenthesis (Yang et al., 2010; Ong and Ranganath, 2005), has to be discarded. It is shown in Figure 3.2 that meaningful STS clustering can be achieved by ignoring some subsequences. The ECG data from (Goldberger et al., 2000) demonstrate that it is not necessary to include some trivial subsequences in a cluster.

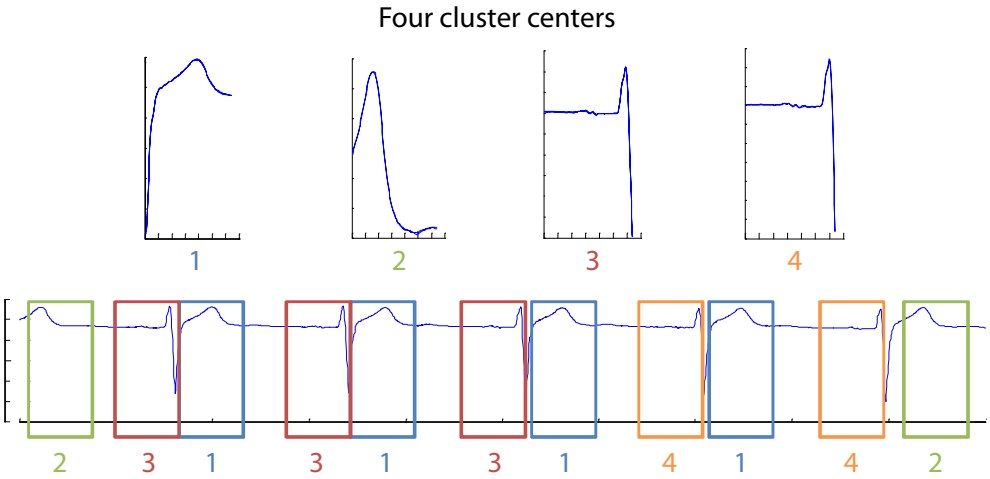

Figure 3.2: Meaningful STS clustering achieved by ignoring some subsequences.

This thesis proposes a new STS clustering framework called Selective Subsequence Time Series Clustering (SSTSC), which performs subsequence clustering to produce meaningful cluster centers. This thesis will show that the cluster centers from the proposed algorithm do represent the actual patterns within the input data, instead of producing sine waves. In essence, this thesis adopts an idea of data encoding to determine proper clusters by clustering only important subsequences. Some subsequences that are not significant will be discarded. On the other hand, because it is hard to exactly specify a window size of the subsequences, the proposed approach allows window size to be varied. The appropriate sliding window size, w, depends on types of data and application requirement. In practice, a user only needs to roughly estimate a value of w, and then the proposed algorithm will determine an appropriate value. However, due to the flexible window length w, the members of clusters could be of different lengths. Moreover, different types of data need different predefined number of clusters k, so the

proposed algorithm automatically determines an appropriate number of clusters depending on characteristics of input data.

The rest of this chapter is organized as follows. Section 3.3.1 will define some definition and notation, followed by problem definition in section 3.3.2 Details of the proposed approach are described in section 3.3.3. Section 3.4 summarizes the frequent episode discovery method that will be used. Section 3.5 shows essential experiments in various domains including real and semi-synthetic data. Finally, conclusion and discussion about future research direction according to the proposed STS clustering algorithm are discussed in section 3.6.

Next will be details of the proposed approach called Selective Subsequence Time Series Clustering (SSTSC) framework. Firstly, it begins with stating the problem definition.

3.3.1 Definition and notation

As mentioned in the chapter 1, this thesis adopts an idea of simple data encryption to determine proper clusters by emulating the clusters as a *codebook.*

Definition 3.1 (Codebook) *A codebook is a data structure used to store codewords, representing repeating parts in an input data. The input data can be compressed by substituting the repeating parts with smaller codeword symbols.*

In this thesis, it emulates cluster centers as the codewords used to represent their member subsequences. Performance of the encoding can be measured by using *Compression ratio* and *Error* defined below.

Definition 3.2 (Compression Ratio) *Compression ratio is a ratio of the data size between before and after compression, including an overhead of construction of a codebook and codeword symbols.*

For example, given a 16-charecter string $S =$ "$ABCDEFGHIJKLMNOP$". Suppose that substrings "ABC" and "HIJ" are similar, we can substitute them with a symbol x, therefore the encoded string $S' =$ "$xDEFGxKLMNOP$". In this case, we can eliminate 6 characters ("ABC" and "HIJ"), but a codeword of size 3, and two x's must be created; thus, the compression is $6 - (3 + 2) = 1$ character, and the *compression ratio* $\tilde{R} = \frac{16-1}{16} = 0.94$.

Definition 3.3 (Error) *Error is a summation of distances from cluster centers to their cluster members.*

For example, the two substrings "ABC" and "HIJ", which are mentioned in the previous definition, are grouped into a cluster C, then a codeword (a cluster center) $\bar{C}$ is created. The *Error* of creating a cluster from those substrings is $\tilde{E}(C) = Dist(\bar{C}, \text{“}ABC\text{”}) + Dist(\bar{C}, \text{“}HIJ\text{”})$.

Next is a problem definition of the proposed STS clustering algorithm.

3.3.2 Problem definition

Input of the proposed algorithm is a single time series data. The problem is to first determine a number of clusters n, and then to group subsequences into proper clusters; some subsequences can be discarded without being assigned to any cluster. The subsequences are extracted using a sliding window approach. The sliding window can be varied in a range specified by a user. For example, it is demonstrated by using a 16-character string $S =$ "$ABCDEFGHIJKLMNOP$" as an input. We use a sliding window w of size 3, and a scaling factor $f = 1.5$; therefore, the length of the subsequences is varied from 2 to 4. The subsequences are extracted into a set $S' = \{\text{“}AB\text{”}, \text{“}BC\text{”}, ..., \text{“}OP\text{”}, \text{“}ABC\text{”}, \text{“}BCD\text{”}, ..., \text{“}NOP\text{”}, \text{“}ABCD\text{”}, \text{“}BCDE\text{”}, ..., \text{“}MNOP\text{”}\}$. The algorithm should produce a set of clusters $\tilde{C} = \{C_1, ..., C_i, ..., C_n\}$. Each cluster consists of its members and a cluster center: $C_i = \{t_{i_1}, t_{i_2}, ..., t_{i_n}, \bar{C}_i\}$, where t_{i_j} is the j^{th} member of the i^{th} cluster, and the $\bar{C}_i$ is the cluster center of the i^{th} cluster.

3.3.3 Clustering method

To form clusters from a set of subsequences, we must iteratively pick one subsequence and assign it to a cluster. However, in the first place, we do not have any predefined cluster yet, and we must make a decision as follows. Intuitively, we can choose two subsequences which are the most similar, to create the first cluster, then the first cluster center is produced. As a result, we can choose other subsequences, which are the most similar to the already created cluster, to be added to the existing cluster; therefore, the cluster center is then updated. Nevertheless, it is better to create a new cluster if there exist two subsequences that are similar to each other more than to the existing cluster center. Moreover, if there are two clusters that can be grouped together, we can decide to merge them to create a new cluster. Thus, we define three operations for producing clusters from a set of subsequences; those are *Create*, *Add*, and

Merge to iteratively select two subsequences to create a new cluster, to assign a subsequence to an existing cluster, and to merge two clusters into a new cluster, respectively.

The proposed approach iteratively selects an operation, which are *Create*, *Add*, and *Merge* to produce a set of clusters. Accordingly, this thesis adopts an idea of data encoding as a heuristic function to choose an optimal operation in each step of clusters construction. This thesis emulates a set of cluster centers as a codebook, where each cluster center is a codeword used to encode the input time series. Some subsequences from the input time series, which are members of a cluster, will be substituted by a small codeword symbol. *Error* of a cluster is determined by a summation of Euclidean distance from the codeword, which is the cluster center, to their member subsequences.

$$\tilde{E}(C_i) = \sum_{j=1}^{m} Dist(t_{i_j}, \bar{C}_i) \tag{3.1}$$

where m is a number of members in the i^{th} cluster.

Increased error $\triangle\tilde{E}$ is obtained after a cluster update.

$$\triangle\tilde{E} = \tilde{E}_{after} - \tilde{E}_{before} \tag{3.2}$$

Compression ratio $\tilde{R}$ is determined by calculating data reduction of the original subsequence including overhead from codeword construction and codeword symbol substitution.

In detail, *Compression ratio* and *Increased Error* for each operation are described below.

1. *Creat*e: Create a new cluster C from two subsequences P of length u, and Q of length v. A new codeword $\bar{C}$ of length w is obtained by merging P and Q. The length of input time series l is reduced by $u + v$. The overhead is added by the codeword construction and the substitution of P and Q by two of a codeword symbol "x" of size 1.

$$\triangle\tilde{E} = \tilde{E}(C) \tag{3.3}$$

$$\tilde{R} = [(u + v) - (w + 2)]/l \tag{3.4}$$

2. *Add*: Update an existing cluster C to a new cluster C' by adding a subsequence P of length u, and update the codeword $\bar{C}$ to $\bar{C}'$. This operation reduces the length of input time series l by u. The overhead is added by substituting P by "x" of size 1.

$$\triangle\tilde{E} = \tilde{E}(C') - \tilde{E}(C) \tag{3.5}$$

$$\tilde{R} = (u-1)/l \tag{3.6}$$

3. *Merge*: two clusters C_i and C_j are merged into a new cluster C'. A codeword of length w is reduced.

$$\triangle\tilde{E} = \tilde{E}(C') - [\tilde{E}(C_i) + \tilde{E}(C_j)] \tag{3.7}$$

$$\tilde{R} = w/l \tag{3.8}$$

The problem can be considered as a search space consisting of nodes of the three operations, which is illustrated in Figure 3.3. The proposed approach uses greedy method to iteratively select a node that has minimal *Increased Error*. To do this, as shown in Figure 3.4, this thesis applies MK motif discovery algorithm (Mueen et al., 2009) to discover a pair of subsequences, which has minimal Euclidean distance, to be the best node for the *Create* operation. To search for the optimal *Add* node, all codewords are used as queries for the subsequence matching algorithm to locate the best subsequence to be added to an existing cluster. The optimal *Merge* node can be determined by searching all nodes, due to its small number of nodes. Note that the subsequences can be of different lengths, so this thesis uses a uniform scaling technique to make them the same length w before applying the motif discovery and the subsequence matching algorithms.

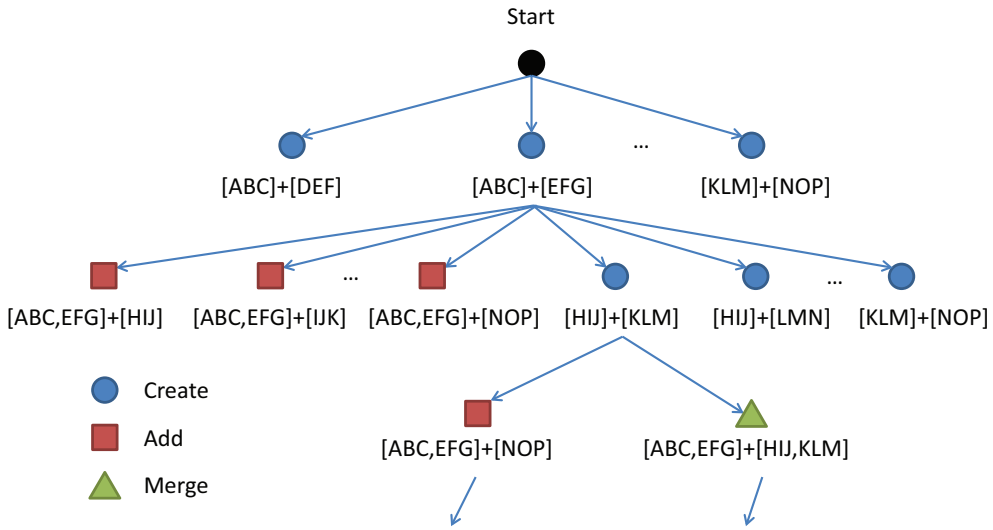

Figure 3.3: The search space consists of *Create*, *Add*, and *Merge* operations.

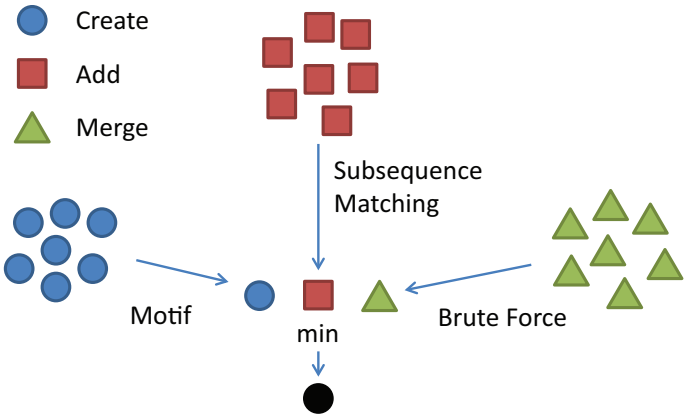

Figure 3.4: The optimal node can be determined by using motif discovery and subsequence matching algorithms.

To determine a proper number of clusters, we must choose a state of creating clusters that provides large *compression ratio* while producing less *error*. From the compression-error plot shown in Figure 3.5, it is obvious that there is a *knee point* in the graph where *errors* are dramatically increased. It means applying an operation after that point will lose the clustering accuracies. Thus, we return clusters in that state as a result of the algorithm.

The *knee point* (Salvador and Chan, 2004) can be determined by scanning along the *compression ratio-error* curve point by point. For each point, calculate two linear regression lines: one from all the points on the left side of the point and another one from all the points on the right side of the point. The *knee* is the point where the summation of errors from the two

lines is minimum.

To find that point, this thesis determine linear fitting function to the compression-error graph and choose a point that gives minimum residual value to the fitting function as shown in Figure 3.5. Consider a special case that a user want to specify the number of clusters k, the proposed algorithm can effortlessly handle it by choosing a latest state that has the number of clusters equal to the one specified by the user.

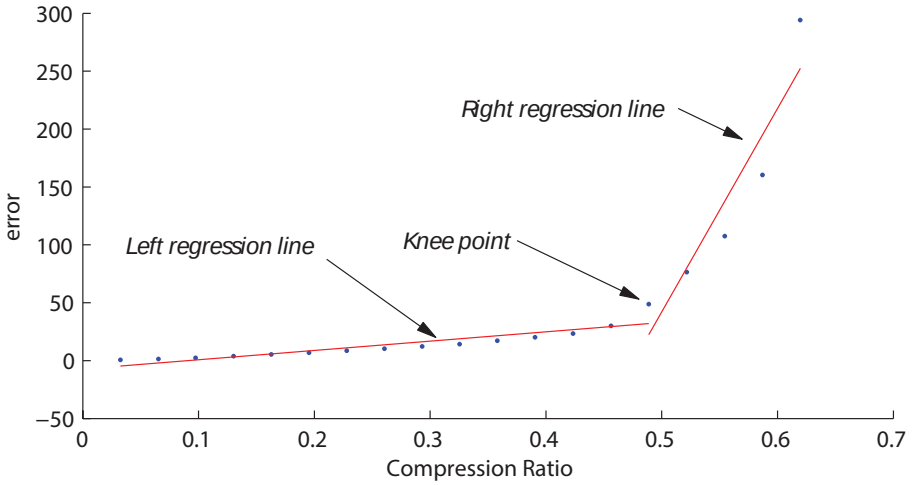

Figure 3.5: The stopping point or *knee point* can be found at the state that has minimum value of summation of regression error between left and right linear regression lines.

Table 3.1 illustrates the main algorithm. The algorithm starts by extracting subsequences from the input sequence by running SUBSEQUENCEEXTRACTOR function. After that, it enters a loop to iteratively select an operator to create clusters until there is no subsequence left. The *Create*, *Add*, and *Merge* operations are applied, then the best one, which gives minimum *error*, is selected in each iteration. Every cluster construction state is kept in a list P for determining the best state later. After breaking the loop, a proper cluster state will be chosen by using STOPPINGSTATE function.

Table 3.1: SSTSC algorithm

Function $[\tilde{C}]$ = SSTSC(T, w, f)	
1.	S = SUBSEQUENCEEXTRACTOR(T, w, f)
2.	while there is an operation left
3.	$[C'[1], S'[1]]$ = CREATECluster($\tilde{C}, S$)
4.	$[C'[2], S'[2]]$ = UPDATECLUSTER($\tilde{C}, S$)
5.	$[C'[3], S'[3]]$ = MERGECLUSTERS($\tilde{C}, S$)
6.	m = ARGMINERROR(C')
7.	$\tilde{C}$ = $C'[m]$
8.	S = $S'[m]$
9.	$P.add(\tilde{C})$
10.	return P.at(STOPPINGSTATE(P))

Details of SUBSEQUENCEEXTRACTOR function are shown in Table 3.2. The function extracts subsequences of length varied from w_{min} to w_{max}, and makes it the same length by using UNIFORMSCALING function. Consequently, the extracted subsequences are normalized by Z-NORMALIZE function, and they are stored in a list of subsequences S.

Table 3.2: Subsequence extractor

Function $[S]$ = SUBSEQUENCEEXTRACTOR(T, w, f)	
1.	$w_{min} = \lceil w/f \rceil$
2.	$w_{max} = \lfloor w \cdot f \rfloor$
3.	l =LENGTH(T)
4.	for $i = w_{min} : w_{max}$
5.	for $j = 1 : l$
6.	t = UNIFORMSCALING($s[j : j + i - 1]$)
7.	Z-NORMALIZE(t)
8.	$t.start$ = j
9.	$t.end$ = $j + i - 1$
10.	$S.add(t)$
11.	return S

Table 3.3 shows CREATECLUSTER function in details. It starts by executing MOTIFDISCOVERY to find a motif pair. After that, a cluster is created from the motif pair, and the motif pair and the subsequences that overlap with them are removed from the list of subsequences S.

Table 3.3: Create operation

Function $[\tilde{C}', S']$ CREATECLUSTER($\tilde{C}$, S)	
1.	$[l_1, l_2]$ = MOTIFDISCOVERY(S)
2.	$C.\bar{C}$ = AVERAGE($S[l_1]$, $S[l_2]$)
3.	$C.addMember(S[l_1])$
4.	$C.addMember(S[l_2])$
5.	$\tilde{C}.add(C)$
6.	remove $S[l_1]$ and $S[l_2]$ and subsequences that overlap $S[l_1]$ and $S[l_2]$ from S
7.	return $\tilde{C}, S$

Table 3.4 explains details of UPDATECLUSTER function. Every cluster center of all created clusters is used as a query sequence for SUBSEQUENCEMATCHING function. The function returns a subsequence from S that is the most similar to the query. The subsequence that produces the least *error* is chosen to be added to the cluster that holds cluster center that was used as the query. The cluster center of that cluster are updated by averaging the old cluster center and the subsequence resulted from the subsequence matching function. After that, the resulted subsequence and its overlapping subsequences are removed from S.

Table 3.4: Add operation

Function $[\tilde{C}', S']$ = UPDATECLUSTER($\tilde{C}$, S)	
1.	for i = 1:$\tilde{C}.numberOfCluster()$
2.	C = $\tilde{C}[i]$
3.	t = SUBSEQUENCEMATCHING(S, $C.\bar{C}$)
4.	$C.\bar{C}$ = AVERAGE($C.\bar{C}$, $S[t]$)
5.	$C.addMember(S[t])$
6.	if $error_{BSF} > C.error()$
7.	C' = C
8.	$i_{BSF} = i$
9.	$error_{BSF}$ = $C.error()$
10.	$t' = t$
11.	$C[i_{BSF}] = C'$
12.	remove $S[t']$ and subsequences that overlap $S[t']$ from S
13.	return $\tilde{C}, S$

The last operation, the *Merge* operation, are described as the MERGECLUSTERS function shown in Table 3.5. All combination pairs of the existing clusters are examined, then a pair that gives minimum *error* will be merged.

To create a cluster center, when performing Create, Add, and merge operations, amplitude averaging approach is used to average two sequences. Given subsequences P =

Table 3.5: Merge operation

Function $[\tilde{C}', S']$ = MERGECLUSTERS($\tilde{C}$, S)	
1.	n = number of clusters in $\tilde{C}$
2.	for $i = 1 : n - 1$
3.	for $j = i + 1 : n$
4.	$C_1 = \tilde{C}[i]$
5.	$C_1 = \tilde{C}[j]$
6.	$C_1.\bar{C}$ = AVERAGE($C_1.\bar{C}$, $C_2.\bar{C}$)
7.	add all members of C_2 to C_1
8.	if $error_{BSF} > C_1.error()$
9.	$C' = C_1$
10.	$l_1 = i$
11.	$l_2 = j$
12.	$\tilde{C}[l_1] = C'$
13.	$\tilde{C}.remove(l_2)$
14.	return $\tilde{C}, S$

$(p_1, ..., p_i, ..., p_n)$ and $Q = (q_1, ..., q_i, ..., q_n)$, a new subsequence $C = (c_1, ..., c_i, ..., c_n)$ is produced by $c_i = \frac{\omega_p p_i + \omega_q q_i}{\omega_p + \omega_q}$, where ω_p and ω_q are weights of P and Q, respectively.

As discussed in the chapter 1, the proposed framework is intentionally designed to be divided into common subtasks, so that it is designed to be manageable for further optimization. The illustrated framework is based on Euclidean distance. However, in some situations (Rodpongpun et al., 2011) patterns or shapes in the stream can be locally warped. In this case, DTW distance can be used to increase the overall effectiveness of the patterns identification. For this reason, it is possible to replace Euclidean distance with DTW distance, such as in error calculation, in subsequence search, or in the motif discovery phase. However, the optimized algorithm for the Euclidean distance in the subsequence search or motif discovery cannot be used because the DTW distance is not a distance metric (Keogh and Ratanamahatana, 2005). This thesis also proposed an optimization for subsequence search algorithm that uses DTW as distance measure. The details is provided in chapter 4.

Moreover, in the same situation when the patterns or shapes in the stream can be locally warped, Euclidean distance will not work very well. Shape-based averaging method (Niennattrakul et al., 2012; Niennattrakul, 2010) can be used instead of amplitude averaging to maintain the characteristics of the patterns. The experiment in section 3.5 will demonstrate the use of the shape-based averaging compared with the amplitude averaging.

3.3.4 Time complexity analysis

Given a time series input T of length m, and a user specific window length w, time complexity of the algorithm using Euclidean distance can be calculated as follows.

For the *Create* operation, suppose a motif discovery algorithm takes $O(M)$ time. Since all subsequences cannot be overlapped, the number of clusters to be created is at most $O(\frac{m}{w})$. Therefore, time taken by all *Create* operations is at most $O(\frac{Mm}{w})$.

For the *Add* operation, time complexity of the subsequence search is $O(mw)$. Each *Add* operation needs to take all cluster centers into account, and the number of clusters is at most $O(\frac{m}{w})$. The maximum number of clustering steps is the same as the number of maximum clusters, which is $O(\frac{m}{w})$. Therefore, the time complexity of all *Add* calculations is at most $O(mw \cdot \frac{m}{w} \cdot \frac{m}{w}) = O(\frac{m^3}{w})$.

For the *Merge* operation, considering a cluster that has been created by merging s clusters so far, there is at most $O(s)$ member subsequences in that cluster. A calculation of *error* in that cluster can be done in $O(ws)$ time. The number of clusters is at most $O(\frac{m}{w})$, so the number of clusters of size s is at most $O(\frac{m}{ws})$. The maximum number of clustering steps is also $O(\frac{m}{w})$. Since number of s can be at most $O(\frac{m}{w})$, the time complexity of all *Marge* operations is at most $O(ws \cdot \left(\frac{m}{ws}\right)^2 \cdot \frac{m}{w}) = O(\frac{m^3}{w^2 s}) = O(\frac{m^2}{w})$.

Therefore, the total time complexity is $O(\frac{Mm}{w} + \frac{m^3}{w} + \frac{m^2}{w}) = O(\frac{Mm}{w} + \frac{m^3}{w})$. Recall that this thesis uses MK motif discovery algorithm (Mueen et al., 2009), whose time complexity is $O(mw)$, so the total time complexity will be $O(\frac{m^3}{w})$.

3.4 Frequent episode discovery from the event sequence

In this section, a detail of frequent episode discovery algorithm will be explained. After applying SSTSC explained in chapter 2, it is straightforward to assign every occurrence of the member in the same cluster to be the same event type, then we have a discrete event sequence. Hence, general frequent episode discovery algorithm can be applied directly. Next is an explanation of how the frequent episode discovery algorithms works.

3.4.1 Frequency counting definitions

As described in section 2.1, there are many episode frequency definitions proposed in the literatures. A study in (Achar et al., 2012) summarizes and proposes a unified view of the apriori-based algorithm for frequent episode discovery, so examples and definitions in this section will mostly recall from (Achar et al., 2012). Recall that the event sequence $D = \langle (E_1, t_1), (E_2, t_2), ..., (E_n, t_n) \rangle$. The example event sequence is:

$$\begin{aligned} \langle (A,1),(A,2),(C,3),(B,3),(A,6),(A,7),(C,8),(B,9),(D,11), \\ (C,12),(A,13),(B,14),(C,15) \rangle \end{aligned} \tag{3.9}$$

Definition 3.4 (Window-based frequency) *(Mannila et al., 1997) A time interval $[t_s, t_e]$ is a window on an event sequence D, where t_s and t_e are integers such that $t_s \leq t_n$ and $t_e \geq t_1$. $(t_e - t_s)$ is the window width of $[t_s, t_e]$. Given a user-defined window width T_x, the window-based frequency f of α, in D is defined as the number of minimal windows of α in D. Note that situations when $t_s < t_1$ or $t_e > t_n$ can be occurred.*

For example, in the event sequence (3.9), there are 5 windows of width 5 that contain an occurrence of $(A \rightarrow B \rightarrow C)$, which are: $[7, 12]$, $[10, 15]$, $[11, 16]$, $[12, 17]$, and $[13, 18]$.

Definition 3.5 (Minimal Occurrence-Based Frequency) *(Mannila et al., 1997) The time-window of an occurrence, h, of α denoted as $[t_{h(v_1)}, t_{h(v_2)}]$. A minimal window of α is a time-window that contains an occurrence of α, such that no proper subwindow of it contains an occurrence of α. A minimal occurrence is an occurrence in a minimal window. The minimal occurrence-based frequency f_{mi} of α in D is defined as the number of minimal windows of α in D.*

For example, in the sequence (3.9), there are 3 minimal windows of an episode $(A \rightarrow B \rightarrow C)$ that are $[2, 8]$, $[7, 12]$, and $[13, 15]$.

Definition 3.6 (Head Frequency) *(Iwanuma et al., 2004) Given a window width k, the head frequency $f_h(\alpha, k)$ of α is the number of windows of width k that contain an occurrence of α starting at the leftmost of the window.*

Definition 3.7 (Total Frequency) *(Iwanuma et al., 2004) Given a window width k, the total*

frequency of α, given by $f_{tot}(\alpha, k)$, is defined as follows.

$$f_{tot}(\alpha, k) = \underset{\beta \preceq \alpha}{min} f_h(\beta, k)$$

Given a window width of 6, the head frequency $f_h(\gamma, 6)$ of $\gamma = (A \rightarrow B \rightarrow C)$ in (3.9) is 4. The total frequency of γ, $f_{tot}(\gamma, k)$, in (3.9) is 3 because the head frequency of $(B \rightarrow C)$ in (3.9) is 3.

Definition 3.8 (Non-Overlapped Frequency) *(Laxman et al., 2005) The non-overlapped frequency $f_{no}$of α in D is given by the cardinality of a maximal non-overlapped set of occurrences of α in D. Two occurrences h_1 and h_2 of α are defined as non-overlapped if either $t_{h_1(v_N)} < t_{h_2(v_1)}$ or $t_{h_2(v_N)} < t_{h_1(v_1)}$. A set of occurrences is defined to be non-overlapped if every pair of occurrences in the set is non-overlapped. A set H of non-overlapped occurrences of α in D is maximal if $|H| \geq |H'|$, where H' is any other set of non-overlapped occurrences of α in D.*

If no event of one occurrence appears in between events of the other, the two occurrences are non-overlapped . The notion of a maximal non-overlapped set is required because there can be many sets of non-overlapped occurrences of an episode with different cardinalities (Laxman, 2006). For example, the non-overlapped frequency of γ in (3.9) is 2. $\langle (A, 2), (B, 3), (C, 8) \rangle$ and $\langle (A, 3), (B, 14), (C, 15) \rangle$ is a maximal set of non-overlapped occurrences.

Definition 3.9 (Non-Interleaved Frequency) *(Laxman, 2006) Two occurrences h_1 and h_2 of α are defined to be non-interleaved if either $t_{h_2(v_j)} \geq t_{h_1(v_{j+1})}, j = 1, 2, ..., N-1$ or $t_{h_1(v_j)} \geq t_{h_2(v_{j+1})}, j = 1, 2, ..., N-1$. If every pair of occurrences in the set H is non-interleaved, a set of occurrences H of α in D is non-interleaved. A set H of non-interleaved occurrences of α in D is maximal if $|H| \geq |H'|$, where H' is any other set of non-interleaved occurrences of α in D. The non-interleaved frequency f_{ni} of α in D is given by the cardinality of a maximal non-interleaved set of occurrences of α in D.*

For example, given an episode $(A \rightarrow B \rightarrow C)$, the occurrences $\langle (A, 2), (B, 3), (C, 8) \rangle$ and $\langle (A, 3), (B, 9), (C, 12) \rangle$are non-interleaved (but overlapped) occurrences in D. Together with $\langle (A, 13), (B, 14), (C, 15) \rangle$, these are occurrences from a set of maximal non-interleaved occurrences of $(A \rightarrow B \rightarrow C)$ in (3.9), hence $f_{ni} = 3$.

Definition 3.10 (Distinct Occurrence-Based Frequency) *(Mahesh Joshi and Kumar, 1999) Given an* N*-node episode α, the h_1 and h_2 are two occurrences of α, and they are defined to be distinct if $h_1(v_i) \neq h_2(v_j) \forall i, j = 1, 2, ..., N$, i.e., the range of their corresponding maps do not intersect. In other words, if two occurrences of an episode do not share any event in the data stream, they are distinct. A set of occurrences is distinct if every pair of occurrences in it is distinct. A set H of distinct occurrences of α in D is maximal if $|H| \geq |H'|$, where H' is any other set of distinct occurrences of α in D. The distinct occurrence-based frequency f_d of α in D is the cardinality of a maximal set of distinct occurrences of α in D.*

The three occurrences that are said to be the maximal non-interleaved occurrences of $(A \rightarrow B \rightarrow C)$ in (3.9) are the same as in the maximal set of distinct occurrences in (3.9).

Table 3.6 shows unified apriori-based algorithm for frequent episode discovery (Achar et al., 2012).

Table 3.6: A unified view of the apriori-based algorithm for frequent episode discovery (Achar et al., 2012)

SERIALEPISODECOUNTER	
Input: Set C_N of *N*-node serial episode, event stream $D = (\Sigma_1, \bar{t}_1), ..., (\Sigma_m, \bar{t}_m)$	
Output: Frequency of episodes in C_N	
1.	for all $\alpha \in C_N$ do
2.	Initialize an automaton of α waiting in the start state.
3.	Initialize frequency of α to ZERO.
4.	for $i = 1$ to m do
5.	for each automaton, $\mathcal{A}$, ready to accept event-type $E \in \Sigma_i$ do
6.	α :=candidate associated with $\mathcal{A}$;
7.	j := state which $\mathcal{A}$ is ready to transit into;
8.	if TRANSIT then
9.	if COPYAUTOMATON then
10.	Add Copy of $\mathcal{A}$ to collection of automata.
11.	Transit $\mathcal{A}$ to state j
12.	if $\exists$an earlier automaton of α already in state j (before $\bar{t}_i$) but not waiting for any $E \in \Sigma_i$ then
13.	if JOIN-AUTOMATON then
14.	Retain $\mathcal{A}$ and retire earlier automaton
15.	if $\mathcal{A}$ reached final state then
16.	Retire $\mathcal{A}$.
17.	if INCREMENT-FREQ then
18.	Increment frequency of α by INC
19.	if RETIRE-AUTOMATON then
20.	Retire all automaton of α and create a state $'0'$ automaton.

There are boolean variables in the Table 3.6 that are TRANSIT, COPY-AUTOMATON,

JOIN-AUTOMATON, INCREMENT-FREQ, and RETIRE-AUTOMATON. Their choices are specified in Tables 3.8, 3.9, (3.10), 3.11, and 3.12, respectively regarding to the frequency counting method listed in the Table 3.7.

Table 3.7: Various frequency counts (Achar et al., 2012)

WB	Windows Based
MO	Minimal occurrences based
MO-X	Minimal occurrence with expiry-time constraints
NO	Non-overlapped
NO-I	Non-overlapped innermost
NO-X	Non-overlapped with expiry-time constraints
NI	Non-interleaved
DO	Distinct occurrences based
AO	All occurrences based
HD	Head frequency

Table 3.8: Conditions for TRANSIT=TRUE (Achar et al., 2012)

WB, MO, MO-X, HD, NO, NO-X, NO-I, AO	Always
NI	If there does not exist an earlier automaton of α already in target state j (before current time $\bar{t}_i$) but not waiting for any $E \in \Sigma_i$
DO	No other earlier earlier automaton for α waiting in same state can transit on an event-type $E \in \Sigma_i$

Table 3.9: Conditions for COPY-AUTOMATON=TRUE (Achar et al., 2012)

WB, MO, MO-X, HD, NI, NO-X, NO-I, DO	Only if $\mathcal{A}$ is in start state
NO	Never
AO	Always

Table 3.10: Conditions for JOIN-AUTOMATON=TRUE (Achar et al., 2012)

WB, MO, MO-X, NO-X, NO-I	Always
DO, AO, HD, NO, NI	Never

Table 3.11: Conditions for INCREMENT-FREQ=TRUE (Achar et al., 2012)

MO, NO, NI, DO, AO, NO-I	Always
WB, NO-X, MO-X, HD	If time difference between first and last state transitions is less than T_X (window width for WB, expiry time for others)

Table 3.12: Conditions for RETIRE-AUTOMATA=TRUE (Achar et al., 2012)

NO, NO-X, NO-I	Always
WB, MO, MO-X, HD, NI, DO, AO, MO-X	Never

Table 3.13: Values taken by INC (Achar et al., 2012)

WB	If (first window which contains current minimal occurrence also contains the previous minimal occurrence), then
	INC = Time diff. between start of last window containing the current minimal occurrence and the start of last window which contains previous minimal occurrence.
	Else
	INC = Time difference between the first and last window containing the current occurrence + 1.

Other counts	INC = 1

3.4.2 Candidate generation

The previous part illustrates the frequency counting of different frequency definitions. This section explain the detail of candidate generation step according to each frequency counting definition. Generally, the candidate generation step exploits a necessary condition that an l-node episode can be frequent if its $(l-1)$-node subepisodes are frequent. In other words, the frequency counting definitions must satisfy anti-monotonicity property that is the frequency of an episode cannot exceed frequency of any of its subepisodes (Achar et al., 2012).

The window-based *(Mannila et al., 1997),* non-overlapped *(Laxman et al., 2005),* and total *(Iwanuma et al., 2004)* frequency countings is known to satisfy the anti-monotonicity property. Also, the distinct occurrence-based frequency can be verified to hold the same (Achar et al., 2012). Consequently, for these frequencies, the l-node episode candidates can be gener-

ated when all of their subepisodes of size $(l-1)$ are frequent. On the other hand, the head, minimal, and non-interleaved frequency countings do not satisfy the anti-monotonicity, which means that subepisodes can be less frequent. However, they can satisfy the anti-monotonicity with some more restricted constraint. Next are some definitions from (Achar et al., 2012) to better explain about the more restricted constraint of anti-monotonicity.

Definition 3.11 (Prefix Subepisode) *(Achar et al., 2012) Given an N-node episode* $\alpha[1] \rightarrow \alpha[2] \rightarrow ... \rightarrow \alpha[N]$*, its K-node prefix subepisode is* $\alpha[1] \rightarrow \alpha[2] \rightarrow ... \rightarrow \alpha[K]$*, for* $K = 1, 2, ..., (N-1)$.

Definition 3.12 (Suffix Subepisode) *(Achar et al., 2012) Given an N-node episode* $\alpha[1] \rightarrow \alpha[2] \rightarrow ... \rightarrow \alpha[N]$*, its (N-K)-node suffix subepisode is* $\alpha[K+1] \rightarrow \alpha[K+2] \rightarrow ... \rightarrow \alpha[N]$*, for* $K = 1, 2, ..., (N-1)$.

Definition 3.13 (Contiguous Subepisode) *(Achar et al., 2012) A K-node subepisode* α_1 *of* α *is a contiguous subepisode if* $\alpha_1 = (\alpha[i] \rightarrow \alpha[i+1] \rightarrow ...\alpha[i+(K-1)]$ *for some* $i = 1, 2, ..., (N-K+1)$.

Under the head frequency, as illustrated in (Iwanuma et al., 2004), only the subepisodes that consist of $\alpha[1]$ are as frequent as α. Therefore, to generate *l*-node candidates, all of their subsequences that have $\alpha[1]$ needs to be frequent. For this reason, the head frequency has some limitations that some $(l-1)$-node suffix subepisodes can be relatively low in frequency. For example, given an event stream of 500 of A's followed by a B and a C, there will be 500 occurrences of an episode $(A \rightarrow B \rightarrow C)$ but only one occurrence of its subepisode $(B \rightarrow C)$. In this case, it will be a problem when frequent episodes have to be causative influences (Achar et al., 2012).

Similar to head frequency, the non-interleaved and the minimal occurrences (windows) occurrences also do not satisfy anti-monotonicity property. However, the anti-monotonicity holds with the $(N-1)$-node suffix and prefix subepisodes. The proof of the anti-monotonicity property of minimal and non-interleaved occurrence-based frequency counting is in (Achar et al., 2012); it is concluded that for the minimal window or non-interleaved frequency, all of contiguous subepisodes have frequency at least equal to the episode. Note that, for the non-contiguous subepisodes, they will have distinct occurrences at least equal to the frequency of the

minimal or non-interleaved occurrences. Consequently, to perform frequent episode discovery under minimal and non-interleaved frequency, an *l*-node candidate can be generated if and only if its prefix and suffix subepisodes of size $(l-1)$ are frequent. The candidate generation step will combine two l-node episodes α and β from the set of frequent serial episodes at level l if the $(l-1)$-node suffix subepisode of α matches the $(l-1)$-node prefix subepisode of β. The result of the combination is an episode $(\alpha[1] \rightarrow \beta[2] \rightarrow ... \rightarrow \alpha[l] \rightarrow \beta[l])$. For example, there are two frequent 3-node episodes: $(B \rightarrow A \rightarrow D)$,and $(A \rightarrow D \rightarrow C)$. At level 4, a 4-node episode candidate $(B \rightarrow A \rightarrow D \rightarrow C)$ can be generated. Candidates at each level will be stored in a lexicographical order, so the episodes having the same $(N-1)$-node prefix subepisode will be together as a block (Mannila et al., 1997). For this reason, extracting their $(N-1)$ suffixes, and then find a block that has a matching of $(N-1)$ prefixes can be done. This type of candidate generation technique is proposed in (Orlando and Foscari, 2004; Patnaik et al., 2008; Srikant and Agrawal, 1996) as mining under gap/inter-event time constraints.

3.5 Experimental results

This section provides experimental results of the proposed framework on various aspects including visualization of the frequent episodes discovered from real-world data, validation of the effectiveness, meaningfulness, and efficiency of the proposed SSTSC algorithm.

3.5.1 Frequent episode discovery using SSTSC

This part will provide experiments of the proposed frequent episode discovery framework on various real-world time series. The objective of this part is to visualize the discovered frequent episodes based on the proposed SSTSC algorithm. The quantitative evaluations will provide later in section 3.5.4.

3.5.1.1 Stock Exchange of Thailand (SET) index data

For a stock index data, knowing how many significant patterns and how they occur is important. However, it is difficult for people to identify those patterns by just looking at a graph. This experiment applies the proposed frequent episode framework to 5-year SET index data (Market Statistics, 2016). The data is recorded at every end of the day except close days from January 1st 2011 to December 31th 2015. Firstly, SSTSC is applied with window length set to 30, so we will get patterns of length 30 days. Scaling factor is set to 1 for simplicity. The results of the SSTSC algorithms is shown with color marked patterns in Figure 3.6 (a).

Secondly, a frequent episode discovery is applied with non-overlapped frequency constraint. The result in Figure 3.6 (b) shows occurrences of a frequent episode consisting of 4 events, which is $\langle A - C - C - B \rangle$ with expiry time constraint set to 365 days. As a result, a user can visualize the frequent patterns in the stock data more easily. Moreover, it is possible to use the result for a prediction purpose. In this case, we get a rule that if an $\langle A - C - C \rangle$ pattern occur, with high possibility, B will occur within 365 days. In this situation, $\langle A - C - C \rangle$ and $\langle A - C - C - B \rangle$ have equal frequency, which is 3, so the confidence level is 1.

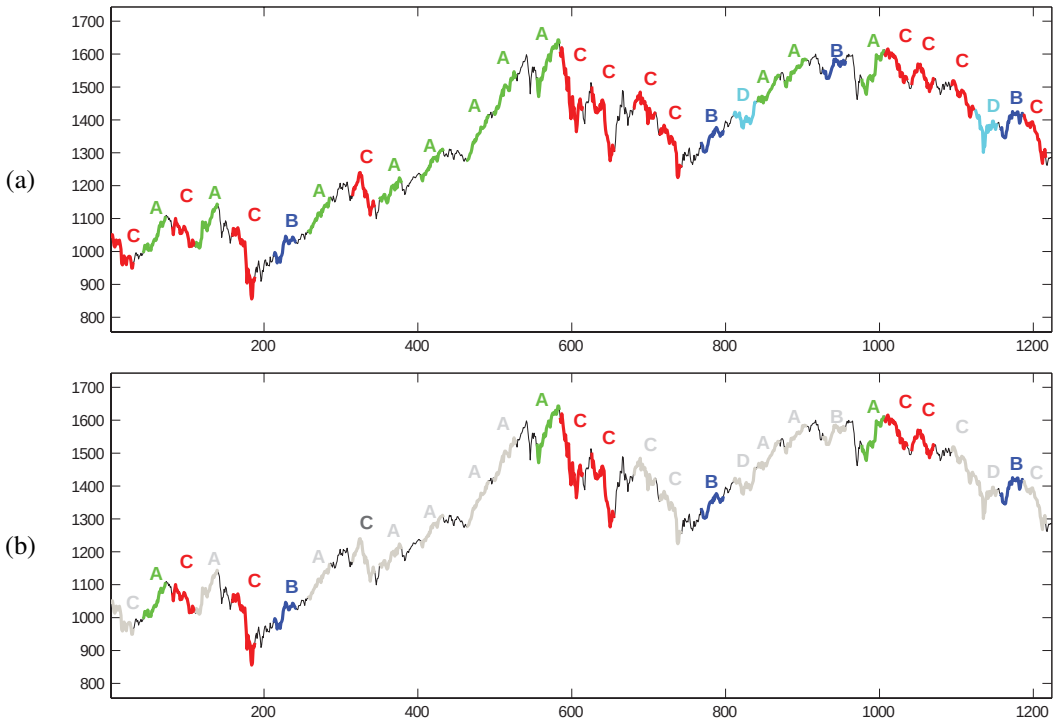

Figure 3.6: (a) SET index data from 2011 to 2015 with frequent patterns marked (output from SSTSC), (b) Occurrences of frequent episodes of size 4.

3.5.1.2 Weather balloon data

In this experiment, the proposed framework is applied on Radiosonde Atmospheric Temperature Products for Assessing Climate (RATPAC) dataset (Free et al., 2005), which is weather data recorded from hydrogen-filled balloons carrying radiosonde up in the air. The data used in this experiment is monthly mean temperature at 850 mb pressure level. The output from SSTSC (window length is set to 12 months) is shown in Figure 3.7 (a), where the patterns of the same group is highlighted with the same color. Figure 3.7 (b) shows frequent episodes of size 2 with non overlapped frequency count, and 365 expiry constraint. The result illustrates possibility of

using the proposed framework for weather analysis and forecasting.

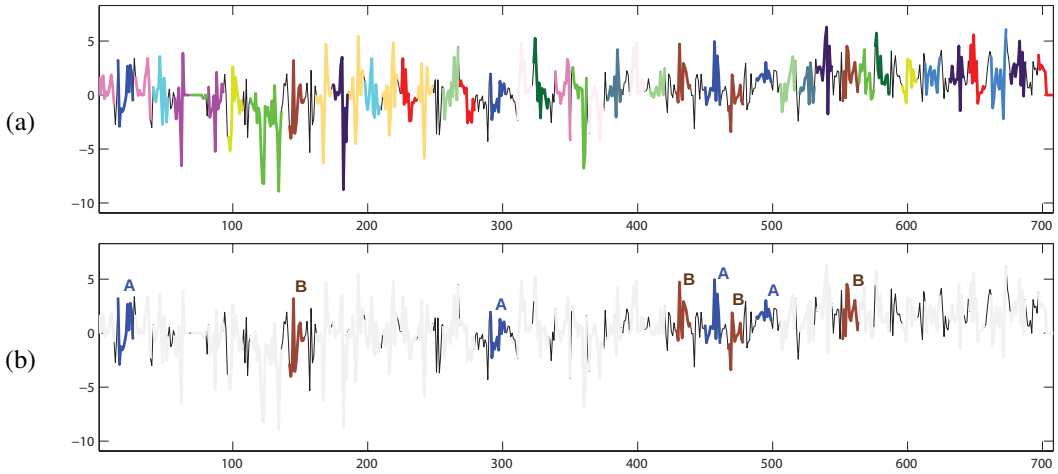

Figure 3.7: (a) Temperature data from a weather balloon with frequent patterns marked (output from SSTSC), (b) Occurrences of frequent episodes of size 2.

3.5.2 Usefulness of SSTSC

This part demonstrates that the proposed SSTSC can be useful by visualizing the clustering results in many types of data domains, i.e., synthetic dataset, data extracted from video surveillance system and images, and real ECG data sequence.

3.5.2.1 Synthetic data

This is an experiment on the *Cylinder-Bell-Funnel (CBF)* dataset from the UCR time series archive (Keogh et al., 2011). It has been shown that most STS clustering algorithms fail to produce meaningful result from this very simple dataset.

This experiment randomly selects data from each class, then concatenates them to a single time series, as shown in Figure 3.8. The cluster results are illustrated as colored subsequences in Figure 3.8. The result shows that the key characteristics of each class are clustered correctly, and the cluster centers can represent the shape of their member subsequences.

3.5.2.2 Video surveillance problem

In this experiment, it will apply the proposed algorithm on the video surveillance domain, which is the gun problem (Keogh et al., 2011). The time series data are captured from the centroid of each actor's right hand performing two actions: *Gun-Draw* and *Point.* The motion

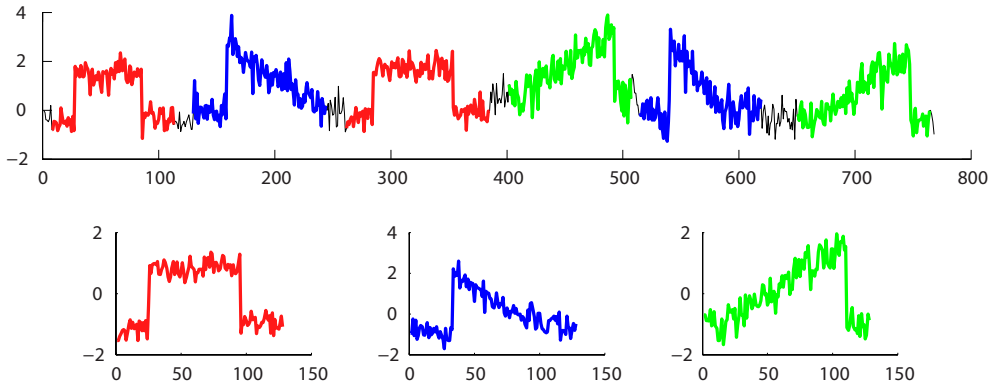

Figure 3.8: *top*) A sequence of CBF dataset. *bottom*) Cluster centers of each class.

of the two classes of action are very similar and difficult to distinguish.

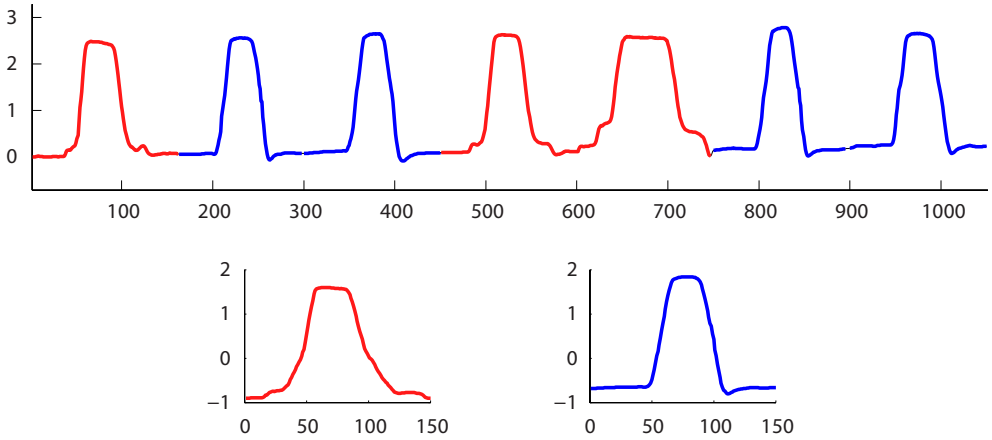

Figure 3.9: *top*) *Gun-Point* data extracted from a video surveillance camera. *bottom*) Cluster centers of each class.

The result of the proposed method, as illustrated in Figure 3.9, shows that all subsequences of motions are clustered correctly to their classes. Furthermore, the cluster centers from the proposed method can preserve the important features and shapes in the data.

3.5.2.3 Time series data extracted from images

This experiment shows the result from clustering data extracted from images, which are created by tracing the local angles from the centroid of an image to its perimeter. The input time series is made by choosing the dataset that has different *complexities* (Batista et al., 2011). The datasets used here are *Face-all* and *OSU-Leaf* (Keogh et al., 2011), which are extracted from human faces with various expressions on the face, and from different species of leaf images.

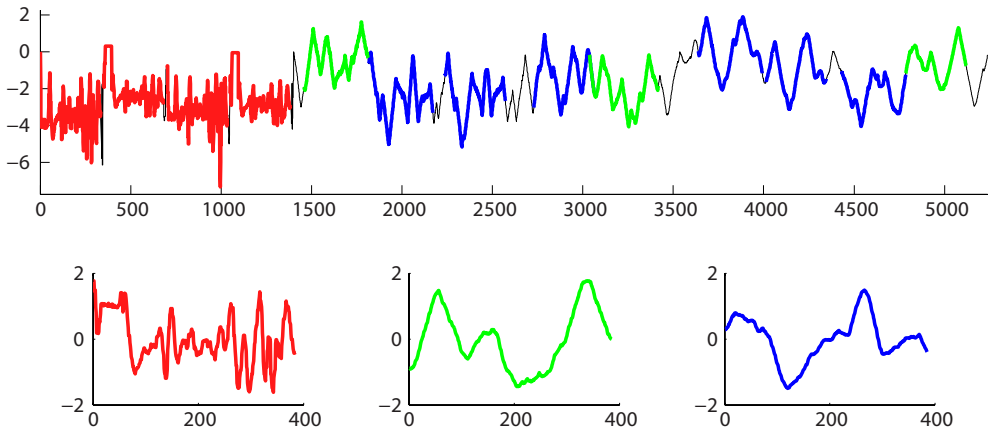

Figure 3.10: *top*) A sequence of data extracted from image of faces and leaves. *bottom*) cluster centers of each class.

Figure 3.10 shows that the proposed algorithm can cluster subsequences of the data even when the data has different *complexity* values. The subsequences of face data are grouped in a cluster, which is shown in red, and the leaf subsequences are separated into two subclasses that have the same shape.

3.5.2.4 ECG data

In this experiment, it is to run the algorithm on a medical dataset, which is an ECG data (Goldberger et al., 2000). Figure 3.11 shows that the beats are of different shapes. If we can separate the beats into clusters, the heart diseases will be diagnosed easier. From the result in Figure 3.11, three groups of heartbeats are clustered. The normal beats are clustered within the same group as shown in green. The abnormal beats, as shown in red, are clustered into the same group. And the blue cluster contains the beats that have minor anomalies, and are clustered separately.

3.5.3 Meaningfulness of SSTSC

As discussed in the beginning that the work from Niennattrakul (Niennattrakul, 2010), called Shape-based Subsequence Time Series Clustering (2STSC), can achieve meaningfulness of STS clustering results. This section will evaluate the proposed SSTSC algorithm comparing with the 2STSC in terms of meaningfulness. To maintain fairness, the datasets used in this experiment are from TSDMA (Niennattrakul, 2010), the same as what were used in (Niennattrakul, 2010). The metric used to evaluate the meaningfulness is also the same as what was used in (Niennattrakul, 2010), which is the Shape-based Meaningfulness Measurement

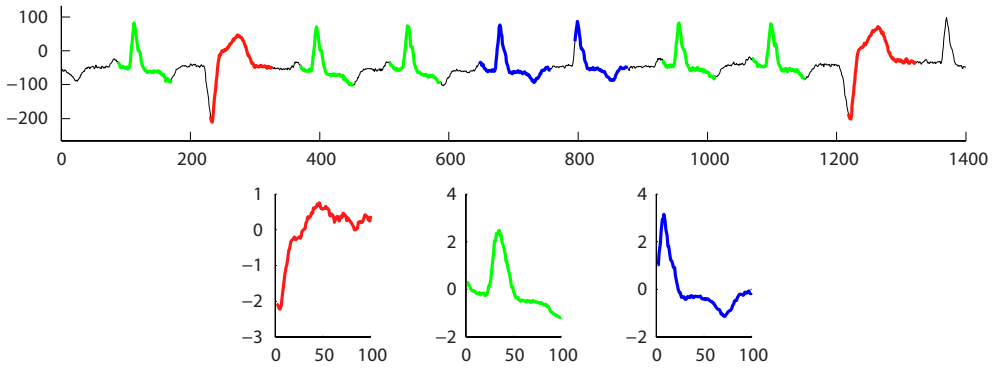

Figure 3.11: *top*) ECG sequences with abnormal heartbeats. *bottom*) Cluster centers from the proposed algorithm.

(SMM). The SMM is introduced by the idea that the clustering results are meaningful when the cluster results can represent subsequences in the input time series. Given an input time series $T = (t_1, t_2, ..., t_n)$ of size n, a set $\mathbb{S} = \{S_1, S_2, ..., S_{n-w+1}\}$ of all subsequences can be extracted by using a sliding window of length w. A set of output $\mathbb{C} = \{C_1, C_2, ..., C_k\}$ of k clusters can be calculated using an SSTSC algorithm where each cluster $C_i = (\mathbb{M}, R)$ consists of a set of cluster members $\mathbb{M} = \{S_j | S_j \in \mathbb{S}\}$ and a cluster representative $R = (r_1, r_2, ..., r_w)$. A set $\mathbb{R} = \{R_1, R_2, ..., R_k\}$ is a set of all cluster representatives. SMM is calculated as a summation of minimum distances from each subsequences to the cluster representatives. The SMM value can be defined as follows.

$$SMM(S, \mathbb{C}) = \frac{|\mathbb{S}| \cdot w}{\sum_{i=1}^{|\mathbb{S}|} min(Distance(S_i, R_j)), \forall R_j \in \mathbb{R}} \tag{3.10}$$

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

The experiment is performed by various STS clustering algorithms including methods proposed in (Niennattrakul, 2010) comparing with the methods proposed in this thesis. The algorithms from (Niennattrakul, 2010) are 2STSC with variations of averaging method: Cubic-Spline Dynamic Time Warping (CDTW) and Iterative Cubic-Spline Dynamic Time Warping (ICDTW), and variations of hierarchical clustering methods: Complete Linkage (CL) and Average Linkage (AL). The algorithms proposed in this thesis are SSTSC with variations of distance measures: Euclidean (EUC) and Dynamic Time warping (DTW) distance, and variations of averaging methods: Amplitude averaging (AA) and Shape-based Averaging (SA). The shape-based averaging method used in this thesis is CDTW.

Variations of parameters, i.e., the number of clusters k and the length of sliding window w are used to illustrate the meaningfulness of the results of all eight STS clustering algorithms. Figures 3.12 and 3.13 illustrate SMMs of Buoy1 and Fortune5004 datasets when the number of clusters k is 3 and the sliding window length w is varied to be 32, 64, and 128. Figures 3.14 and 3.15 show SMMs of Buoy1 and Fortune5004 datasets when the number of clusters k is varied to be 3, 5, and 7, and the sliding window length w is 3.

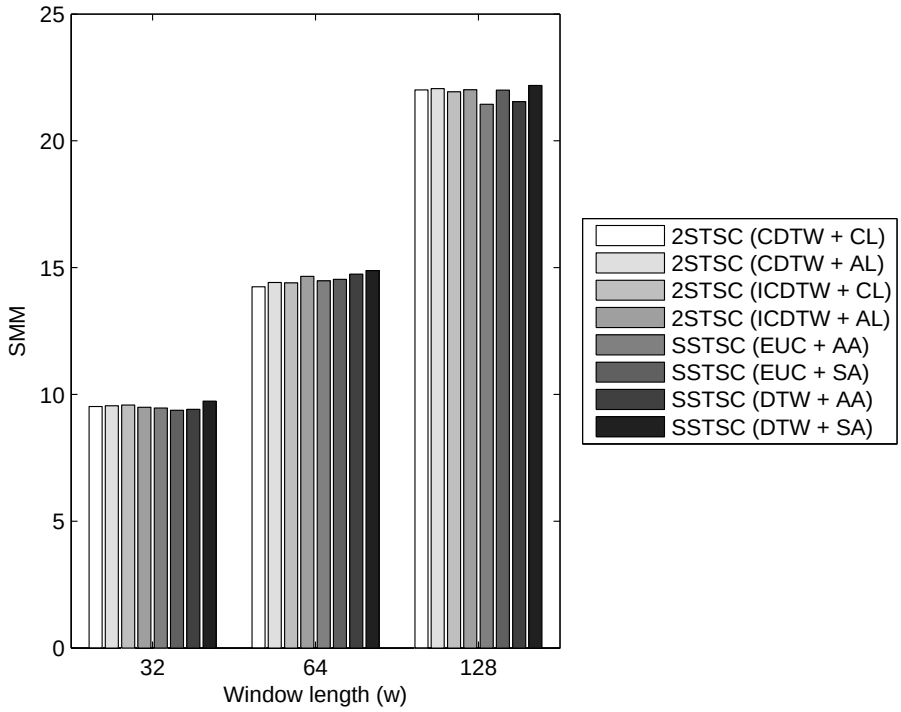

Figure 3.12: SMMs of Buoy1 dataset when number of clusters (k) is 3 and the length of sliding window (w) is varied.

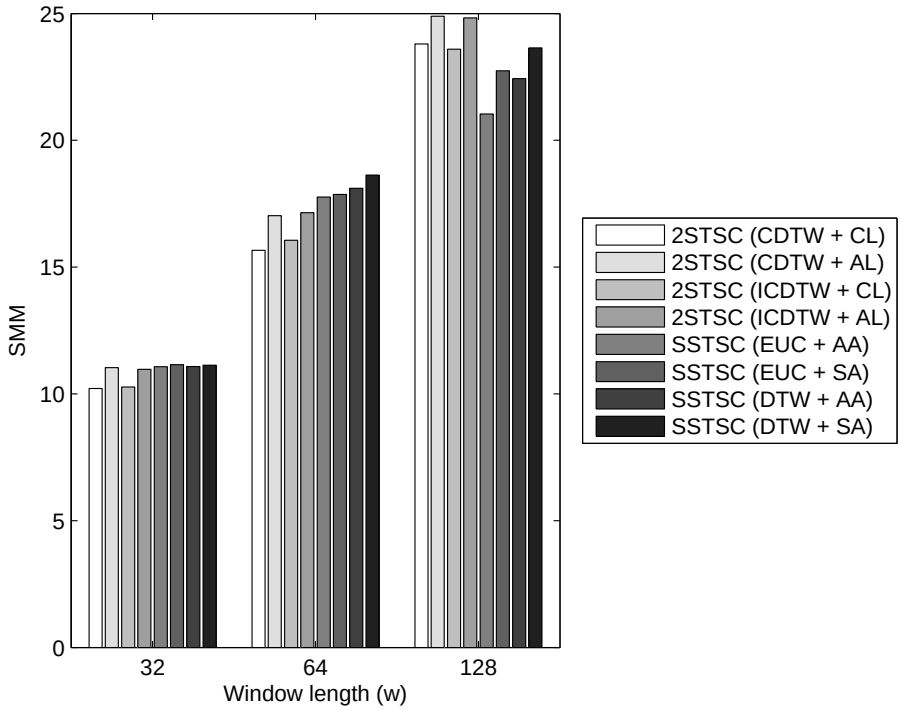

Figure 3.13: SMMs of Fortune5004 dataset when number of clusters (k) is 3 and the length of sliding window (w) is varied.

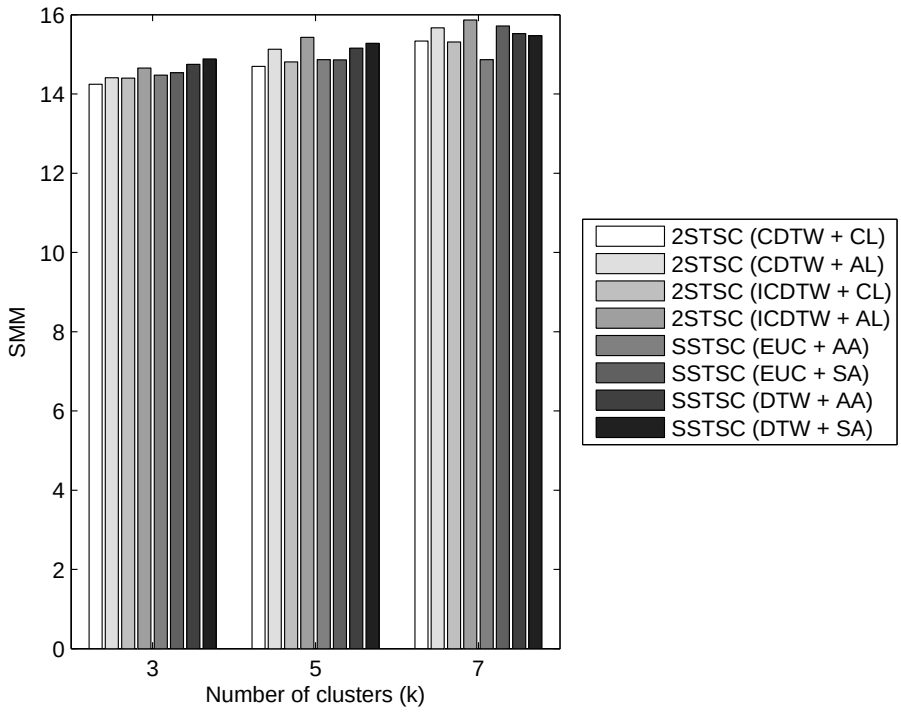

Figure 3.14: SMMs of Buoy1 dataset when number of clusters (k) is varied and the length of sliding window (w) is 64.

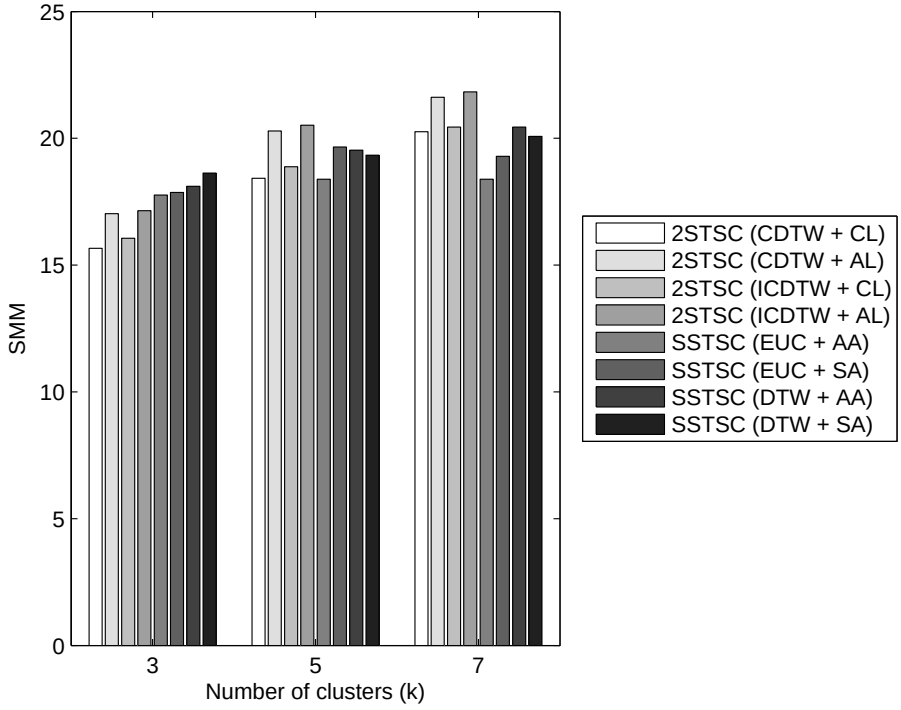

Figure 3.15: SMMs of Fortune5004 dataset when number of clusters (k) is varied and the length of sliding window (w) is 64.

The SMMs results show that despite the SMM measure includes all subsequences, which can be trivial or ineffective to include in clustering results, the algorithms proposed in this thesis can give comparable meaningfulness comparing with algorithms proposed in (Niennattrakul, 2010). The objectives of this thesis is not only achieving meaningfulness of the result, but also the quality of identified patterns without inflation and redundancy. The evaluation for that objective is provided in section 5.4.

For visualization purposes, Figures 3.16, 3.17, 3.18, 3.19, and 3.20 show cluster representatives of Buoy1 dataset with variations of w and k. In the same way, Figures 3.21, 3.22, 3.23, 3.24, and 3.25 show cluster representatives of Fortune5004 dataset with variations of w and k.

Note that datasets details and SMMs values of all datasets are reported in Appendix A.

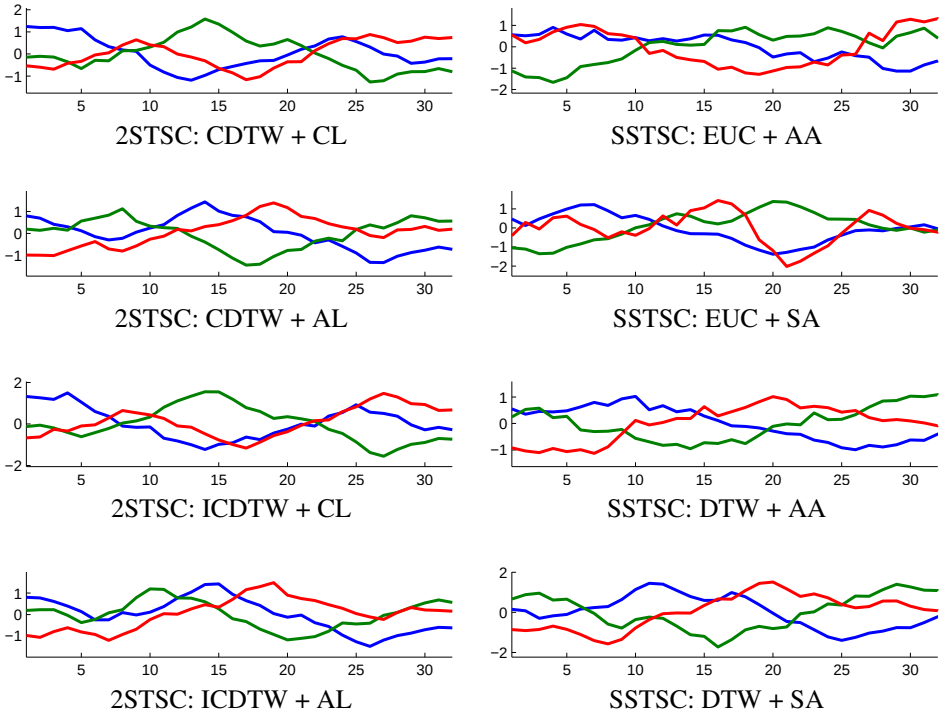

Figure 3.16: Cluster representatives of buoy1 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 32$.

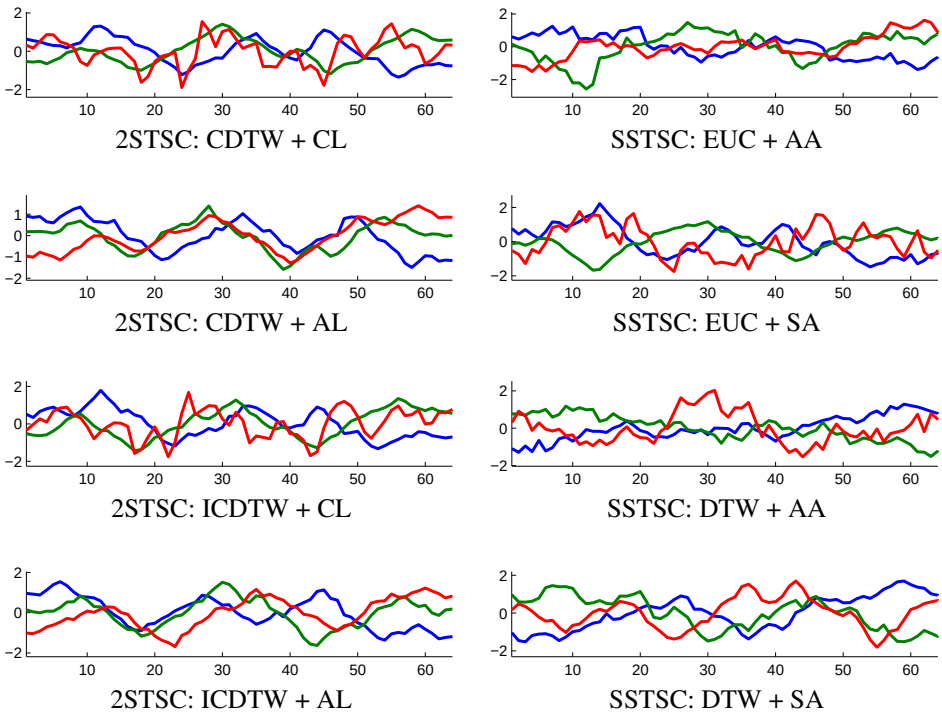

Figure 3.17: Cluster representatives of buoy1 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 64$.

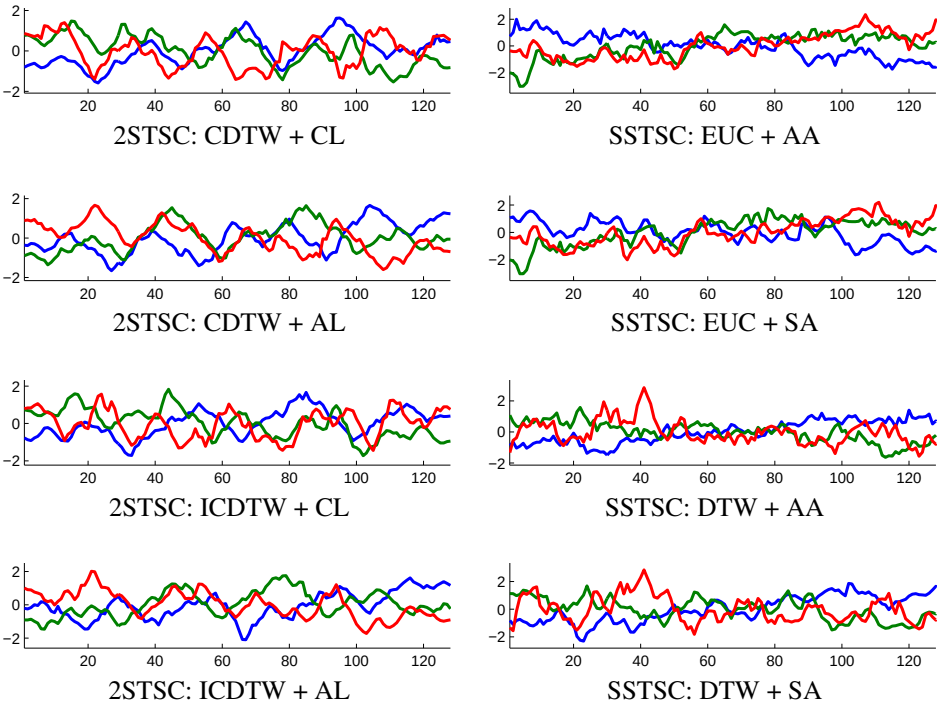

Figure 3.18: Cluster representatives of buoy1 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 128$.

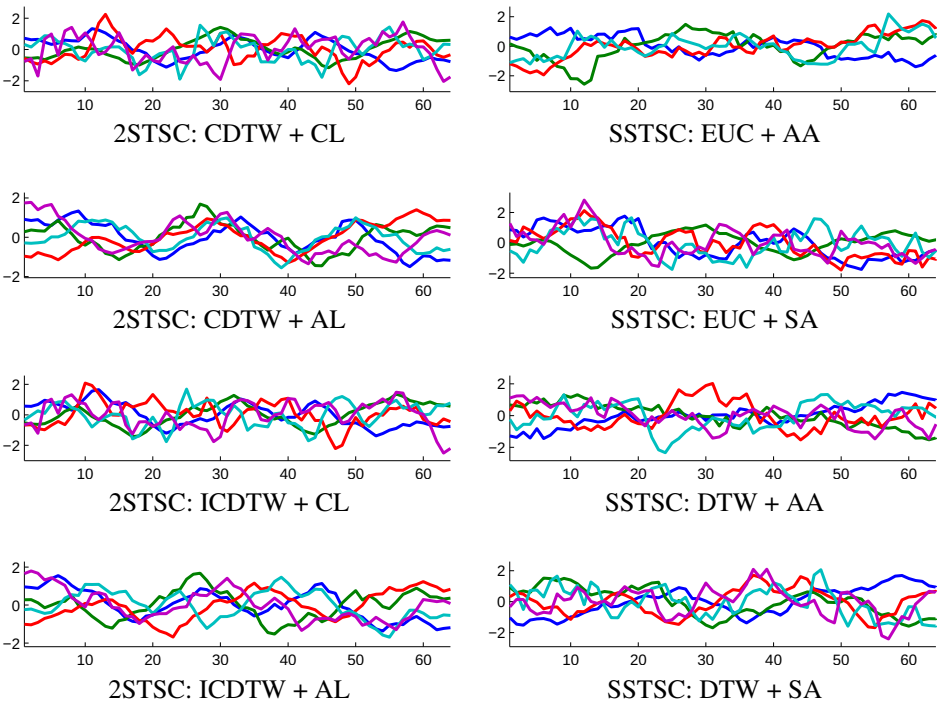

Figure 3.19: Cluster representatives of buoy1 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 5$ and $w = 64$.

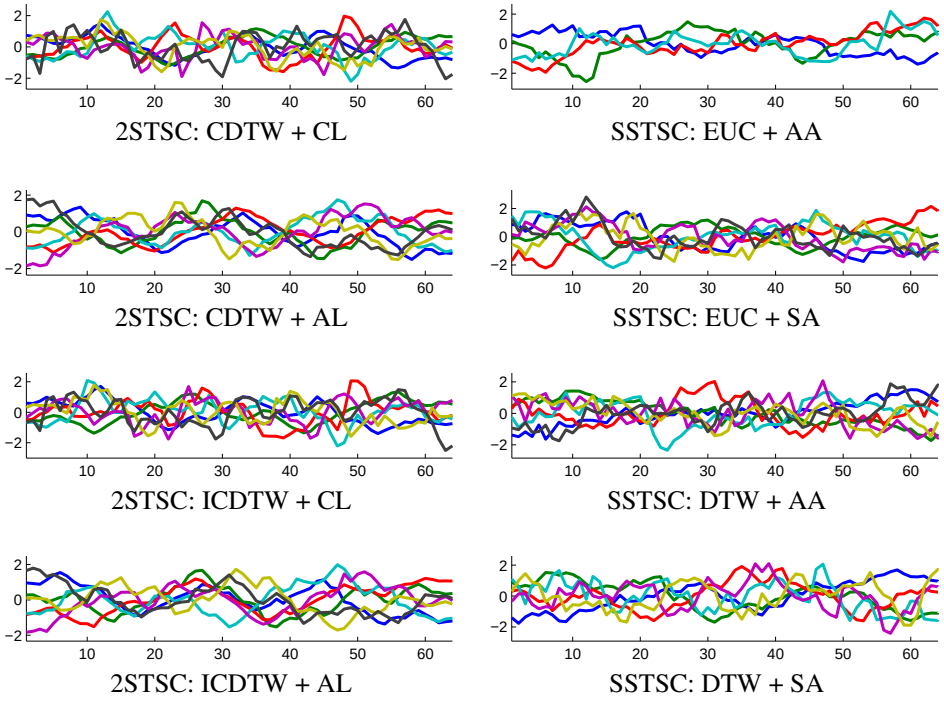

Figure 3.20: Cluster representatives of buoy1 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 7$ and $w = 64$.

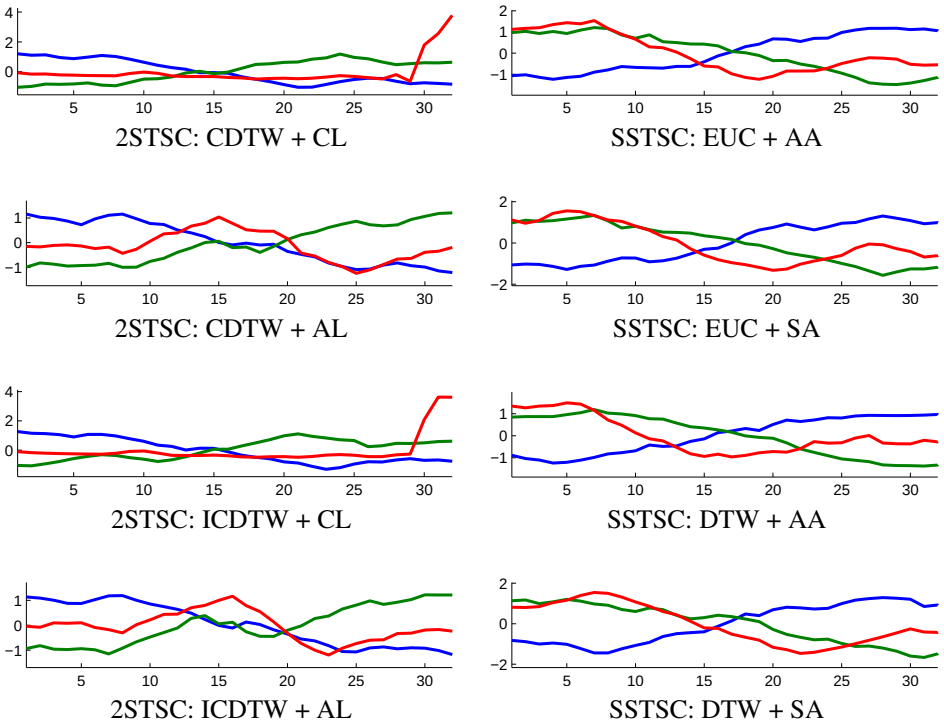

Figure 3.21: Cluster representatives of Fortune5004 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 32$.

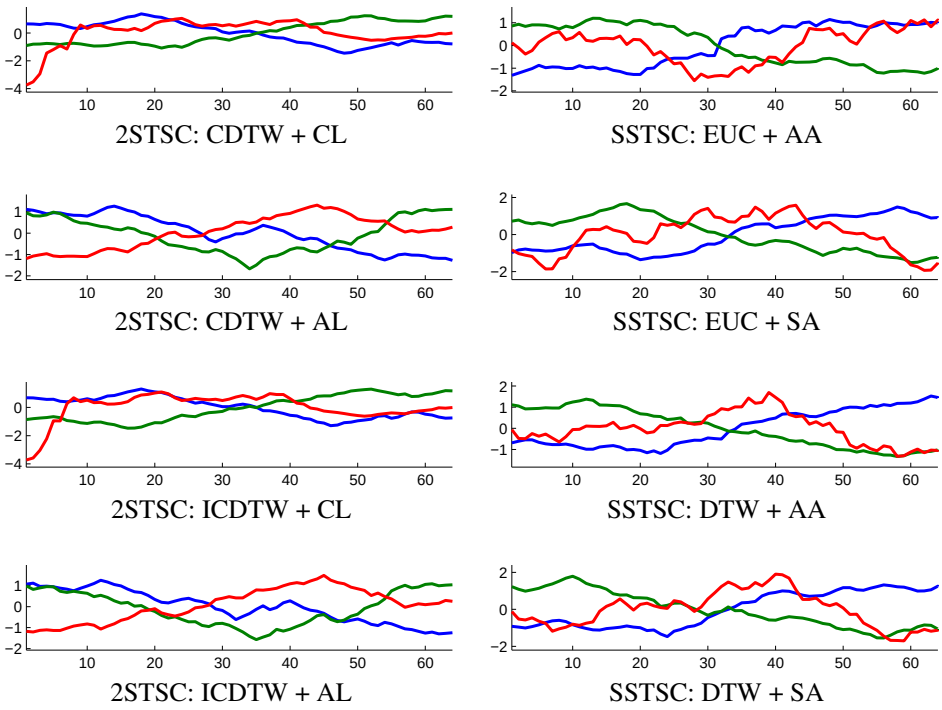

Figure 3.22: Cluster representatives of Fortune5004 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 64$.

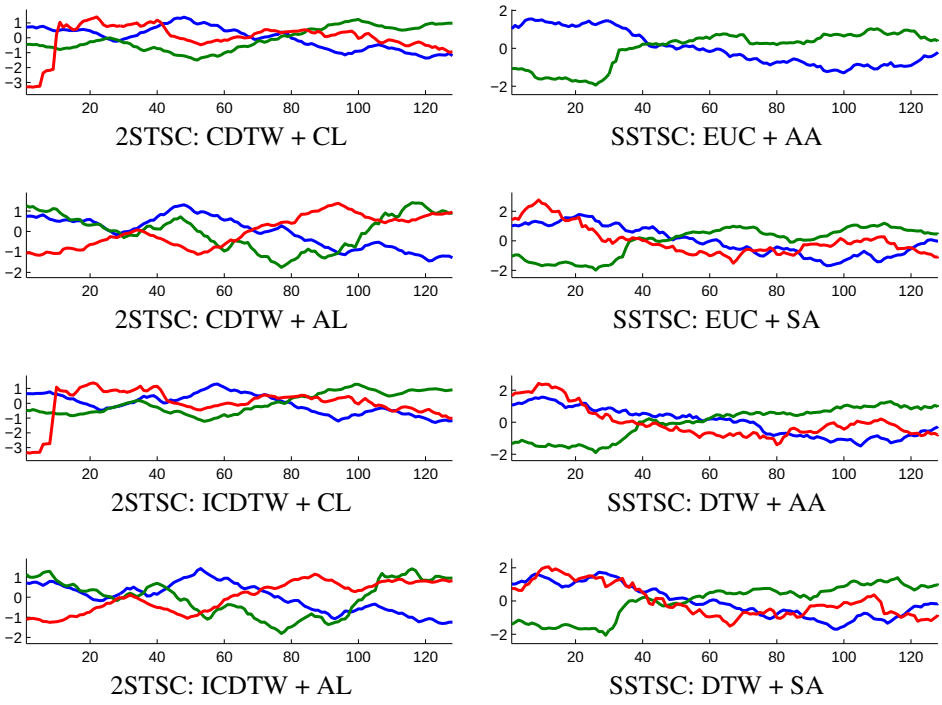

Figure 3.23: Cluster representatives of Fortune5004 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 128$.

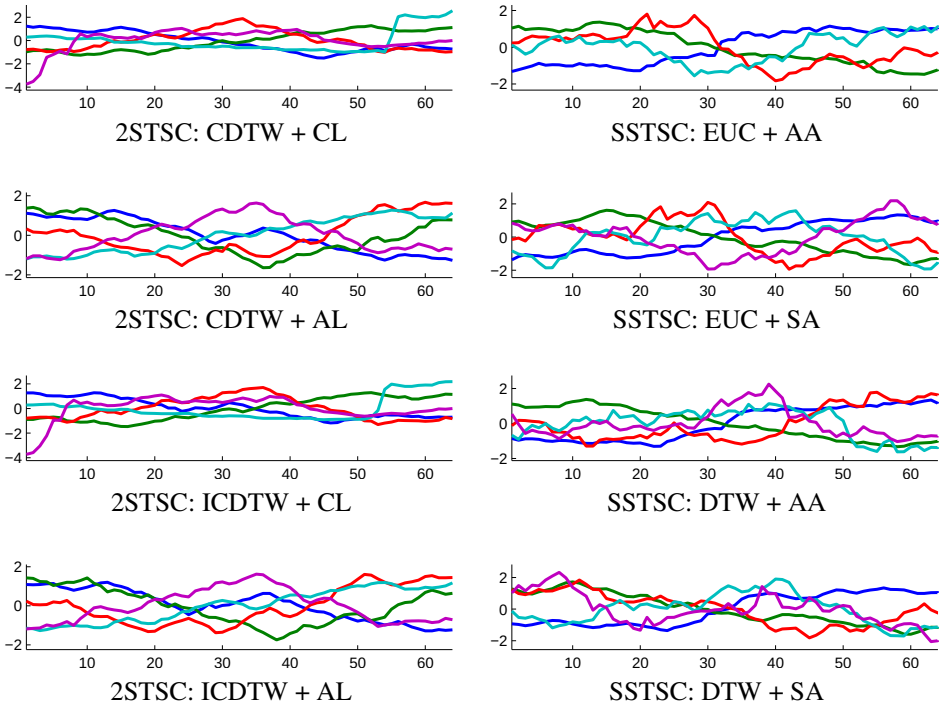

Figure 3.24: Cluster representatives of Fortune5004 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 5$ and $w = 64$.

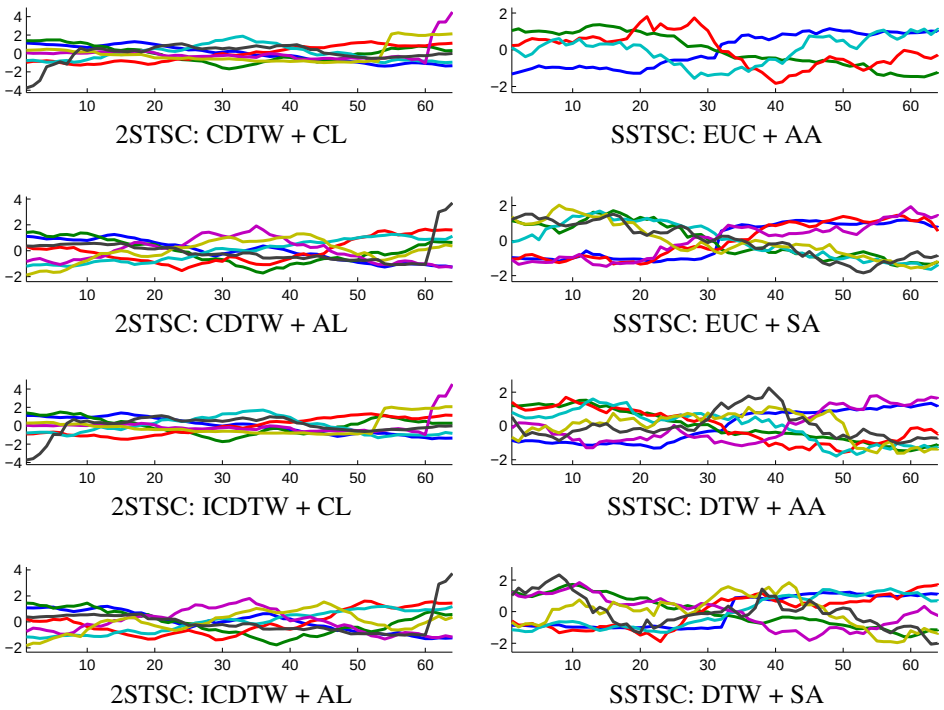

Figure 3.25: Cluster representatives of Fortune5004 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 7$ and $w = 64$.

3.5.4 Effectiveness of SSTSC

The objective of this thesis is to propose an STS clustering algorithm that can effectively identify interesting patterns, so that frequent episode discovery from real-valued time series can be achieved. To evaluate the proposed SSTSC algorithm in terms of effectiveness, it is important to examine on how well the STS clustering can identify interesting patterns, and also discard insignificant ones. For this reason, the datasets are required to be completely annotated, so semi-synthetic stream datasets are introduced to simulate time series that have significant and unimportant patterns together. The stream datasets are made from the UCR Classification/Clustering archive (Keogh et al., 2011) by combining training data sequences and synthesized random walk sequences together. A stream is initialized with a random walk sequence, and then it is repeatedly appended with a training data sequence and a random walk sequence until a specific number of the training data n is reached. The stream finally ends with a random walk sequence. To smooth the stream, before concatenation, each sequence is offset by the last value of the stream. The training data sequences used to construct the stream are randomly selected equally from each class in the training data. A number of training data sequences to be selected for each class is $m = \lceil \frac{20}{c} \rceil$, where c is a number of total classes in the dataset. Therefore, the total number of training data sequences n for all dataset will be at least 20. The length of all random walk sequences is set to $0.5 \cdot w$, where w is the length of the training data sequences. The total number of datasets used in this experiment is 53. The detail of each dataset are provided in Appendix B

Recall that the algorithms proposed in this thesis are SSTSC with variations of distance measures: Euclidean and Dynamic Time Warping (DTW) distance measures, and variations of averaging methods: amplitude averaging and shape-based averaging. In this experiment, other than the *Z*-normalization technique, a level-normalization is added to be a variation of each algorithm to examine the difference between two normalization techniques. Given a subsequence $T_{i,n} = (t_i, t_{i+1}, ..., t_{i+n-1})$ whose mean is μ. The level-normalized time series is $T'_{i,n} = \left(t'_i, t'_{i+1}, ..., t'_{i+n-1}\right)$, where $t'_k = t_k - \mu$. In the experiment, SSTSC algorithms with Euclidean distance and scaling factor $f = 1$ is applied on all of the 53 datasets, while SSTSC algorithms with Euclidean distance and scaling factor $f = 1.2$ is applied on 51 datasets that are subset of all datasets. SSTSC algorithms with DTW distance is applied on 28 datasets that are subset of all datasets. Although, datasets are discarded for some algorithms due to computational resource limitation, the selected datasets still cover various domains of data.

The evaluation metrics used in this section are divided into two groups, and are explained next.

3.5.4.1 Pattern-retrieval-based metrics

Pattern-retrieval-based metrics are used to evaluate how well the STS clustering algorithms can retrieve the annotated patterns regardless of classes or labels of the retrieved patterns. This type assessments is important in terms of how well the algorithms can collect significant patterns without over identification. The following are matrices of this type used in the evaluation in this chapter.

Accuracy on Retrieval (AoR), and Accuracy on Detection (AoD): AoR reflects quality of an algorithm in terms of how well it can collect expected patterns in a data stream; on the other hand, AoD reflects quality of the returned results (Rodpongpun et al., 2011). Given a time series S, a set of expected pattern sequences E, and a set of retrieved sequences R. Firstly, an overlapping subsequence is defined. Let $S[t_s : t_e]$ be the subsequence starting at t_s and ending at t_e. Overlapping subsequence $O_{X,Y}$, where $X = S[a : b]$ and $Y = S[c : d]$, and overlap percentage $P_{X,Y}$ are defined as $O_{X,Y} = S[min\{a, c\} : min\{b, d\}]$ and $P_{X,Y} = \frac{|O_{X,Y}|}{max\{b,d\}-min\{a,c\}+1}$, respectively. Both AoR and AoD can be defined over overlapping subsequence $O_{X,Y}$ and overlapping percentage $P_{X,Y}$ as

$$AoR = \frac{|\{O_{X,Y}|P_{X,Y} > p, X \in R, Y \in E\}|}{|E|} \tag{3.11}$$

$$AoD = \frac{\sum\{P_{X,Y}|P_{X,Y} > p, X \in R, Y \in E\}}{|\{O_{X,Y}|P_{X,Y} > p, X \in R, Y \in E\}|} \tag{3.12}$$

where p is a threshold of $P_{X,Y}$ that defines a sequence in R as a discovered sequence.

Excess Rate (ER) determines the ratio of overly-identified patterns $I_{X,Y}$ over all retrieved subsequences R, where the overly-identified subsequences are the subsequences that has overlapping percentage $P_{X,Y}$ lower than the threshold p. The ER is formally denoted as

$$ER = \frac{|\{I_{X,Y}|P_{X,Y} < p, X \in R, Y \in E\}|}{|R|} \tag{3.13}$$

3.5.4.2 Cluster-accuracy-based metrics

Cluster-accuracy-based metrics are used to evaluate how accurate the STS clustering algorithms can identify types of the patterns comparing with pre-annotated classes. This type of assessments is important in terms of how well the algorithms can distinguish the retrieved patterns.

Rand Index (RI) is a widely used metric for evaluating clustering algorithms (Rand, 1971). It measures similarity of two set of clusters and provide value in range $[0, 1]$. Given a set of n objects $G = \{g_1, g_2, ..., g_n\}$ and suppose that $U = \{u_1, u_1, ..., u_y\}$ is the pre-annotated cluster and $V = \{v_1, v_1, ..., v_z\}$ is a cluster labeled by a clustering algorithm of objects in G such that $\bigcup_{i=1}^{y} u_i = G = \bigcup_{j=1}^{z} v_j$ and $u_i \cap u_i' = \emptyset = v_j \cap v_j'$ for $1 \leq i \neq i' \leq y$ and $1 \leq j \neq j' \leq z$. From a number of all combinations of pairs $\binom{n}{2}$ from the given set, results can be represented in four different types of pairs:

a - objects in a pair that are given the same label in U and the same label in V;

b - objects in a pair that are given the same label in U but the different label in V;

c - objects in a pair that are given the different label in U but the same label in V;

d - objects in a pair that are given the different label in U and the different label in V;

The RI is denoted by $RI(U, V) = \frac{a+d}{a+b+c+d}$.

Precision is denoted by $Precision(U, V) = \frac{a}{a+c}$ (Manning et al., 2008).

Recall is denoted by $Recall(U, V) = \frac{a}{a+b}$ (Manning et al., 2008).

F1-score is denoted by $F1(U, V) = \frac{2 \times Precision \times Recall}{Precision + Recall}$ (Manning et al., 2008).

Note that, for all cluster-accuracy-based metrics, the set of objects to be calculated G contains only objects with overlapping percentage $P_{X,Y}$ higher than the threshold p. The non-retrieved objects are discarded in the calculation because it will affect the value of the number

of agreement (*a* and *d)* and disagreement *(c* and *d)*, so that the value of cluster-accuracy-based metrics can be misinterpreted.

3.5.4.3 Effectiveness evaluation results

This section provides results of each proposed algorithms by varying the overlap threshold p to be 40% and 80%. The 80% overlap threshold is a very extreme criteria in a medical domain (Sivaraks, 2014), while the 40% is more relaxed.

Due to space limitations, in all tables, variations of the proposed SSTSC algorithms are denoted as follows.

E-AA-Z: SSTSC with Euclidean distance, amplitude averaging, and *Z*-normalization.

E-AA-L: SSTSC with Euclidean distance, amplitude averaging, and level-normalization.

E-SA-Z: SSTSC with Euclidean distance, shape-based averaging, and *Z*-normalization.

E-SA-L: SSTSC with Euclidean distance, shape-based averaging, and level-normalization.

D-AA-Z: SSTSC with Euclidean distance, amplitude averaging, and *Z*-normalization.

D-SA-Z: SSTSC with Euclidean distance, shape-based averaging, and *Z*-normalization.

Mean μ, standard deviation σ, minimum and maximum values of each evaluation metric on each dataset when the number of clusters k is set to the number of classes in the dataset, and the scaling factor f of 1 are shown in Figure 3.14. Similarly, Figure 3.15 shows the results when the scaling factor f is set to 1.2. In the same way, mean μ, standard deviation σ, minimum and maximum values of each evaluation metric on each dataset when the number of clusters k is suggested by the SSTSC algorithms, and the scaling factor f of 1 are shown in Figure 3.16, and the scaling factor f of 1.2 are shown in Figure 3.17.

Table 3.14: Summary of all evaluation metrics for each algorithms with given overlap thresholds (p), scaling factor (f) of 1 and the number of clusters (k) is set to the number of classes in the dataset.

Algorithm		E-AA-Z		E-AA-L		E-SA-Z		E-SA-L		D-AA-Z		D-SA-Z	
p		40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
RI	μ	0.60	**0.64**	0.60	0.61	0.58	0.57	0.58	0.58	0.55	0.56	0.59	0.60
	σ	0.12	0.19	0.11	0.15	0.10	**0.20**	0.12	0.12	0.11	0.14	0.12	0.19
	min	0.45	0.33	0.43	0.38	0.45	0.00	0.38	0.39	0.45	0.33	**0.47**	0.17
	max	0.94	**1.00**	0.82	**1.00**	0.82	**1.00**	0.82	0.82	0.90	0.96	0.90	**1.00**
Pre.	μ	0.46	0.49	0.46	0.46	0.44	0.46	0.45	0.46	0.45	0.51	0.49	**0.54**
	σ	0.15	**0.25**	0.12	0.16	0.15	0.21	0.13	0.13	0.12	0.21	0.11	0.18
	min	0.08	0.00	0.19	0.00	0.10	0.00	0.15	0.15	0.15	0.00	**0.30**	0.17
	max	0.89	**1.00**	0.70	**1.00**	0.79	**1.00**	0.79	0.79	0.89	**1.00**	0.89	**1.00**
Rec.	μ	0.55	0.57	0.57	0.57	0.55	0.56	0.61	0.60	0.61	0.59	0.63	**0.68**
	σ	0.17	**0.28**	0.15	0.20	0.17	0.25	0.16	0.17	0.17	0.19	0.15	0.20
	min	0.10	0.00	0.23	0.00	0.13	0.00	0.19	0.19	0.15	0.00	**0.40**	0.31
	max	0.90	**1.00**	0.82	**1.00**	0.82	**1.00**	0.90	**1.00**	0.90	0.90	0.90	**1.00**
F1	μ	0.50	0.52	0.51	0.50	0.49	0.50	0.51	0.51	0.51	0.52	0.54	**0.58**
	σ	0.15	**0.25**	0.13	0.17	0.15	0.22	0.13	0.13	0.12	0.16	0.11	0.17
	min	0.09	0.00	0.21	0.00	0.12	0.00	0.18	0.18	0.15	0.00	**0.34**	0.29
	max	0.90	**1.00**	0.73	**1.00**	0.80	**1.00**	0.80	0.80	0.90	0.90	0.90	**1.00**
AoR	μ	0.88	0.66	0.94	0.78	0.89	0.67	0.95	0.79	0.95	0.76	**0.96**	0.77
	σ	0.17	**0.35**	0.11	0.28	0.15	0.34	0.09	0.26	0.09	0.27	0.07	0.28
	min	0.55	0.05	0.60	0.10	0.50	0.10	0.70	0.20	0.61	0.17	**0.78**	0.19
	max	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
AoD	μ	0.86	0.95	0.90	0.96	0.86	0.95	0.90	0.96	0.89	0.96	0.89	**0.96**
	σ	**0.13**	0.04	0.11	0.03	0.13	0.03	0.10	0.03	0.10	0.03	0.10	0.03
	min	0.62	0.85	0.60	0.86	0.64	0.85	0.65	**0.90**	0.67	0.89	0.66	0.90
	max	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	1.00	1.00	1.00
ER	μ	0.14	0.35	0.07	0.22	0.14	**0.35**	0.07	0.22	0.09	0.26	0.08	0.26
	σ	0.17	**0.36**	0.12	0.30	0.17	0.35	0.12	0.27	0.11	0.28	0.10	0.29
	min	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
	max	0.50	**0.95**	0.45	0.91	0.50	0.91	0.39	0.83	0.42	0.84	0.27	0.83

Table 3.15: Summary of all evaluation metrics for each algorithms with given overlap thresholds (p), scaling factor (f) of 1.2 and the number of clusters (k) is set to the number of classes in the dataset.

Algorithm		E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
p		40%	80%	40%	80%	40%	80%	40%	80%
RI	μ	**0.56**	0.52	0.55	0.55	0.54	0.55	0.53	0.52
	σ	0.10	**0.15**	0.10	0.11	0.08	0.14	0.09	0.12
	min	**0.44**	0.00	0.33	0.28	0.43	0.36	0.37	0.00
	max	0.81	0.83	0.84	0.84	0.74	**1.00**	0.73	0.77
Pre.	μ	0.42	0.43	0.42	0.42	0.41	**0.44**	0.40	0.40
	σ	0.10	**0.18**	0.11	0.13	0.12	0.18	0.11	0.13
	min	**0.21**	0.00	0.14	0.00	0.11	0.13	0.13	0.00
	max	0.58	**1.00**	0.64	0.70	0.68	**1.00**	0.52	0.54
Rec.	μ	0.54	0.53	0.56	0.54	0.54	**0.61**	0.58	0.53
	σ	0.14	**0.24**	0.17	0.20	0.16	0.21	0.18	0.20
	min	**0.28**	0.00	0.17	0.00	0.17	0.19	0.15	0.00
	max	0.82	**1.00**	0.89	**1.00**	0.88	**1.00**	0.90	0.85
F1	μ	0.47	0.45	0.47	0.46	0.46	**0.50**	0.47	0.45
	σ	0.11	0.16	0.12	0.15	0.12	**0.16**	0.12	0.14
	min	**0.26**	0.00	0.15	0.00	0.13	0.17	0.14	0.00
	max	0.64	0.75	0.67	0.80	0.67	**1.00**	0.62	0.65
AoR	μ	0.92	0.66	**0.95**	0.75	0.92	0.67	0.95	0.79
	σ	0.15	**0.30**	0.11	0.25	0.13	0.27	0.12	0.26
	min	**0.45**	0.10	0.40	0.15	0.35	0.15	**0.45**	0.00
	max	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
AoD	μ	0.81	0.90	0.85	0.90	0.82	0.91	0.87	**0.91**
	σ	0.09	0.03	0.07	0.03	0.08	0.03	0.09	**0.13**
	min	0.64	0.82	0.66	**0.85**	0.65	0.85	0.60	0.00
	max	0.97	0.99	0.96	0.99	0.95	1.00	0.98	**1.02**
ER	μ	0.21	**0.42**	0.14	0.31	0.20	0.40	0.11	0.25
	σ	0.18	**0.30**	0.16	0.27	0.17	0.28	0.15	0.28
	min	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
	max	0.64	0.92	0.69	0.88	0.71	0.88	0.61	**1.00**

Table 3.16: Summary of all evaluation metrics for each algorithms with given overlap thresholds (p), scaling factor (f) of 1 and the number of clusters (k) is set automatically by the algorithms.

Algorithm		E-AA-Z		E-AA-L		E-SA-Z		E-SA-L		D-AA-Z		D-SA-Z	
p		40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
RI	μ	0.61	0.62	0.62	0.63	0.61	0.60	0.62	0.63	0.58	0.60	0.60	**0.64**
	σ	0.11	0.19	0.11	0.15	0.11	**0.24**	0.11	0.15	0.09	0.14	0.10	0.18
	min	0.42	0.00	0.45	0.31	0.35	0.00	0.34	0.17	0.45	0.33	**0.48**	0.17
	max	0.84	**1.00**	0.88	**1.00**	0.83	**1.00**	0.84	**1.00**	0.82	0.90	0.86	**1.00**
Pre.	μ	0.50	0.51	0.52	0.51	0.51	0.55	0.54	0.57	0.54	0.61	0.53	**0.62**
	σ	0.19	**0.30**	0.17	0.21	0.19	0.29	0.21	0.25	0.19	0.27	0.18	0.24
	min	0.18	0.00	0.19	0.00	0.13	0.00	0.00	0.00	0.22	0.00	**0.24**	0.17
	max	0.91	**1.00**	0.84	**1.00**	0.96	**1.00**	**1.00**	**1.00**	0.97	**1.00**	**1.00**	**1.00**
Rec.	μ	0.41	0.43	0.41	0.42	0.37	0.45	0.35	0.41	0.42	0.48	0.47	**0.55**
	σ	0.19	0.29	0.18	0.22	0.20	**0.30**	0.22	0.25	0.25	0.24	0.24	0.26
	min	0.06	0.00	0.11	0.00	0.07	0.00	0.00	0.00	0.06	0.00	0.12	**0.14**
	max	**1.00**	**1.00**	0.84	**1.00**	0.84	**1.00**	0.90	**1.00**	0.84	**1.00**	**1.00**	**1.00**
F1	μ	0.41	0.43	0.43	0.44	0.38	0.46	0.38	0.43	0.40	0.48	0.45	**0.53**
	σ	0.15	0.25	0.14	0.18	0.15	**0.26**	0.17	0.20	0.15	0.17	0.14	0.20
	min	0.09	0.00	0.16	0.00	0.12	0.00	0.00	0.00	0.10	0.00	**0.18**	**0.18**
	max	0.72	**1.00**	0.73	**1.00**	0.72	**1.00**	0.78	**1.00**	0.61	0.86	0.67	**1.00**
AoR	μ	0.88	0.65	0.93	0.76	0.87	0.63	0.91	0.75	0.92	0.72	**0.93**	0.73
	σ	0.15	0.35	0.11	0.30	0.17	**0.37**	0.12	0.30	0.09	0.28	0.07	0.27
	min	0.50	0.05	0.60	0.10	0.40	0.00	0.60	0.05	0.61	0.17	**0.75**	0.19
	max	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
AoD	μ	0.85	0.95	0.89	0.96	0.84	0.95	0.89	0.96	0.88	0.96	0.89	**0.96**
	σ	0.14	0.04	0.12	0.04	**0.14**	0.04	0.12	0.03	0.11	0.03	0.11	0.03
	min	0.63	0.85	0.60	0.86	0.61	0.84	0.63	**0.89**	0.66	0.89	0.66	0.89
	max	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00	1.00	1.00	1.00
ER	μ	0.14	0.36	0.07	0.23	0.14	**0.38**	0.08	0.24	0.09	0.27	0.08	0.27
	σ	0.17	0.36	0.12	0.31	0.18	**0.38**	0.13	0.31	0.11	0.30	0.10	0.31
	min	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
	max	0.52	0.95	0.45	0.91	0.50	**1.00**	0.43	0.95	0.42	0.84	0.29	0.83

Table 3.17: Summary of all evaluation metrics for each algorithms with given overlap thresholds (p), scaling factor (f) of 1.2 and the number of clusters (k) is set automatically by the algorithms.

Algorithm		E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
p		40%	80%	40%	80%	40%	80%	40%	80%
RI	μ	**0.62**	0.57	0.61	0.60	0.61	0.59	0.60	0.59
	σ	0.11	**0.24**	0.10	0.16	0.11	0.20	0.10	0.17
	min	0.33	0.00	**0.45**	0.00	0.40	0.00	0.44	0.00
	max	0.84	**1.00**	0.88	**1.00**	0.82	**1.00**	0.82	**1.00**
Pre.	μ	**0.54**	0.52	0.49	0.50	0.53	0.51	0.50	0.51
	σ	0.20	**0.31**	0.21	0.27	0.22	0.30	0.22	0.30
	min	**0.19**	0.00	0.11	0.00	0.15	0.00	0.13	0.00
	max	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Rec.	μ	0.27	0.33	0.32	**0.37**	0.24	0.31	0.24	0.28
	σ	0.12	0.24	0.20	**0.27**	0.15	0.23	0.14	0.20
	min	**0.09**	0.00	0.03	0.00	0.04	0.00	0.03	0.00
	max	0.70	**1.00**	0.90	**1.00**	0.70	**1.00**	0.63	**1.00**
F1	μ	0.34	0.37	0.35	**0.37**	0.30	0.35	0.29	0.32
	σ	0.11	**0.22**	0.16	0.20	0.13	0.22	0.12	0.18
	min	**0.13**	0.00	0.04	0.00	0.06	0.00	0.06	0.00
	max	0.53	**1.00**	0.75	**1.00**	0.60	**1.00**	0.55	**1.00**
AoR	μ	0.88	0.60	**0.93**	0.70	0.88	0.62	0.92	0.74
	σ	0.21	**0.34**	0.13	0.29	0.18	0.31	0.14	0.30
	min	0.15	0.00	**0.40**	0.05	0.29	0.10	**0.40**	0.00
	max	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
AoD	μ	0.80	0.89	0.82	0.89	0.80	0.89	0.86	**0.91**
	σ	0.09	0.03	0.09	0.03	0.09	0.03	0.10	**0.13**
	min	0.62	0.82	0.64	0.83	0.61	**0.84**	0.60	0.00
	max	0.94	0.98	0.94	0.95	0.94	0.97	**0.98**	**0.98**
ER	μ	0.20	**0.44**	0.11	0.32	0.20	0.43	0.10	0.27
	σ	0.23	**0.35**	0.16	0.31	0.21	0.32	0.16	0.31
	min	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
	max	0.81	**1.00**	0.69	0.95	0.74	0.91	0.64	**1.00**

Visualization of the results and the values of all evaluation metrics on each dataset with a scaling factor f of 1 are provided in Appendix B, and with a scaling factor f of 1.2 are provided in Appendix C.

The results show that the proposed algorithms can identify most of the planted patterns, and also be able to distinguish the class of the retrieved patterns. Moreover, table 3.15 and table 3.17 demonstrate that the proposed algorithms are able to handle variable size of patterns with a flexible scaling factor parameter f, while maintaining effective results. The algorithms with DTW distance and shape-based averaging can perform well in the datasets that have a lot of locally warping patterns in the same class. However, the algorithms that use DTW as a distance measure consume a lot of computational time. An optimization to a subsequence matching

subtask that use DTW is proposed in chapter 4.

Equally important, the proposed algorithms also provide suggestion of number of clusters k, and the results of the algorithms that use the suggested parameter k show comparable effectiveness comparing with that when k is manually fixed to be the same as number of planted classes. Table 3.18 provides number of clusters k chosen at the *knee point* of compression ratio - error line from each dataset by each proposed algorithm. The bottom of the table shows exact match percentage and mean of absolute difference for each algorithm. The results show that the numbers of exact matches of suggested number of cluster k to the number of planted classes are acceptable considering that the algorithms have to cluster so many subsequences mixed with important and trivial ones, and also have to decide a reasonable point where the clusters of patterns are proper.

Table 3.18: Number of cluster (k) chosen at the *knee point* of compression ratio-error line from each dataset by each proposed algorithm. At the bottom of the table shows exact match percentage and mean of absolute difference for each algorithm.

Dataset	k in constructed datasets	k chosen by each Algorithm									
		f = 1						f = 1.2			
		E-AA-Z	E-AA-L	E-SA-Z	E-SA-L	D-AA-Z	D-SA-Z	E-AA-Z	E-AA-L	E-SA-Z	E-SA-L
ItalyPowerDemand	2	3	4	3	5	2	2	6	4	6	6
SonyAIBORobotSurfaceII	2	5	5	8	7	10	8	9	8	9	10
SonyAIBORobotSurface	2	6	3	6	3	7	3	7	5	10	3
DistalPhalanxOutlineCorrect	2	4	2	2	2	2	3	5	5	5	5
MiddlePhalanxOutlineCorrect	2	2	2	2	2	4	3	6	5	4	5
PhalangesOutlinesCorrect	2	4	4	4	5	2	2	4	4	4	5
ProximalPhalanxOutlineCorrect	2	2	3	2	2	2	2	5	4	5	5
DistalPhalanxOutlineAgeGroup	3	3	3	3	3	3	4	5	6	4	5
MiddlePhalanxOutlineAgeGroup	3	3	3	3	3	5	5	3	4	3	3
ProximalPhalanxOutlineAgeGroup	3	2	2	2	2	2	2	5	4	4	4
TwoLeadECG	2	3	3	4	3	2	8	5	4	5	6
MoteStrain	2	5	5	7	7	5	5	9	5	8	7
ECG200	2	4	4	3	4	6	6	5	4	8	3
CBF	3	3	4	7	9	6	6	7	5	10	9
Two_Patterns	4	2	4	4	8	10	7	9	7	11	10
ECGFiveDays	2	4	4	4	4	10	3	4	8	4	8
ECG5000	5	6	6	7	4	7	5	6	5	8	5
Gun_Point	2	5	5	5	5	3	3	4	4	4	5
wafer	2	7	6	8	8	8	6	4	6	4	8
ChlorineConcentration	3	6	5	6	2	6	6	7	7	8	9
Wine	2	2	2	2	2	3	3	4	7	6	6
Strawberry	2	5	5	6	6	2	3	4	4	2	3
ArrowHead	3	2	3	3	5	4	3	5	5	8	5
Trace	4	5	5	6	5	2	2	4	3	4	6
ToeSegmentation1	2	7	6	9	8	10	6	7	8	9	9
Coffee	2	3	3	4	4	6	5	4	5	6	4
ToeSegmentation2	2	6	5	7	7	9	6	8	6	10	10
FaceFour	4	5	6	7	8	2	4	7	5	11	9
yoga	2	5	4	6	5	-	-	7	6	9	7
Ham	2	4	2	4	6	-	-	7	6	9	9
Meat	3	2	3	3	4	-	-	6	4	6	4
Beef	5	4	4	3	3	-	-	6	5	6	7
FordA	2	4	4	10	10	-	-	6	4	11	11
FordB	2	2	5	8	10	-	-	5	8	10	10
ShapeletSim	2	2	3	6	10	-	-	2	3	2	2
BeetleFly	2	5	7	7	10	-	-	7	9	8	11
BirdChicken	2	7	6	7	7	-	-	9	8	10	8
Earthquakes	2	2	2	2	4	-	-	2	3	2	11
Herring	2	3	2	3	3	-	-	5	4	5	4
OliveOil	4	4	4	4	4	-	-	7	6	6	5
Car	4	4	4	5	5	-	-	5	4	5	5
Lighting2	2	2	2	2	8	-	-	2	5	8	7
Computers	2	6	4	7	6	-	-	9	6	8	6
LargeKitchenAppliances	3	8	5	9	7	-	-	11	5	12	4
RefrigerationDevices	3	2	3	5	10	-	-	5	5	13	13
ScreenType	3	2	5	2	2	-	-	5	6	6	9
SmallKitchenAppliances	3	3	6	9	6	-	-	9	4	9	7
WormsTwoClass	2	5	5	2	11	-	-	5	8	2	11
Worms	5	3	4	2	6	-	-	6	6	9	10
StarLightCurves	3	5	5	6	5	-	-	6	6	5	5
Haptics	5	6	4	7	5	-	-	7	4	7	4
CinC_ECG_torso	4	8	6	10	9	-	-	-	-	-	-
HandOutlines	2	4	4	4	4	-	-	-	-	-	-
Exact match percentage (%)		24.53	28.30	24.53	15.09	25.00	21.43	9.62	5.77	11.54	5.77
Mean of absolute difference		1.77	1.53	2.62	3.08	2.82	2.00	3.16	2.75	4.16	4.10

3.5.5 Comparison of SSTSC with the brute-force method

This section will demonstrate the performance on searching through the search space. Figure 3.26 illustrates the result of the proposed algorithm comparing with the brute-force method tested on the ECG data used in section 3.5.2.4. It contains roughly 6000 possible paths of the input time series of length 1400 data points and a window size of 100. The result

of the proposed method is shown in a thick blue line. The main goal is to maintain *error* while maximizing the *compression ratio*. As shown in Figure 3.26, the proposed algorithm can search through the search space closely following the optimal path by examining just 1 out of 6000 possible paths.

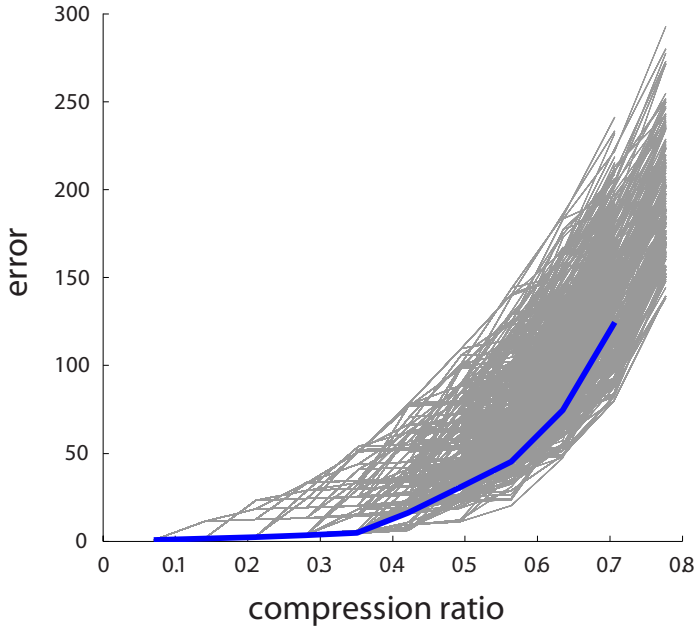

Figure 3.26: The proposed algorithm (shown in blue) comparing with the brute-force method.

3.6 Conclusion

In this chapter, a frequent episode discovery framework for real-valued time series, which is the main objective of this thesis, is proposed. The main concept of the framework is to identify interesting patterns in a real-valued time series sequence, so that the frequent episode discovery algorithms can be applied. A new Subsequence Time Series (STS) clustering named Selective Subsequence Time Series Clustering (SSTSC) is introduced to transform real-valued patterns to sets of events based on their shapes. The proposed SSTSC is designed to identify significant patterns while discarding trivial ones to minimize pattern inflations and redundancies. More importantly, the proposed method can maintain the meaningfulness of the clustering results, which means the clustering outputs truly represent characteristics of the time series input. The proposed algorithm also allows different lengths of member subsequences. As a result, the introduced framework assures the efficiency and effectiveness of the proposed algorithm by experimenting on various data domains. Additionally, the proposed method can perform clustering by requiring only a few parameters where users can easily and flexibly adjust.

CHAPTER IV

EFFICIENT SUBSEQUENCE SEARCH ON STREAMING DATA BASED ON TIME WARPING DISTANCE

Due to the age of data explosion, analysis of data stream in real time is crucial in many data mining tasks including classification, clustering, anomaly detection, and pattern discovery. Commonly, these tasks require a subsequence matching algorithm as an important subroutine. Recently, SPRING (Sakurai et al., 2007), a breakthrough subsequence matching algorithm for data stream under Dynamic Time Warping (DTW) distance (Ratanamahatana and Keogh, 2005) has been proposed. SPRING can report an optimal subsequence in linear time. More specifically, it incrementally updates DTW distance, for each new streaming data point, only in time complexity of the query sequence's length. After the proposal of SPRING, many authors (Athitsos et al., 2008; Niennattrakul et al., 2009; Peng et al., 2008) have introduced fast algorithms to improve performance of subsequence matching. This thesis claims that all of those past research works (Athitsos et al., 2008; Niennattrakul et al., 2009; Peng et al., 2008; Sakurai et al., 2007) are meaningless because the query sequence and candidate sequences from the data stream were not normalized. Normalization (Han et al., 2006) is essential to achieve accurate and meaningful distance calculation, as it normalizes the data to have similar offset and distribution, regardless of the distance measure used, especially for DTW distance measure.

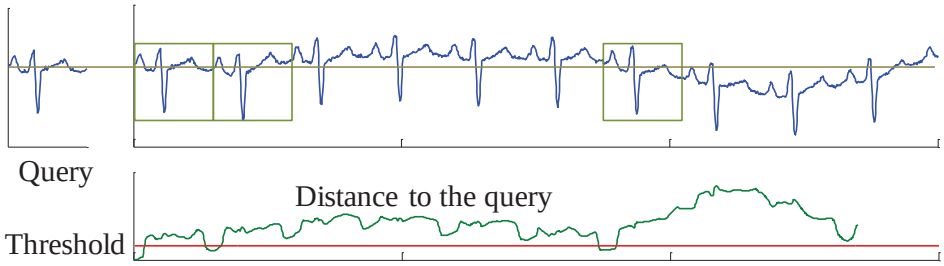

Figure 4.1: Subsequence search without normalization in ECG data. Many subsequences with similar shape to the query are left undetected.

Unfortunately, as mentioned above, current subsequence matching algorithms concern mostly about speed enhancement, but neither on accuracy nor meaningfulness. Figure 4.1 illus-

trates subsequence searching in ECG data (Goldberger et al., 2000). Many subsequences with similar shape to the query are missed by the search without normalization.

However, there is an effort to resolve this problem by trying other approaches; the latest one devises some hardware (Sart et al., 2010) to accelerate the computation time. The authors propose two techniques, i.e., GPUs and FPGAs, to speed up subsequence matching using DTW with normalization. They have shown that GPUs and FPGAs can help speed up the search significantly. However, it is not practical in real world problems; implementation is hardware dependent, and some systems are not flexibly adjusted to the problem.

This thesis introduces a novel subsequence matching algorithm called MSM (Meaningful Subsequence Matching) for data stream under DTW distance. MSM consists of two new ideas. First, this thesis introduces a multi-resolution lower bound, LB_GUN (Lower-Bounding distance function under Global constraint, Uniform scaling, and Normalization) combining with the well-known LB_Keogh (Keogh and Ratanamahatana, 2005) lower-bounding function. LB_GUN is a new lower-bounding distance function extended from LB_Keogh. Second, SSM (Scaling Subsequence Matrix) is used for lower-bounding distance estimation of LB_GUN by incrementally estimating value of normalized data point while guaranteeing no false dismissals. The distances for every scaled query sequence are stored in SSM, and then MSM algorithm monitors SSM to report the optimal range query or the optimal top-k query when a new streaming data point is received. From these two ideas, MSM can monitor data stream nearly in linear time, and it also achieves much higher accuracy than existing algorithms as expected. The remainder of this chapter is organized as follows. Problem definitions are provided in Section 4.1. MSM, the proposed method, is described in Section 4.2. Experimental results are reported in Section 4.3, and the proposed work is concluded in Section 4.4.

4.1 Problem definition

This thesis focuses on two main query problems on streaming time series data, i.e., optimal range query and optimal top-k query. The objective of the optimal range query is to find non-overlapping normalized subsequences from a data stream, whose distance between a candidate sequence and a query sequence must be less than a threshold , where the query sequence is scaled and normalized under uniform scaling between scaling range$[f_{min}, f_{max}]$. On the other hand, optimal top-k query reports top-*k* non-overlapping normalized subsequences. Nevertheless, the scaled query sequences and all candidate subsequences in the data stream must

be normalized in order to return meaningful results. A naive method to monitor incoming data stream first initializes a set of normalized scaled query sequences, and then candidate sequences are extracted from the data stream using sliding-window model. After normalization, distance calculation is performed on the extracted subsequences and non-overlapping optimal results are reported (if any). However, this naive method requires as high as $O(n^3)$ time complexity for each new incoming streaming data point.

4.2 Proposed method

Since the naive method consumes too high time complexity, this thesis proposes a novel approach for subsequence matching which gives meaningful result. The proposed method is called MSM algorithm (Meaningful Subsequence Matching), which contains two new ideas, i.e., a multi-resolution lower-bounding function *LB_GUN* (Lower-Bounding function under Global constraint, Uniform scaling, and Normalization), and SSM (Scaling Subsequence Matrix) which incrementally estimates value of *LB_GUN* under global constraint, uniform scaling, and normalization in linear time while guaranteeing no false dismissals. Three following subsections of *LB_GUN*, SSM, and MSM algorithm are precisely described.

4.2.1 Lower-bounding distance under Global constraint, Uniform Scaling, and Normalization (LB_GUN)

LB_GUN is a lower-bounding function of DTW distance extended from *LB_Keogh* (Keogh and Ratanamahatana, 2005) whose distance calculation can be done in linear time. Before calculation, *LB_GUN* first creates an envelope E' from scaled and normalized envelopes. More specifically, three sequence sets are generated, i.e., sets of $\tilde{Q}$, $\tilde{R}$, and $\tilde{E}$. The scaled query set $\tilde{Q} = \{Q'_{n_{min}}, \ldots, Q'_k, \ldots, Q'_{n_{max}}\}$ is first generated by scaling and normalizing a query sequence Q to every normalized scaled query sequence R'_k, and the scaled global constraint $\tilde{R} = \{R'_{n_{min}}, \ldots, R'_k, \ldots, R'_{n_{max}}\}$ set is derived from scaling a specific global constraint set $\tilde{R}$ with all possible scaling lengths from n_{min} to n_{max}. An envelope E_k of a normalized scaled query sequence Q'_k and a scaled global constraint R'_k for sequence length k is created as in *LB_Keogh*, and is stored in the envelope set $\tilde{E} = \{E_{n_{min}}, \ldots, E_k, \ldots, E_{n_{max}}\}$. Then, E' is generated by merging all envelopes in the set $\tilde{E}$ together, where $E' = \left\{\langle u'_1, l'_1\rangle, \ldots, \langle u'_i, l'_i\rangle, \ldots, \left\langle u'_{n_{max}}, l'_{n_{max}}\right\rangle\right\}$. To find lower-bounding distance between a query sequence Q and a candidate sequence C under global constraint, uniform scaling, and normalization, an envelope E' of a query sequence Q is generated as mention

above. $LB_{GUN}(Q, C, n)$ is shown in Equation 4.1.

$$LB_{GUN}(Q, C, n) = \frac{1}{\sigma_{C_{1\ldots n}}} \left(\sum_{i=1}^{n} \alpha_i + \mu_{C_{1\ldots n}} \sum_{i=1}^{n} \beta_i \right) + \sum_{i=1}^{n} \gamma_i \tag{4.1}$$

$$\alpha_i = \begin{cases} c_i & ; c'_i \geqslant u'_i \\ -c_i & ; c'_i \leqslant l'_i \\ 0 & ; otherwise \end{cases} \tag{4.2}$$

$$\beta_i = \begin{cases} -1 & ; c'_i \geqslant u'_i \\ 1 & ; c'_i \leqslant l'_i \\ 0 & ; otherwise \end{cases} \tag{4.3}$$

$$\gamma_i = \begin{cases} -u'_i & ; c'_i \geqslant u'_i \\ l_i & ; c'_i \leqslant l'_i \\ 0 & ; otherwise \end{cases} \tag{4.4}$$

where $\mu_{C_{1\ldots n}}$ and $\sigma_{C_{1\ldots n}}$ are arithmetic mean and standard deviation of data points 1 to n of a candidate sequence C, $c'_i = (c_i - \mu_{C_{1\ldots i}})/\sigma_{C_{1\ldots i}}$, n_{min} and n_{max} are desired scaling lengths, and $n_{min} \leqslant n \leqslant n_{max}$.

4.2.2 Scaling Subsequence Matrix

SSM (Scaling Subsequence Matrix) is another important component in *MSM* algorithm. It stores lower-bounding distances determined by *LB_GUN* for each new incoming streaming data point s_t at time t from data stream S. Suppose we have a query sequence Q; each element $\langle t, j \rangle$ of the matrix contains five values, i.e., v_t, j, w_t, j,x_t, j,y_t, j, and z_t, j, calculated from time $t - j$ to time t. Therefore, values in matrix element $\langle t, j \rangle$ can be incrementally updated from the matrix element $\langle t - 1, j - 1 \rangle$ according to the following equations.

$$v_{t,j} = v_{t-1,j-1} + \begin{cases} s_t & ; s'_t \geqslant u'_j \\ -s_t & ; s'_t \leqslant l'_j \\ 0 & ; otherwise \end{cases} \tag{4.5}$$

$$w_{t,j} = w_{t-1,j-1} + \begin{cases} -1 & ; s'_t \geqslant u'_j \\ 1 & ; s'_t \leqslant l'_j \\ 0 & ; otherwise \end{cases} \tag{4.6}$$

$$x_{t,j} = x_{t-1,j-1} + \begin{cases} -u'_j & ; s'_t \geqslant u'_j \\ l'_j & ; s'_t \leqslant l'_j \\ 0 & ; otherwise \end{cases} \tag{4.7}$$

$$y_{t-j} = y_{t-1,j-1} + s_t \tag{4.8}$$

$$z_{t,j} = z_{t-1,j-1} + (s_t)^2 \tag{4.9}$$

$$lb_{t,j} = \frac{1}{\sigma_{t,j}} (v_{t,j} + \mu_{t,j} \cdot w_{t,j}) + x_{t,j} \tag{4.10}$$

where $s'_t = \frac{s_t - \mu_{t,j}}{\sigma_{t,j}}$, $\mu_{t,j} = \frac{y_{t,j}}{j}$, $\sigma_{t,j} = \sqrt{\frac{z_{t,j}}{j} - (\mu_{t,j})^2}$, u_j and l_j are from an enveloped E' generated from a query sequence Q, $1 \leqslant j \leqslant n_{max}$, $n_{min} \leqslant j \leqslant n_{max}$, and $lb_{t,j}$ is a lower-bounding distance *LB_GUN* for an element $\langle t, j \rangle$.

4.2.3 Meaningful Subsequence Matching

Since SSM is updated at every arrival of new streaming data point s_t, the proposed MSM algorithm can monitor both optimal range query and optimal top-*k* query. More specifically, for optimal range query, MSM first calculates and updates values including lower-bounding distances in SSM, which is an estimation of *LB_GUN* and then checks whether a best-so-far distance d_{best} is smaller than a threshold ε. If so, MSM reports an optimal subsequence when there is no overlapping subsequence, and MSM resets d_{best} and values in SSM. For all lb_{tj} which are smaller than d_{best} in a range from n_{min} to n_{max}, *LB_GUN* and *LB_Keogh* are calculated and compared to d_{best} to prune off the DTW distance calculation. If they are not pruned by any lower-bounding distances, DTW distance is computed to update d_{best} and the optimal subsequence's position. Additionally, MSM uses only two columns of SSM that are

values in time t and values in time $t-1$. All lower-bounding distances and DTW distance are normalized by dividing by i. The MSM algorithm for optimal range query is described in Table 4.1.

Table 4.1: MSM algorithm for optimal range query

MSMOPTIMALRANGEQUERY	
Input: a new streaming data point s_t	
Output: an optimal subsequence (if any)	
1.	update v_i, w_i, x_i, y_i and z_i for all i, $1 \leq i \leq n_{max}$ and lb_i for all i, $n_{min} \leq I \leq n_{max}$
2.	if $(d_{best} < \varepsilon \; and \; \forall i, t_{best}^{end} \leq t-i)$
3.	Report$(d_{best}, S[t_{best}^{end}, t_{best}^{start}])$
4.	$d_{best} = \infty$
5.	reset v_i, w_i, x_i, y_i and $z_i = \infty$ for all i, $t_{best}^{end} > t$–i
6.	for $(i = n_{min} \; to \; n_{max})$
7.	if $(lb_i \leq d_{best})$
8.	if $(LB_{GUN}(Q_i', Normalize(S[t–i : t])) < d_{best}$
9.	if $(LB_{Keogh}(Q_i', Normalize(S[t–i : t])) < d_{best}$
10.	$distance = DTW(Q_i', Normalize(S[t–i : t]))$
11.	if $(distance \leq d_{best})$
12.	$d_{best} = distance; t_{best}^{end} = t–i; t_{best}^{start} = t$
13.	substitute v_i', w_i', x_i', y_i' and z_i' for v_i, w_i, x_i, y_i, and z_i

MSM algorithm for optimal top-*k* query is implemented based on the optimal range query. With a priority queue, MSM stores the *k*-best non-overlapping subsequence with DTW distance from the result of MSMOPTIMALRANGEQUERY. First, it initializes a threshold ε to positive infinity. Then, for every new streaming data point s_t, the queue is updated, and the threshold ε is set to the largest DTW distance in the queue. The MSM algorithm for optimal top-*k* query is described in Table 4.2.

Table 4.2: MSM algorithm for optimal top-*k* query

MSMOPTIMALTOPKQUERY	
Input: a new streaming data point s_t	
Output: update set P of top-*k* subsequence	
1.	$[C, d_C]$= MSMOPTIMALRANGEQUERY(s_t, ε)
2.	If$(C \neq NULL)$
3.	$P.push(C, d_C)$
4.	if $(size(P) > k)$
5.	$P.pop()$
6.	$\varepsilon = P.peek().d_C$

4.3 Experimental results

Since none of the current subsequence matching algorithms under DTW distance can handle the changes of data distribution, offset, and scaling, this section will compare the proposed method with naive approach in terms of computational time only since the proposed method and the naive method will both achieve the same accuracy. On the other hand, this section compare the proposed method's accuracy with SPRING, the best existing subsequence matching under DTW distance. Note that this section does not compare the running time with that of SPRING; while SPRING will have smaller running time, its results are inaccurate due to lack of normalization, therefore not a reasonable comparison.

Streaming datasets are generated by combining training data sequences from the UCR classification/clustering datasets (Keogh et al., 2011) and synthesized random walk sequences. A stream is initialized with a random walk sequence, and then a training data sequence is appended to the stream. To smooth the stream, before concatenation, each sequence is offset by the last value of the stream. The dataset to be used in the experiments are Aidac, Beef, CBF, Coffee, ECG200, Gun Point, Lighting7, Olive Oil, Trace and Synthetic Control which are represented by Data 1, Data 2, Data 3, Data 4, Data 5, Data 6, Data 7, Data 8, Data 9, and Data 10, respectively

In the first experiment, it will compare the MSM algorithm with naive method in terms of computational cost by measuring the number of distance calculations. Figure 4.2 shows the numbers of all distance calculations by varying global constraints to 2, 4, 6, 8 and 10 respectively, and in Figure 4.3, scaling range $[f_{min}, f_{max}]$ are varied from [0.8, 1.2], [0.85, 1.15], [0.9, 1.1], and [0.95, 1.05], respectively. The numbers of all distance calculations are normalized to 100% which represent numbers of DTW calculations used in the naive method. As expected, MSM is much faster than the naive method by a large margin. Additionally, in MSM, the proposed multi-resolution lower-bounding function is efficiently used to filter out several candidate sequences in linear time while guaranteeing no false dismissals; therefore, MSM algorithm requires only a small number of DTW distance calculations comparing with the naive method.

Then, this section compares the MSM algorithm with SPRING to measure performance in terms of accuracy, both Accuracy-on-Retrieval (AoR) and Accuracy-on-Detection (AoD). AoR reflects quality of an algorithm that is able to find the patterns in a data stream; on the

other hand, AoD reflects quality of the returned results. Suppose we have data stream S, a set of expected pattern sequences E, and a set of retrieved sequences R. We first define an overlapping subsequence. Let $S[t_s : t_e]$ be the subsequence starting at t_s and ending at t_e. Overlapping subsequence $O_{X,Y}$, where $X = S[a : b]$ and $Y = S[c : d]$, and overlap percentage $P_{X,Y}$ are defined as $O_{X,Y} = S[min\{a, c\} : min\{b, d\}]$, and $P_{X,Y} = \frac{|O_{X,Y}|}{max\{b,d\}-min\{a,c\}+1}$, respectively. Both AoR and AoD can be defined over overlapping subsequence $O_{X,Y}$ and overlapping percentage $P_{X,Y}$ as $AoR = \frac{|\{O_{X,Y}|P_{X,Y}>p,X\in R,Y\in E\}|}{|E|}$ and $AoD = \frac{\sum\{P_{X,Y}|P_{X,Y}>p,X\in R,Y\in E\}}{|\{O_{X,Y}|P_{X,Y}>p,X\in R,Y\in E\}|}$, respectively, where p is a threshold of $P_{X,Y}$ that defines a sequence in R as a discovered sequence. Figure 4.4 and Figure 4.5 compare AoRs of MSM and SPRING under various scaling ranges and global constraints, respectively. Figure 4.6 and Figure 4.7 illustrate AoDs on every scaling range and global constraint, respectively. The results show that MSM produces more meaningful result since SPRING does not support global constraint (illustrated as one single column of 100% global constraint in Figure 4.5 and Figure 4.7), uniform scaling, nor normalization.

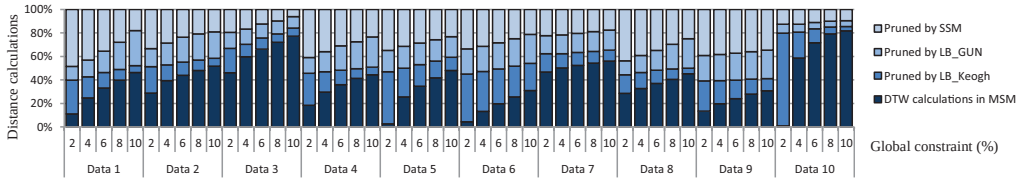

Figure 4.2: DTW distance calculations filtered out by MSM with varying global constraints.

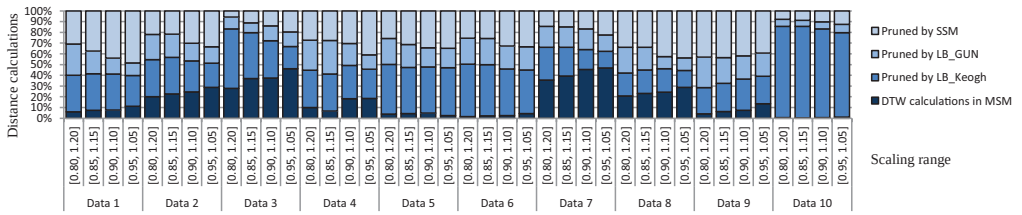

Figure 4.3: DTW distance calculations filtered out by MSM with varying scaling ranges.

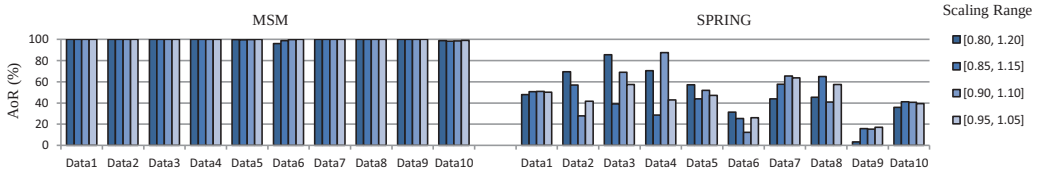

Figure 4.4: MSM outperforms SPRING at every scaling range in terms of AoR.

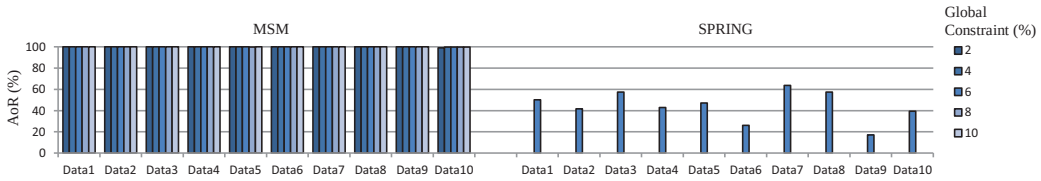

Figure 4.5: MSM outperforms SPRING at every global constraint value in terms of AoR

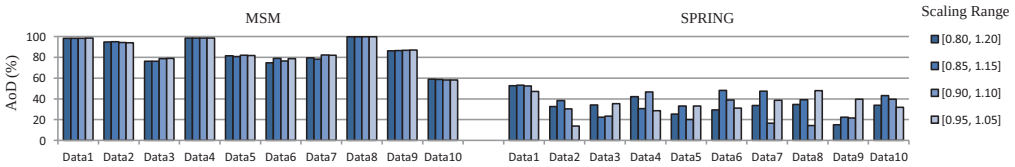

Figure 4.6: MSM outperforms SPRING at every scaling range in terms of AoD

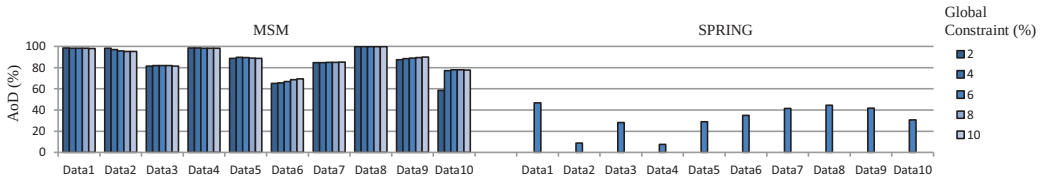

Figure 4.7: MSM outperforms SPRING at every global constraint value in terms of AoD

4.4 Conclusion

This chapter proposes a novel and meaningful subsequence matching algorithm, so called MSM (Meaningful Subsequence Matching), under global constraint, uniform scaling, and normalization. Two ideas have been introduced in MSM algorithm, i.e., a multi-resolution lower-bounding function LB_GUN (Lower-Bounding distance function under Global constraint, Uniform scaling, and Normalization, and a Scaling Subsequence Matrix (SSM) which estimates value of LB_GUN for each candidate subsequence. The proposed algorithm can update lower-bounding distance incrementally under normalization, while guaranteeing no false dismissals in linear time. With these two ideas, MSM algorithm can efficiently monitor a data stream and can answer both optimal range query and optimal top-k query problems. Since none of the current algorithms produce meaningful result, this thesis evaluates the proposed method comparing with the naive method in terms of time consumption and SPRING, the best existing subsequence matching under DTW distance, in terms of accuracies. As expected, the proposed MSM algorithm is much faster and more accurate by a very large margin.

CHAPTER V

CONCLUSIONS

This thesis introduces a new framework of frequent episode discovery on real-valued time series by proposing a concept to identify events from interesting patterns based on their shapes instead of a single value in a sequence. A new Subsequence Time Series (STS) clustering named Selective Subsequence Time Series Clustering (SSTSC) is introduced to transform a real-valued time series to a discrete event sequence. The proposed SSTSC is designed to identify significant patterns and discard trivial ones to minimize pattern inflations and redundancies, which occur in other works, while maintaining the meaningfulness of clustering results at the same time. More specifically, outputs of the SSTSC truly present characteristics of a time series input.

The proposed SSTSC is design to avoid overly-identified patterns by using data compression as a heuristic function to selectively collect interesting patterns. The algorithm are also made to allow length of the patterns to be variable by utilizing the uniform scaling technique. This thesis also explores utilization of Dynamic Time Warping (DTW) distance and shape-based averaging method to increase effectiveness of the previously proposed algorithm. Furthermore, the proposed algorithm provides parameter suggestion, so it can perform clustering by requiring only a few parameters where users can easily and flexibly adjust. The framework is also designed to be divided into common subtasks, such as subsequence matching and motif discovery, so that it is made to be manageable for further optimization.

This thesis assures the effectiveness and efficiency of the proposed algorithm by experimenting in various data domains. The experimental results show that the algorithm can identify significant patterns without giving redundancies and inflating results, and more importantly, can ensure that the results are meaningful by an extensive set of experimental evaluations.

This thesis also proposes an efficiency improvement on a usage of DTW as a distance measure in the framework. One of the subtasks that consumes high computational power is the subsequence matching task. A new subsequence matching algorithm, so called Meaningful Subsequence Matching (MSM) is proposed. The MSM provides subsequence search with DTW distance under global constraint, uniform scaling, and normalization. The MSM algo-

rithm introduces a multi-resolution lower-bounding function LB_GUN (Lower-Bounding distance function under Global constraint, Uniform scaling, and Normalization), and a Scaling Subsequence Matrix (SSM) that estimates value of LB_GUN. The SSM and LB_GUN guarantee lower bound with no false dismissals. The experiments show a huge increase of efficiency and effectiveness of the proposed MSM algorithm.

The following are the summary of contributions of this thesis.

- This is the first work to address frequent episode discovery problem on real-valued time series effectively.
- The proposed framework is shown to be able to be applied in real world applications.
- This thesis proposes a new frequent episode discovery framework. The following are details of the proposed framework.
  - The framework includes a new STS clustering algorithm for identification of interesting patterns, while trivial patterns are discarded.
  - The framework uses a compression based objective function to perform clustering.
  - The framework allows variability in length of the patterns.
  - The framework provides the best cluster number parameter suggestion.
  - The framework is suitable for frequent episode discovery and other applications such as rule discovery and prediction.
- The proposed framework is intentionally designed to be divided into common subtasks, so that it is designed to be manageable for further optimization.
- This thesis proposes effectiveness improvements to the previously proposed STS clustering method by utilizing DTW distance, and shape-based averaging technique.
- This thesis proposes efficiency improvements on using DTW in a subsequence search subtask.

Future research directions

Because this thesis proposed a new framework, there are some future improvements that can be explored. In motif discovery subtask, using DTW distance can provide exceptional

results; however, it is traded with very high computational time in exchange. For this reason, efficiency improvement on the usage of DTW can make the motif discovery more useful. Other directions are to explore other compression techniques to improve efficiency and effectiveness of the proposed framework.

References

Achar, A., Laxman, S., and Sastry, P. S. A unified view of the apriori-based algorithms for frequent episode discovery. Knowl. Inf. Syst., 2012, 31,2:223–250.

Agrawal, R. and Srikant, R. Fast Algorithms for Mining Association Rules. In Very Large Data Bases, 1994.

Agrawal, R. and Srikant, R. Mining Sequential Patterns. In International Conference on Data Engineering, pp. 3–14, 1995.

Alvarez, F. M., Troncoso, A., Riquelme, J. C., and Ruiz, J. S. A. Energy Time Series Forecasting Based on Pattern Sequence Similarity. IEEE Transactions on Knowledge and Data Engineering, 2011, 23:1230–1243.

Athitsos, V., Papapetrou, P., Potamias, M., Kollios, G., and Gunopulos, D. Approximate embedding-based subsequence matching of time series. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08, pp. 365–378, New York, NY, USA, 2008. ACM.

Batista, G. E. A. P. A., Wang, X., and Keogh, E. J. A complexity-invariant distance measure for time series. In Proceedings of the 2011 SIAM International Conference on Data Mining (SDM'11), pp. 699–710, Arizona, USA, 2011.

Bouqata, B., Carothers, C. D., Szymanski, B. K., and Zaki, M. J. Vogue: a novel variable order-gap state machine for modeling sequences. In European Conference on Principles of Data Mining and Knowledge Discovery, pp. 42–54. Springer, 2006.

Burden, R. L., Faires, J. D., and Reynolds, A. G. Numerical Analysis. Brooks Cole, 1997.

Camerra, A., Palpanas, T., Shieh, J., and Keogh, E. J. iSAX 2.0: Indexing and Mining One Billion Time Series. In IEEE International Conference on Data Mining, pp. 58–67, 2010.

Casas-Garriga, G. Discovering unbounded episodes in sequential data. In European Conference on Principles of Data Mining and Knowledge Discovery, pp. 83–94. Springer, 2003.

Chen, J. R. Making subsequence time series clustering meaningful. In Proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05), pp. 114–121, Texas, USA, 2005.

Chen, J. R. Useful clustering outcomes from meaningful time series clustering. In Proceedings of the Sixth Australasian Conference on Data Mining and Analytics - Volume 70, AusDM '07, pp. 101–109, Darlinghurst, Australia, Australia, 2007a. Australian Computer Society, Inc.

Chen, J. R. Making clustering in delay-vector space meaningful. Knowledge and Information Systems, 2007b, 11,3:369–385.

Cotofrei, P. and Stoffel, K. Classification rules + time = temporal rules. In Proceedings of 2002 International Conference on Computational Science, pp. 572–581, Amsterdan, Netherlands, 2002.

Das, G., Lin, K., Mannila, H., Renganathan, G., and Smyth, P. Rule discovery from time series. In 4th International Conference on Knowledge Discovery and Data Mining (KDD'98), pp. 16–22, New York, USA, 1998.

A. Araújo, R.de . A class of hybrid morphological perceptrons with application in time series forecasting. Knowledge-Based Systems, May 2011, 24,4:513–529.

Denton, A. M., Besemann, C. A., and Dorr, D. H. Pattern-based time-series subsequence clustering using radial distribution functions. Knowledge and Information Systems, January 2009, 18:1–27.

Drusinsky, D. Monitoring temporal rules combined with time series. In In CAV'03, volume 2725 of LNCS, pp. 114–118. Springer-Verlag, 2003.

Free, M., Seidel, D. J., Angell, J. K., Lanzante, J., Durre, I., and Peterson, T. C. Radiosonde atmospheric temperature products for assessing climate (ratpac): A new data set of large-area anomaly time series. Journal of Geophysical Research: Atmospheres, 2005, 110,D22:n/a–n/a. D22101.

Frith, M. C., Li, M. C., and Weng, Z. Cluster-buster: Finding dense clusters of motifs in dna sequences. Nucleic acids research, 2003, 31,13:3666–3668.

Fu, A. W.-C., Keogh, E., Lau, L. Y., Ratanamahatana, C. A., and Wong, R. C.-W. Scaling and time warping in time series querying. The VLDB Journal, 2008, 17:899–921.

Fujimaki, R., Hirose, S., and Nakata, T. Theoretical analysis of subsequence time-series clustering from a frequency-analysis viewpoint. In Proceedings of the 2008 SIAM International Conference on Data Mining (SDM'08), pp. 506–517, Georgia, USA, 2008.

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation, 2000, 101,23:e215–e220.

Goldin, D., 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 15th ACM International Conference on Information and Knowledge Management, CIKM '06, pp. 347–356, New York, NY, USA, 2006. ACM.

Google Inc. Google Finance. [Online]. Available from : `https://www.google.com/finance` , 2016.

Han, J., Kamber, M., and Pei, J. Data Mining: Concepts and Techniques, Second Edition. Morgan Kaufmann, 2 edition, January 2006.

Huang, K. and Chang, C. Efficient mining of frequent episodes from complex sequences. Information Systems, 2008, 33:96–114.

Idé, T. Why does subsequence time-series clustering produce sine waves? In 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'06), pp. 211–222, Singapore, 2006.

Iwanuma, K., Takano, Y., and Nabeshima, H. On anti-monotone frequency measures for extracting sequential patterns from a single very-long data sequence. In Cybernetics and Intelligent Systems, 2004 IEEE Conference on, volume 1, pp. 213–217. IEEE, 2004.

Jin, X., Lu, Y., and Shi, C. Distribution discovery: Local analysis of temporal rules. In Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (ICDM'02), pp. 469–480, London, UK, 2002.

Keogh, E. and Lin, J. Clustering of time-series subsequences is meaningless: implications for previous and future research. Knowledge and Information Systems, 2005, 8,2: 154–177.

Keogh, E. and Ratanamahatana, A. C. Exact indexing of dynamic time warping. Knowledge and Information Systems, 2005, 7,3:358–386.

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` , 2011.

Kumar, N., Lolla, N., Keogh, E., Lonardi, S., and Ratanamahatana, C. A. Time-series bitmaps: a practical visualization tool for working with large time series databases. In SIAM 2005 Data Mining Conference, pp. 531–535. SIAM, 2005.

Kumar, R. P., Nagabhushan, P., and Chouakria-Douzal, A. Wavesim and adaptive wavesim transform for subsequence time-series clustering. In Information Technology, 2006. ICIT '06. 9th International Conference on, pp. 197–202, Dec 2006.

Lai, C.-P., Chung, P.-C., and Tseng, V. S. A novel two-level clustering method for time series data analysis. Expert Systems with Applications, 2010, 37:6319–6326.

Laxman, S. Discovering Frequent Episodes : Fast Algorithms, Connections With HMMs And Generalizations. PhD thesis, Indian Institute of Science, 2006.

Laxman, S., Sastry, P. S., and Unnikrishnan, K. P. Discovering frequent episodes and learning hidden markov models: a formal connection. IEEE Transactions on Knowledge and Data Engineering, Nov 2005, 17,11:1505–1517.

Laxman, S., Sastry, P. S., and Unnikrishnan, K. P. A fast algorithm for finding frequent episodes in event streams. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '07, pp. 410–419, New York, NY, USA, 2007.

Lee, Y.-S. and Tong, L.-I. Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming. Knowledge-Based Systems, 2011, 24,1:66 – 72.

Li, C.-S., Yu, P. S., and Castelli, V. Malm: a framework for mining sequence database at multiple abstraction levels. In Proceedings of the 7th international conference on Information and knowledge management (CIKM'98), pp. 267–272, New York, USA, 1998.

Li, H. and Guo, C. Piecewise cloud approximation for time series mining. Knowledge-Based Systems, 2011, 24,4:492 – 500.

Li, J., Xia, G., and Shi, X. Association rules mining from time series based on rough set. In Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference on, volume 1, pp. 509–516, 2006.

Li, Y., Ning, P., Wang, X. S., and Jajodia, S. Discovering Calendar-based Temporal Association Rules. In Workshops, pp. 111–118, 2001.

Li, Y., Lin, J., and Oates, T. Visualizing variable-length time series motifs. In SDM, pp. 895–906. SIAM, 2012.

Li-ping, Q. and Mei, B. Predicting trend in futures prices time series using a new association rules algorithm. In International Conference on Management Science and Engineering, 2009.

Lin, J., Keogh, E. J., and Truppel, W. Clustering of streaming time series is meaningless. In Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, DMKD 2003, San Diego, California, USA, June 13, 2003, pp. 56–65, 2003.

Lutsiv, E. Association rules discovery in multivariate time series. In Spring Young Researchers Colloquium on Databases and Information Systems, 2007.

Mahesh Joshi, G. K. and Kumar, V. A universal formulation of sequential patterns. Technical report, Department of Computer Science, Universit of Minnesota, Minneapolis, 1999.

Mannila, H. and Toivonen, H. Discovering generalized episodes using minimal occurrences. In In Proceedings of the 2nd International Conference on Knowledge Discovery in Databases and Data Mining, pp. 146–151. AAAI Press, 1996.

Mannila, H., Toivonen, H., and Verkamo, A. I. Discovering frequent episodes in sequences (extended abstract). In In 1st Conference on Knowledge Discovery and Data Mining, pp. 210–215, 1995.

Mannila, H., Toivonen, H., and Inkeri Verkamo, A. Discovery of frequent episodes in event sequences. Data Min. Knowl. Discov., January 1997, 1,3:259–289.

Manning, C. D., Raghavan, P., and Schütze, H. Introduction to Information Retrieval. New York, NY, USA, Cambridge University Press, 2008.

Market Statistics. The Stock Exchange of Thailand. [Online]. Available from : `http://www.set.or.th/en/market/market_statistics.html` , 2016.

Martínez-Álvarez, F., Troncoso, A., Riquelme, J. C., and Aguilar-Ruiz, J. S. Lbf: A labeled-based forecasting algorithm and its application to electricity price time series. In 2008 Eighth IEEE International Conference on Data Mining, pp. 453–461. IEEE, 2008.

Martínez-Álvarez, F., Troncoso, A., and Riquelme, J. C. Improving time series forecasting by discovering frequent episodes in sequences. In International Symposium on Intelligent Data Analysis, pp. 357–368. Springer, 2009.

Martinez-BallesterosF, M., Martinez-Alvarez, F., Troncoso, A., and Riquelme, J. C. An evolutionary algorithm to discover quantitative association rules in multidimensional time series. Soft Computing, 2011, pp. 1–20.

Meger, N. and Rigotti, C. Constraint-Based Mining of Episode Rules and Optimal Window Sizes. In Principles of Data Mining and Knowledge Discovery, pp. 313–324, 2004.

Morchen, F. and Ultsch, A. Mining hierarchical temporal patterns in multivariate time series. In Proceedings of the 27th Annual German Conference in Artificial Intelligence, pp. 127–140. Springer, 2004.

Mueen, A., Keogh, E. J., Zhu, Q., Cash, S., and Westover, B. Exact Discovery of Time Series Motifs. In SIAM International Conference on Data Mining (SDM'09), pp. 473–484, Nevada, USA, 2009.

Mueen, A., Keogh, E., and Young, N. Logical-shapelets: an expressive primitive for time series classification. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '11, pp. 1154–1162, New York, NY, USA, 2011. ACM.

Ng, A. and Fu, A. W.-C. Mining frequent episodes for relating financial events and stock trends. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 27–39. Springer, 2003.

Niennattrakul, V. Meaningful Subsequence Clustering for Time Series Data Stream. PhD thesis, Chulalongkorn University, 2010.

Niennattrakul, V., Wanichsan, D., and Ratanamahatana, C. A. Accurate subsequence matching on data stream under time warping distance. In Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th International Conference on, volume 02, pp. 752–755, May 2009.

Niennattrakul, V., Srisai, D., and Ratanamahatana, C. A. Shape-based template matching for time series data. Knowledge-Based Systems, 2012, 26:1 – 8.

Oates, T. Identifying distinctive subsequences in multivariate time series by clustering. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 322–326. ACM, 1999.

Oates, T. Peruse: An unsupervised algorithm for finding recurring patterns in time series. In Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on, pp. 330–337. IEEE, 2002.

Ohsaki, M., Nakase, M., and Katagiri, S. Analysis of subsequence time-series clustering based on moving average. In Proceedings of the 9th IEEE International Conference on Data Mining (ICDM'09), pp. 902–907, Washington, DC, USA, 2009.

Ong, S. C. W. and Ranganath, S. Automatic sign language analysis: A survey and the future beyond lexical meaning. IEEE Transactions on Pattern Analysis and Machine Intelligence, June 2005, 27:873–891.

Orlando, S. and Foscari, U. C. A new algorithm for gap constrained sequence mining. In in ŚSAC Proceedings of the 2004 ACM Symposium on Applied Computing (SAC)Š, ACM, pp. 540–547. Press, 2004.

Patnaik, D., Sastry, P. S., and Unnikrishnan, K. P. Inferring neuronal network connectivity from spike data: A temporal data mining approach. Scientific Programming, 2008, 16,1: 49–77.

Patnaik, D., Marwah, M., Sharma, R. K., and Ramakrishnan, N. Temporal data mining approaches for sustainable chiller management in data centers. ACM Transactions on Intelligent Systems and Technology (TIST), 2011, 2,4:34.

Peng, Z., Liang, S., Yan, J., WeiHong, H., and ShuQiang, Y. Fast similarity matching on data stream with noise. In Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on, pp. 194–199. IEEE, 2008.

Pradhan, G. N. and Prabhakaran, B. Association rule mining in multiple, multidimensional time series medical data. In International Conference on Multimedia Computing and Systems/International Conference on Multimedia and Expo, pp. 1720–1723, 2009a.

Pradhan, G. and Prabhakaran, B. Association rule mining in multiple, multidimensional time series medical data. In Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on, pp. 1720–1723, 2009b.

Qin, L.-X. and Shi, Z.-Z. Efficiently mining association rules from time series. International Journal of Information Technology, 2006, 12,4:30–38.

Rakthanmanon, T., Keogh, E. J., Lonardi, S., and Evans, S. Mdl-based time series clustering. Knowledge and Information Systems, 2012, 33,2:371–399.

Rand, W. M. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association, 1971, 66,336:846–850.

Ratanamahatana, C. A. and Keogh, E. Making Time-series Classification More Accurate Using Learned Constraints, chapter 2, pp. 11–22. 2004.

Ratanamahatana, C. A. and Keogh, E. Three Myths about Dynamic Time Warping Data Mining, chapter 50, pp. 506–510. 2005.

Rodpongpun, S., Niennattrakul, V., and Ratanamahatana, C. A. Efficient subsequence search on streaming data based on time warping distance. ECTI Transactions on Computer and Information Technology, 2011, 5,1:2–8.

Rodpongpun, S., Niennattrakul, V., and Ratanamahatana, C. A. Selective subsequence time series clustering. Knowledge-Based Systems, 2012, 35,0:361 – 368.

Sakurai, Y., Faloutsos, C., and Yamamuro, M. Stream monitoring under the time warping distance. In 2007 IEEE 23rd International Conference on Data Engineering, pp. 1046–1055, April 2007.

Salvador, S. and Chan, P. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on, pp. 576–584, Nov 2004.

Sang Hyun, P., Wesley, W., et al. Discovering and matching elastic rules from sequence databases. Fundamenta Informaticae, 2001, 47,1-2:75–90.

Sarker, B. K., Hirata, T., Uehara, K., and Bhavsar, V. C. Mining Association Rules from Multistream Time Series Data on Multiprocessor Systems. 2005.

Sart, D., Mueen, A., Najjar, W., Keogh, E., and Niennattrakul, V. Accelerating dynamic time warping subsequence search with gpus and fpgas. In 2010 IEEE International Conference on Data Mining, pp. 1001–1006. IEEE, 2010.

Schluter, T. and Conrad, S. About the analysis of time series with temporal association rule mining. In IEEE Symposium on Computational Intelligence and Data Mining, 2011.

Shokoohi-Yekta, M., Chen, Y., Campana, B., Hu, B., Zakaria, J., and Keogh, E. Discovery of meaningful rules in time series. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1085–1094. ACM, 2015.

Sivaraks, H. Morphology discovery in ECG artifacts using time series mining. PhD thesis, Department of Computer Engineering, Chulalongkorn University, 2014.

Srikant, R. and Agrawal, R. Mining sequential patterns: Generalizations and performance improvements. In Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology, EDBT '96, pp. 3–17, London, UK, UK, 1996. Springer-Verlag.

Srivatsan Laxman, B. C. Efficient episode mining of dynamic event streams. IEEE Computer Society, December 2012.

Tang, H. and Liao, S. S. Discovering original motifs with different lengths from time series. Knowledge-Based Systems, October 2008, 21,7:666–671.

Unnikrishnan, K., Shadid, B. Q., Sastry, P., and Laxman, S. March 24 2009. US Patent 7,509,234.

Wan, D., Zhang, Y., and Li, S. Discovery Association Rules in Time Series of Hydrology. In IEEE International Conference on Integration Technology, 2007.

Wan, L., Chen, L., and Zhang, C. Mining dependent frequent serial episodes from uncertain sequence data. In 2013 IEEE 13th International Conference on Data Mining, pp. 1211–1216. IEEE, 2013a.

Wan, L., Chen, L., and Zhang, C. Mining frequent serial episodes over uncertain sequence data. In Proceedings of the 16th International Conference on Extending Database Technology, pp. 215–226. ACM, 2013b.

Wang, B. and Chen, Y. Fuzzy logic for mining episodal association rules in time series. In IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009.

Wang, J., Zhang, Y., Zhou, L., Karypis, G., and Aggarwal, C. C. Contour: an efficient algorithm for discovering discriminating subsequences. Data Mining and Knowledge Discovery, 2009, 18,1:1–29.

Wang, M.-F., Wu, Y.-C., and Tsai, M.-F. Exploiting frequent episodes in weighted suffix tree to improve intrusion detection system. In Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on, pp. 1246–1252. IEEE, 2008.

Wang, X., Smith, K., and Hyndman, R. Characteristic-based clustering for time series data. Data Min. Knowl. Discov., November 2006, 13,3:335–364.

Warasup, K. and Nukoolkit, C. Discovery association rules in time series data. KMUTT Research and Development Journal, 2006, 4:447–462.

Weng, X. and Shen, J. Classification of multivariate time series using locality preserving projections. Knowledge-Based Systems, October 2008a, 21,7:581–587.

Weng, X. and Shen, J. Classification of multivariate time series using two-dimensional singular value decomposition. Knowledge-Based Systems, October 2008b, 21,7:535–539.

Wu, H., Salzberg, B., Sharp, G. C., Jiang, S. B., Shirato, H., and Kaeli, D. Subsequence matching on structured time series data. In Proceedings of the 2005 ACM SIGMOD international conference on Management of data (SIGMOD'05), pp. 682–693, New York, USA, 2005.

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 3rd international conference on Body area networks, p. 2. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

Yaik, O. B., Yong, C. H., and Haron, F. Time series prediction using adaptive association rules. In Distributed Frameworks for Multimedia Applications, pp. 310–314, 2005.

Yairi, T., Kato, Y., and Hori, K. Fault detection by mining association rules from house-keeping data. In Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space, pp. 18–21, Montreal, Canada, 2001.

Yang, R., Sarkar, S., and Loeding, B. Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32: 462–477.

Yang, T. and Wang, J. Clustering unsynchronized time series subsequences with phase shift weighted spherical k-means algorithm. Journal of Computers, 2014, 9,5:1103–1108.

Yankov, D., Keogh, E., Medina, J., Chiu, B., and Zordan, V. Detecting time series motifs under uniform scaling. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 844–853. ACM, 2007.

Yingchareonthawornchai, S., Sivaraks, H., Rakthanmanon, T., and Ratanamahatana, C. A. Efficient proper length time series motif discovery. In 2013 IEEE 13th International Conference on Data Mining, pp. 1265–1270, Dec 2013.

Zakaria, J., Mueen, A., and Keogh, E. J. Clustering time series using unsupervised-shapelets. In ICDM'12, pp. 785–794, 2012.

Zolhavarieh, S., Aghabozorgi, S., and Teh, Y. W. A review of subsequence time series clustering. The Scientific World Journal, 2014, 2014.

CHAPTER VI

PUBLICATIONS

1. Wei Chen, Sura Rodpongpun, William Luo, Nathan Isaacson, Lauren Kark, Heba Khamis, Stephen James Redmond, *An eight-legged tactile sensor to estimate coefficient of static friction*. 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 15), Milan, Italy, 2015.

2. Navin Madicar, Haemwaan Sivaraks, Sura Rodpongpun, Chotirat Ann Ratanamahatana, *An Enhanced Parameter-Free Subsequences Time Series Clustering for High-Variability-Width Data*. The First International conference on Soft Computing and Data Mining (SCDM 14), Johor, Malaysia, 2014.

3. Navin Madicar, Haemwaan Sivaraks, Sura Rodpongpun, Chotirat Ann Ratanamahatana. *Parameter-Free Subsequences Time Series Clustering with Various-width Clusters*. In Proceedings of International Conference on Knowledge and Smart Technology (KST 13), Thailand, 2013.

4. Sorrachai Yingchareonthawornchai, Haemwaan Sivaraks, Sura Rodpongpun, Chotirat Ann Ratanamahatana. *The Proper Length Motif Discovery Algorithm*. In Proceedings of International Computer Science and Engineering Conference (ICSEC 12), Pattaya, Thailand, 2012.

5. Sura Rodpongpun, Vit Niennattrakul, Chotirat Ann Ratanamahatana. *Selective Subsequence Time Series Clustering*. Knowledge-Based System. 35: 361-368, 2012.

6. Sura Rodpongpun, Vit Niennattrakul, Chotirat Ann Ratanamahatana. *Efficient Subsequence Search on Streaming Data Based on Time Warping Distance*. ECTI Transactions on Computer and Information Technology (ECTI-CIT). 5, 1: 1-7, 2011.

APPENDICES

APPENDICES A

COMPLETE EXPERIMENTAL RESULTS OF THE EXPERIMENT IN SECTION 3.5.3

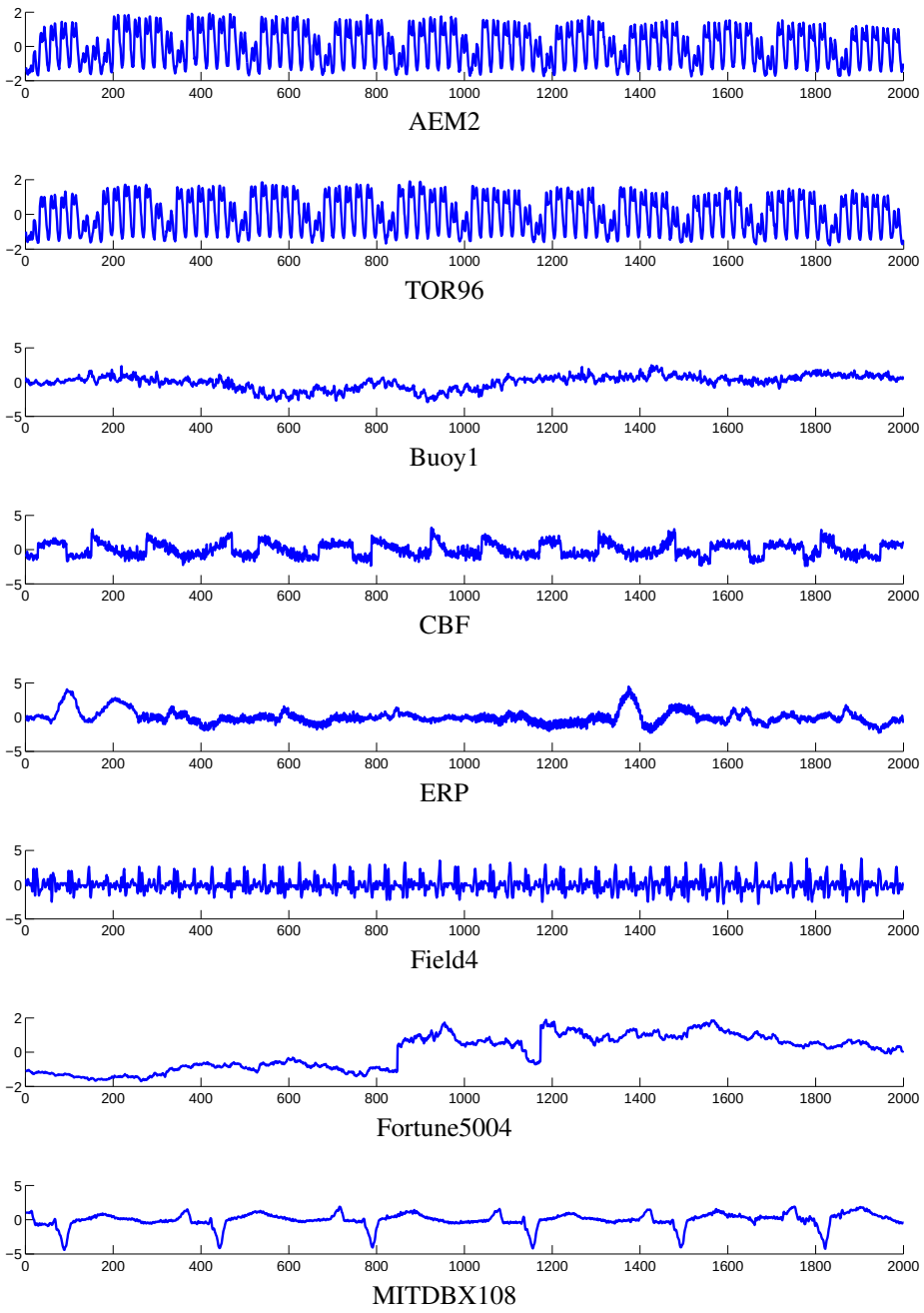

Figure A.1: TSDMA datasets used in the experiments in section 3.5.3.

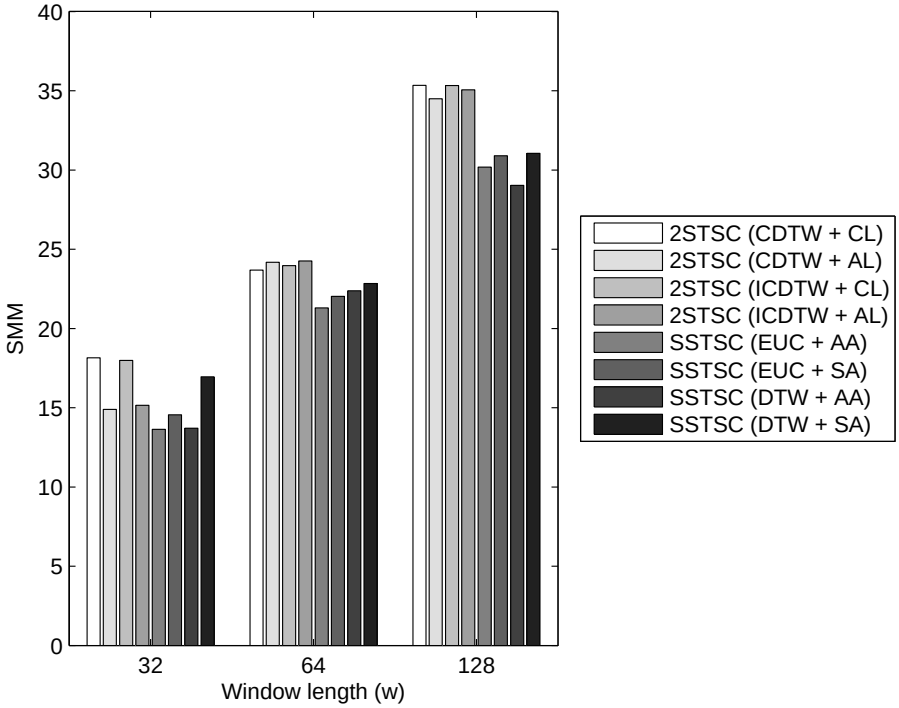

Figure A.2: SMMs of AEM2 dataset when number of clusters (k) is 3 and the length of sliding window (w) is varied.

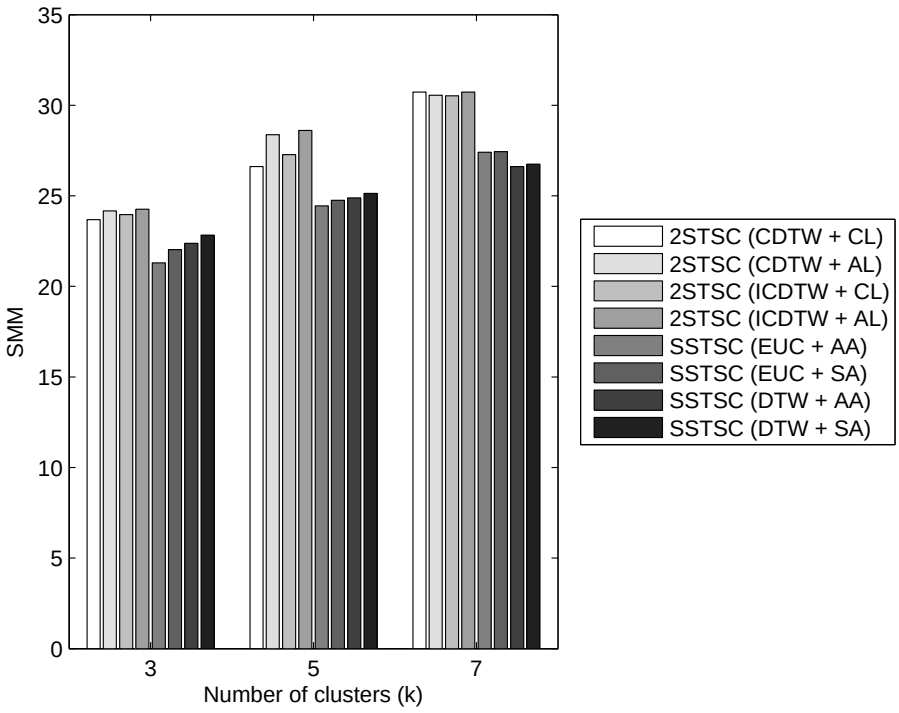

Figure A.3: SMMs of AEM2 dataset when number of clusters (k) is varied and the length of sliding window (w) is 64.

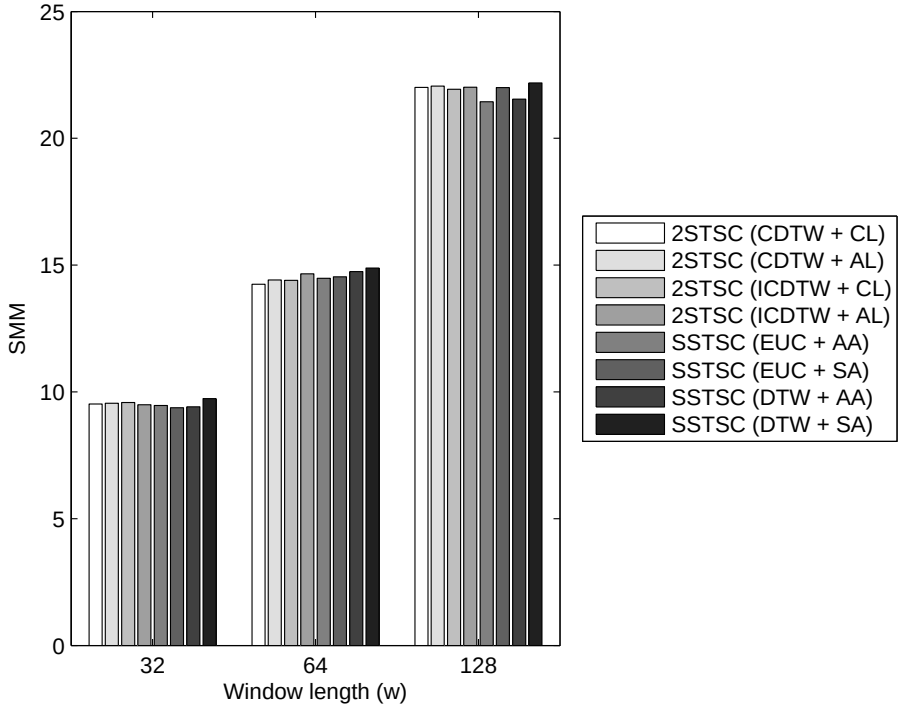

Figure A.4: SMMs of Buoy1 dataset when number of clusters (k) is 3 and the length of sliding window (w) is varied.

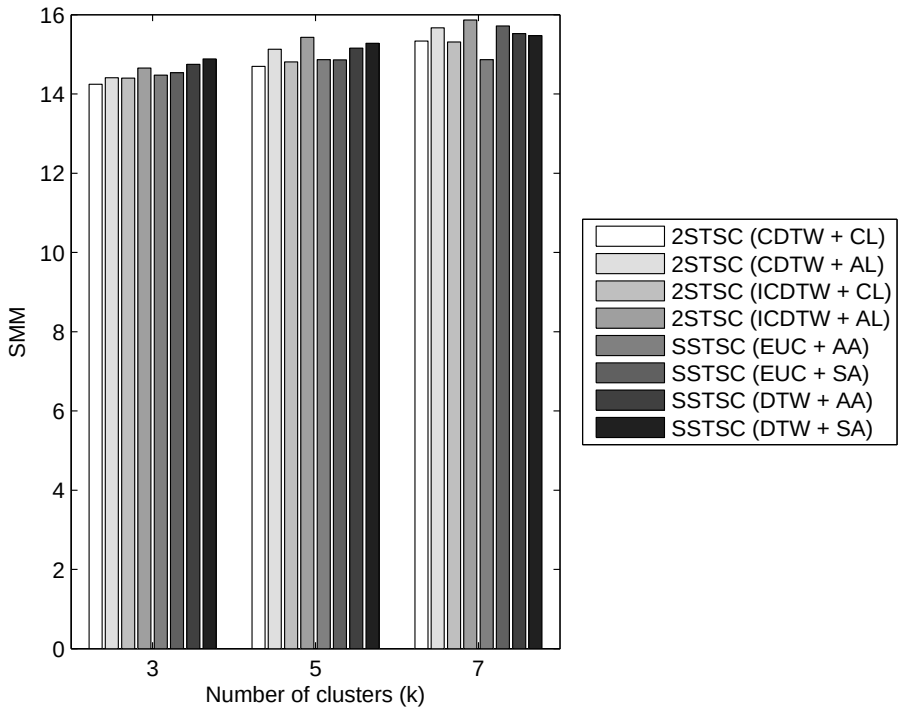

Figure A.5: SMMs of Buoy1 dataset when number of clusters (k) is varied and the length of sliding window (w) is 64.

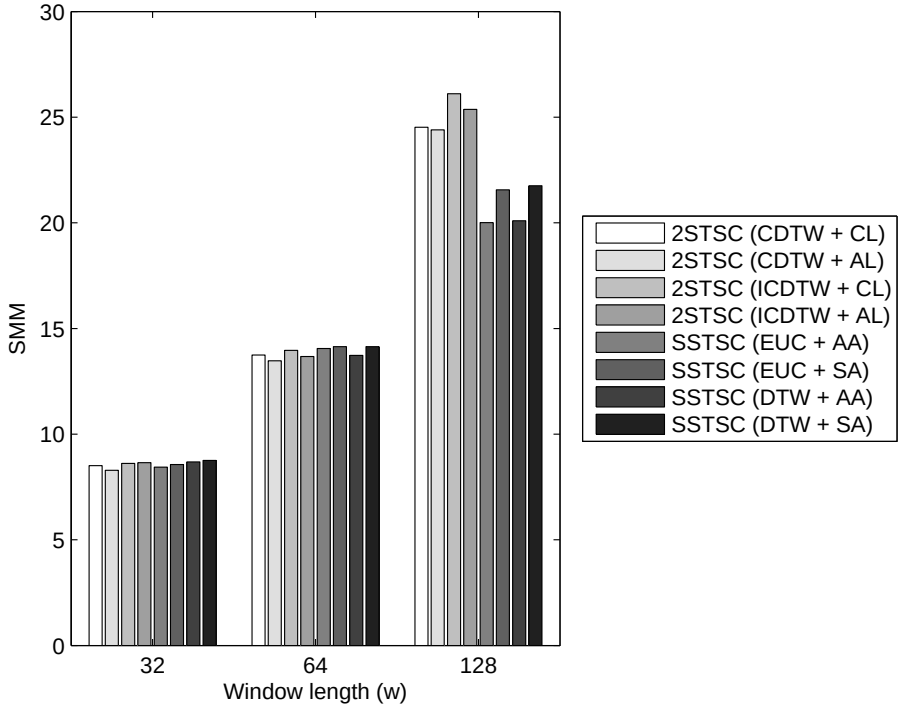

Figure A.6: SMMs of CBF dataset when number of clusters (k) is 3 and the length of sliding window (w) is varied.

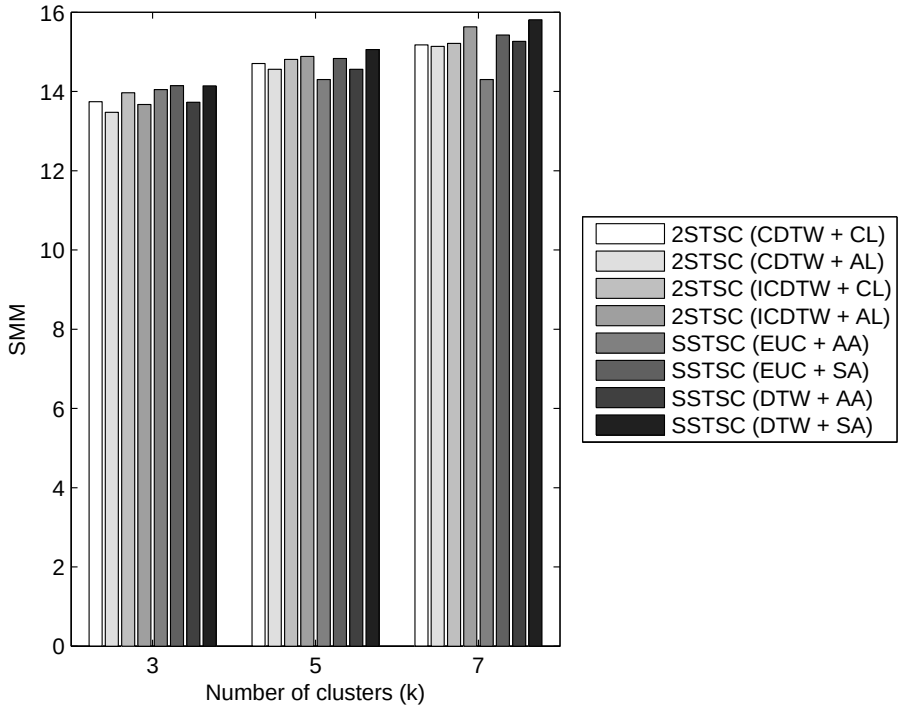

Figure A.7: SMMs of CBF dataset when number of clusters (k) is varied and the length of sliding window (w) is 64.

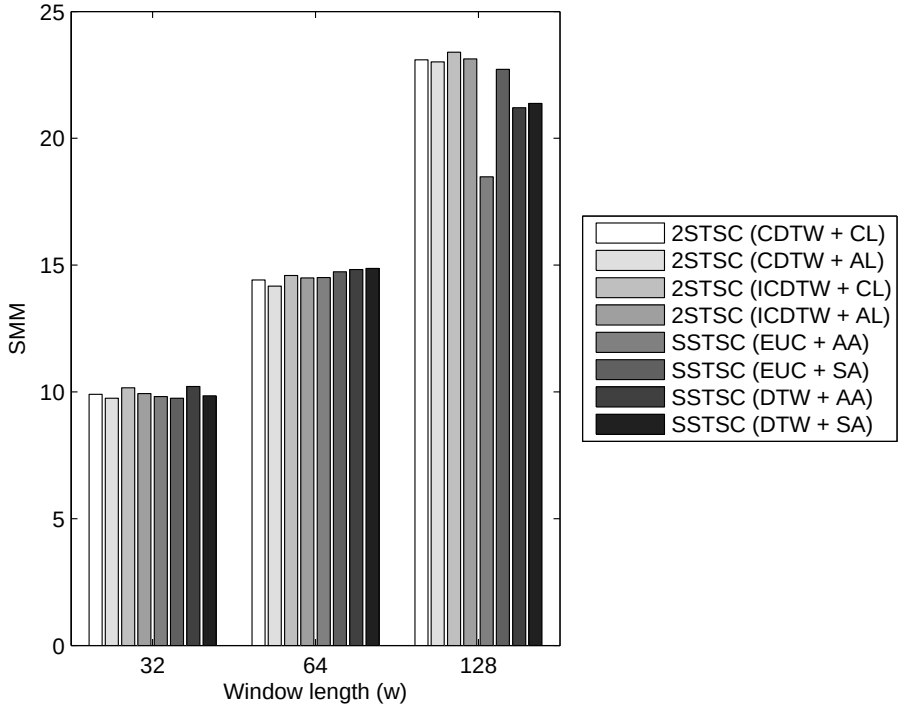

Figure A.8: SMMs of ERP dataset when number of clusters (k) is 3 and the length of sliding window (w) is varied.

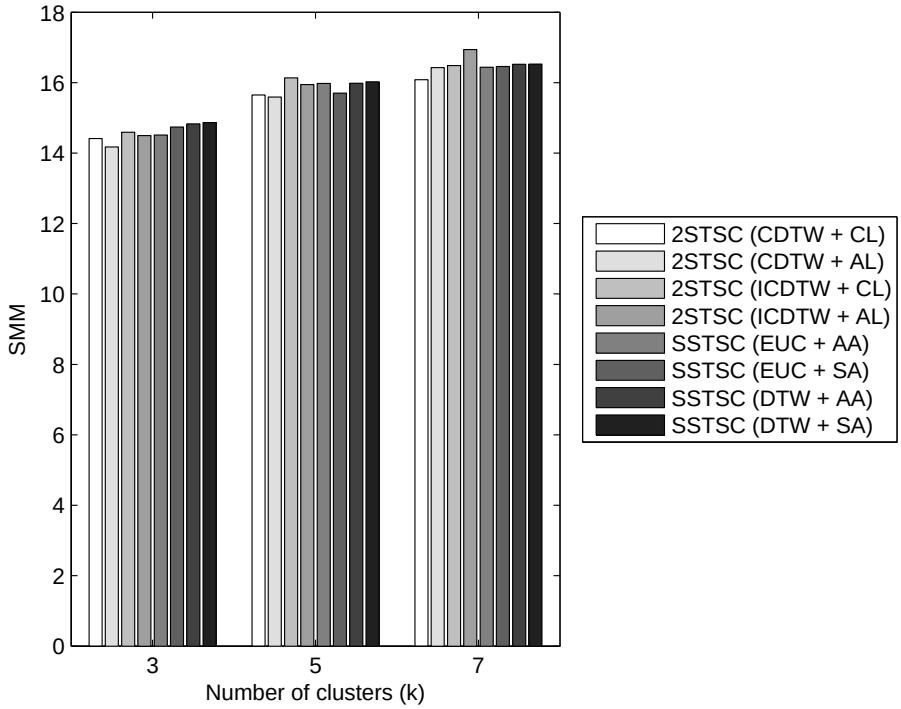

Figure A.9: SMMs of ERP dataset when number of clusters (k) is varied and the length of sliding window (w) is 64.

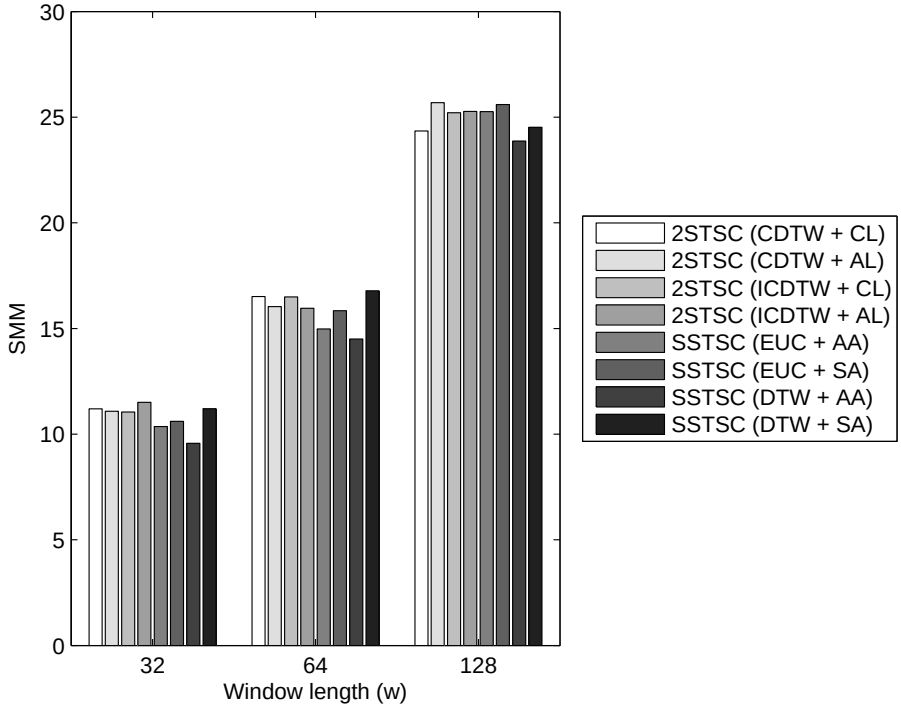

Figure A.10: SMMs of Field4 dataset when number of clusters (k) is 3 and the length of sliding window (w) is varied.

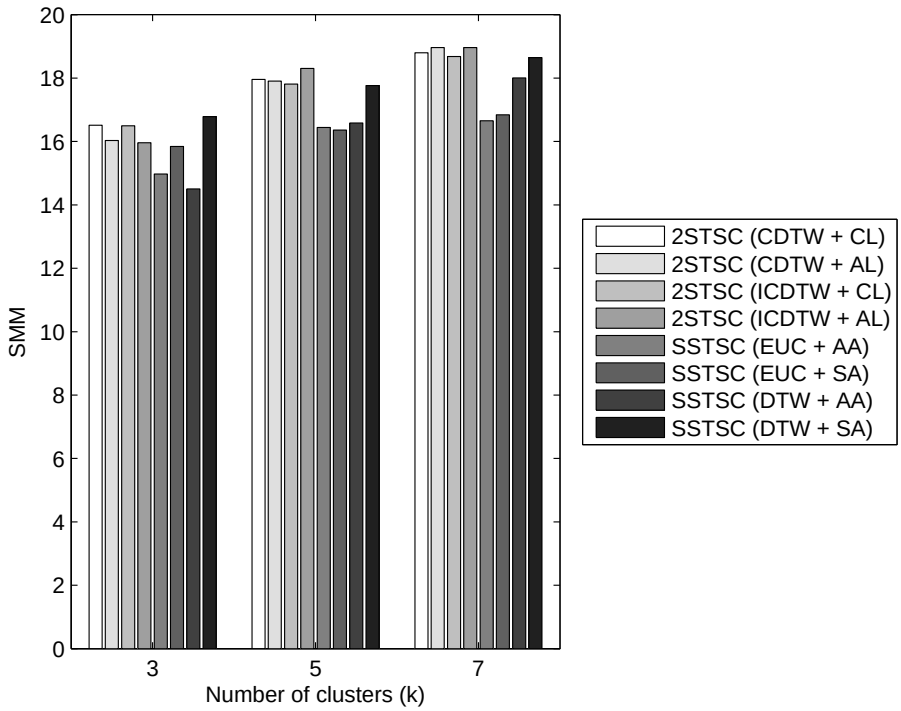

Figure A.11: SMMs of Field4 dataset when number of clusters (k) is varied and the length of sliding window (w) is 64.

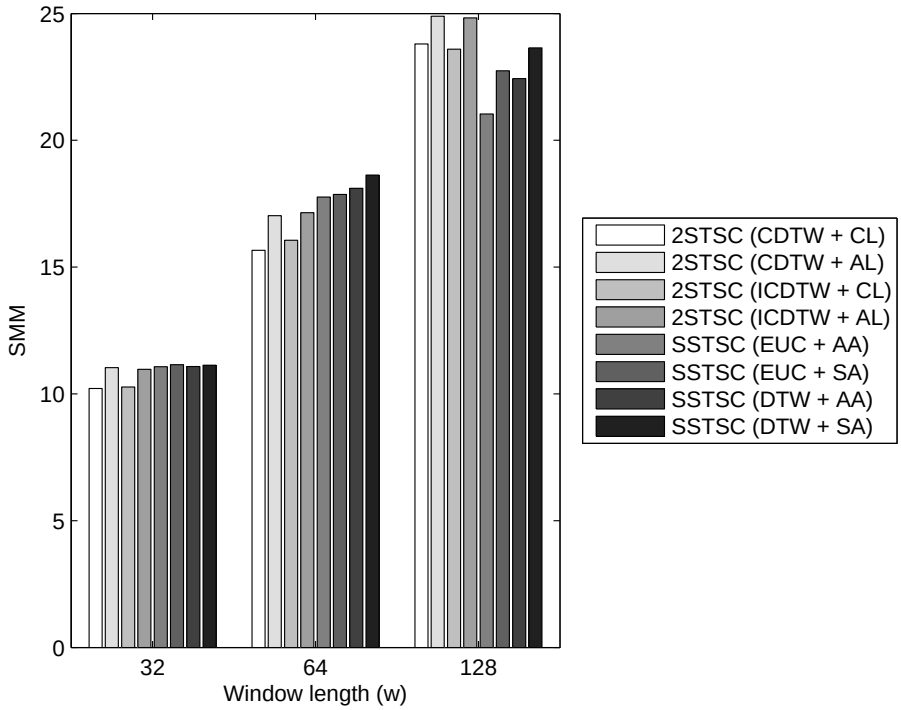

Figure A.12: SMMs of Fortune5004 dataset when number of clusters (k) is 3 and the length of sliding window (w) is varied.

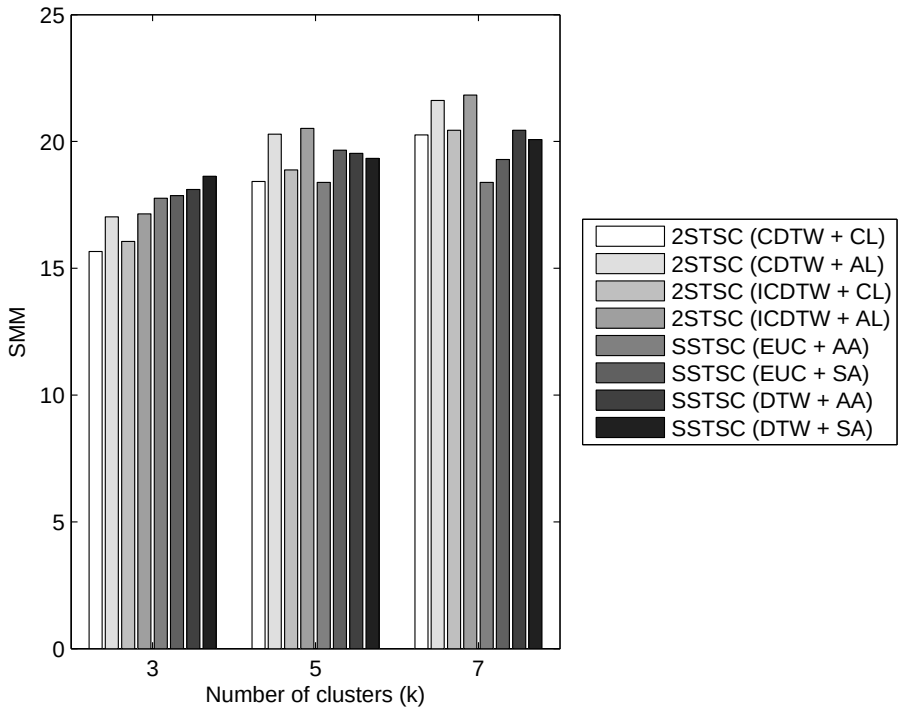

Figure A.13: SMMs of Fortune5004 dataset when number of clusters (k) is varied and the length of sliding window (w) is 64.

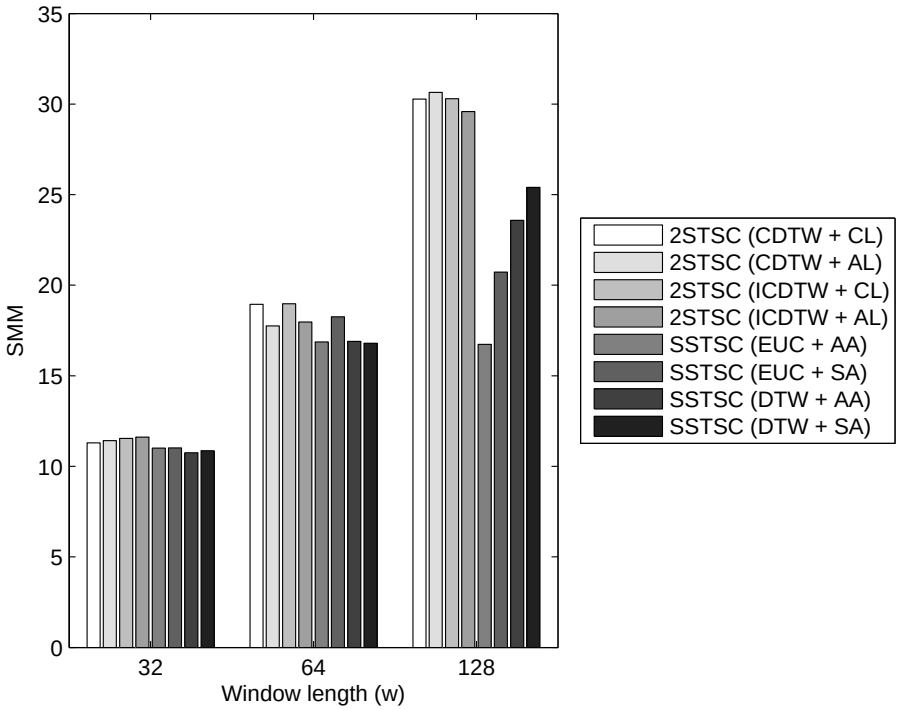

Figure A.14: SMMs of MITDBX108 dataset when number of clusters (k) is 3 and the length of sliding window (w) is varied.

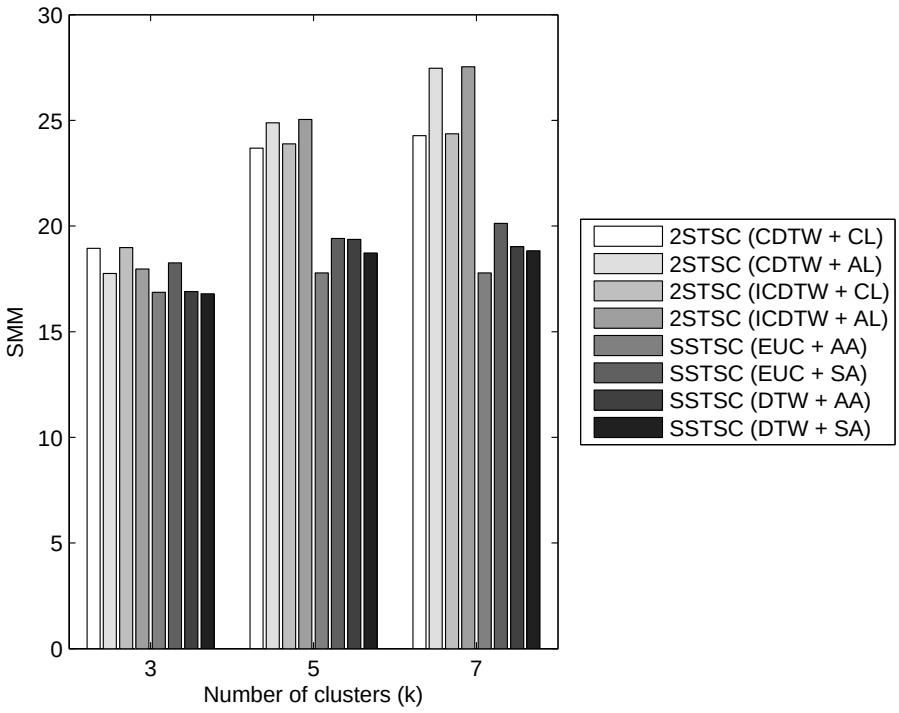

Figure A.15: SMMs of MITDBX108 dataset when number of clusters (k) is varied and the length of sliding window (w) is 64.

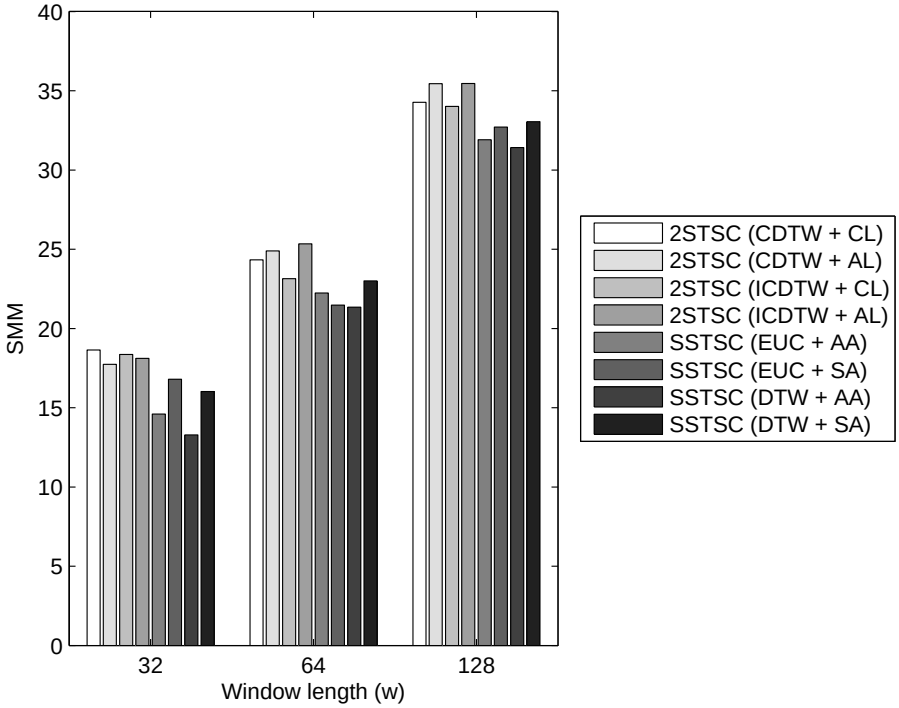

Figure A.16: SMMs of TOR96 dataset when number of clusters (k) is 3 and the length of sliding window (w) is varied.

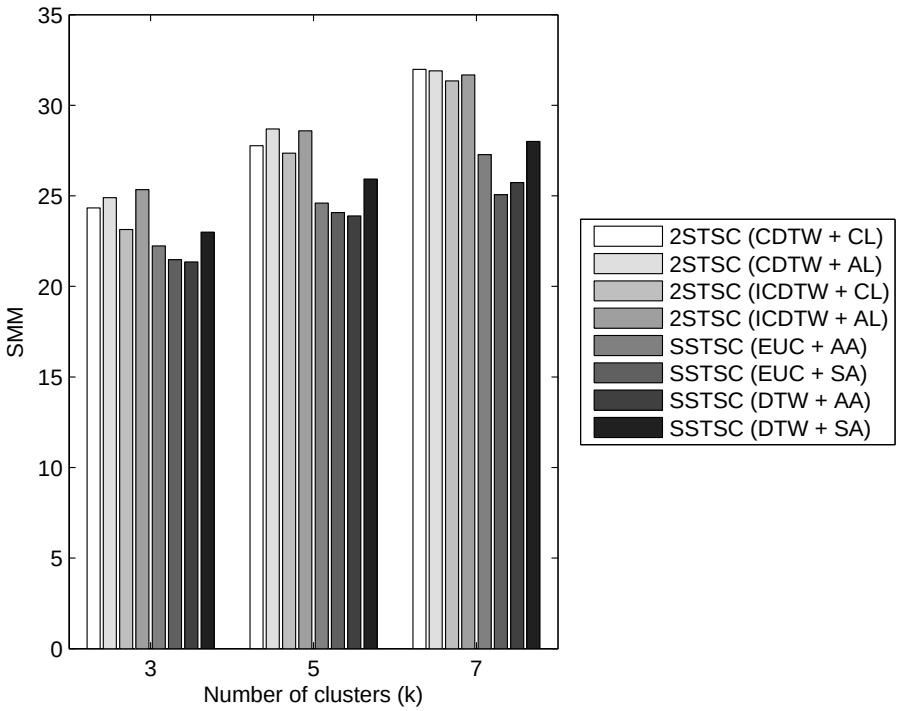

Figure A.17: SMMs of TOR96 dataset when number of clusters (k) is varied and the length of sliding window (w) is 64.

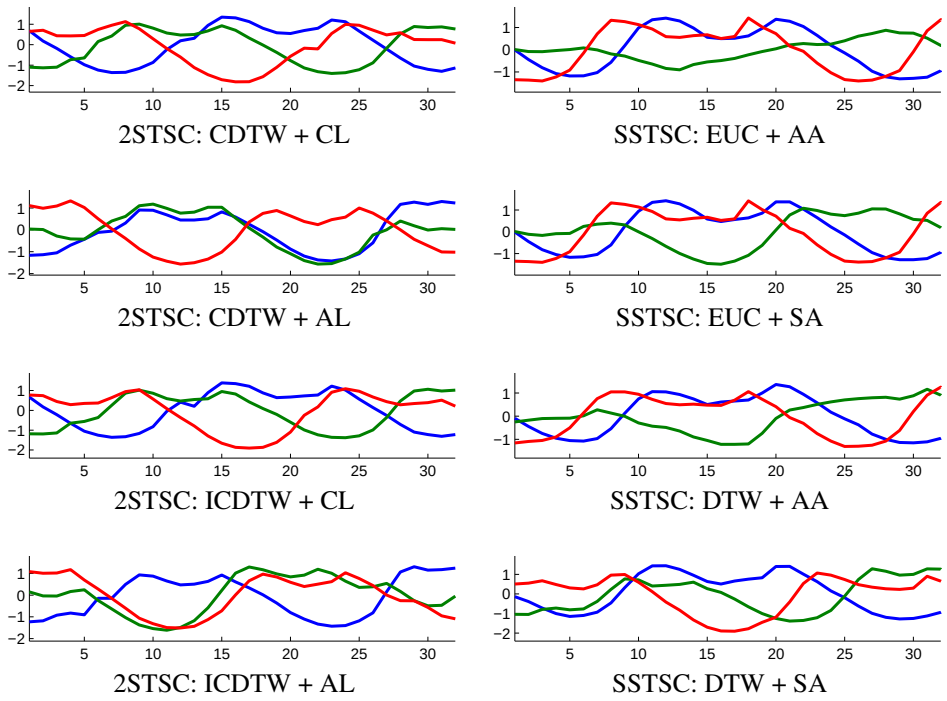

Figure A.18: Cluster representatives of AEM2 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 32$.

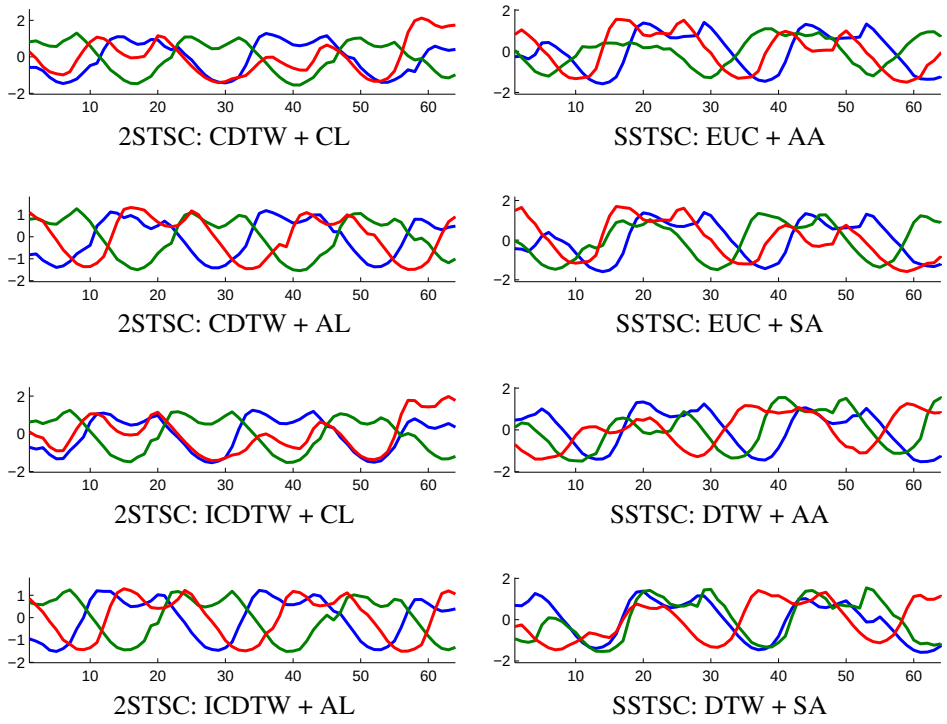

Figure A.19: Cluster representatives of AEM2 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 64$.

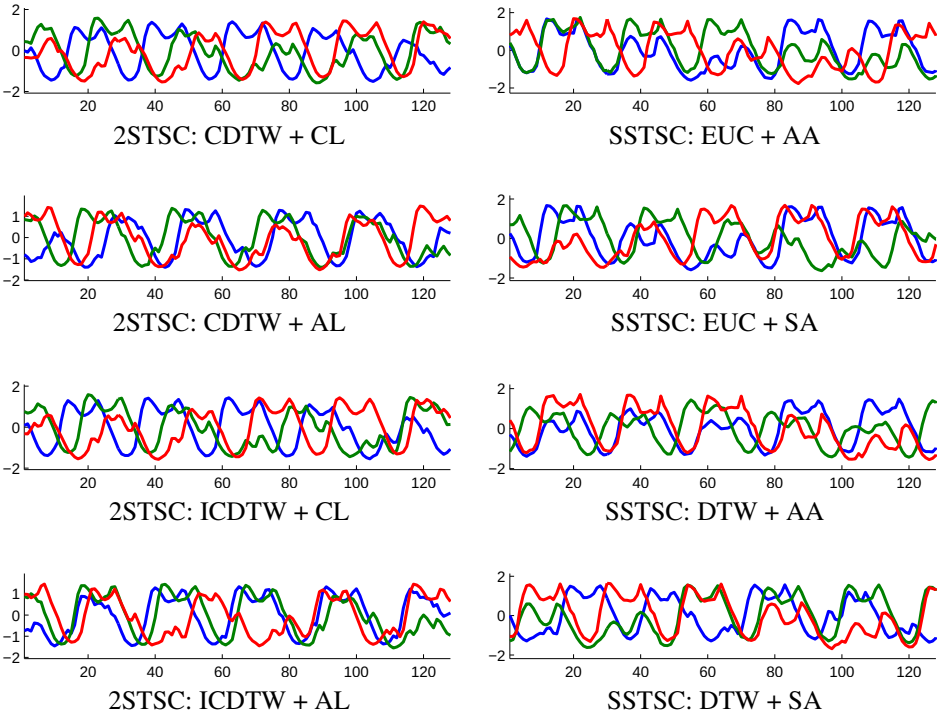

Figure A.20: Cluster representatives of AEM2 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 128$.

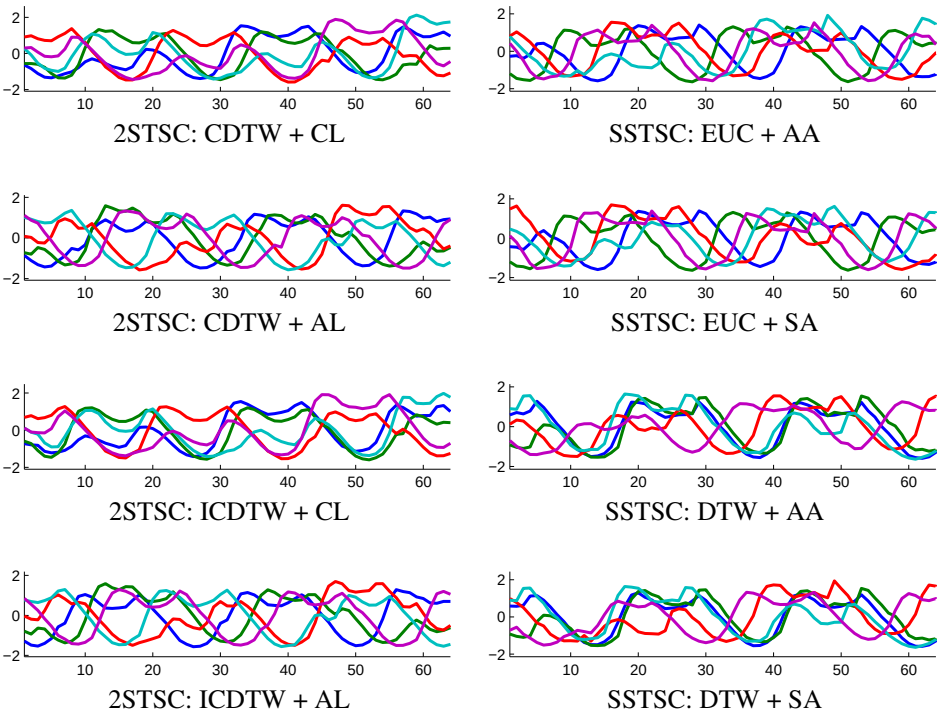

Figure A.21: Cluster representatives of AEM2 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 5$ and $w = 64$.

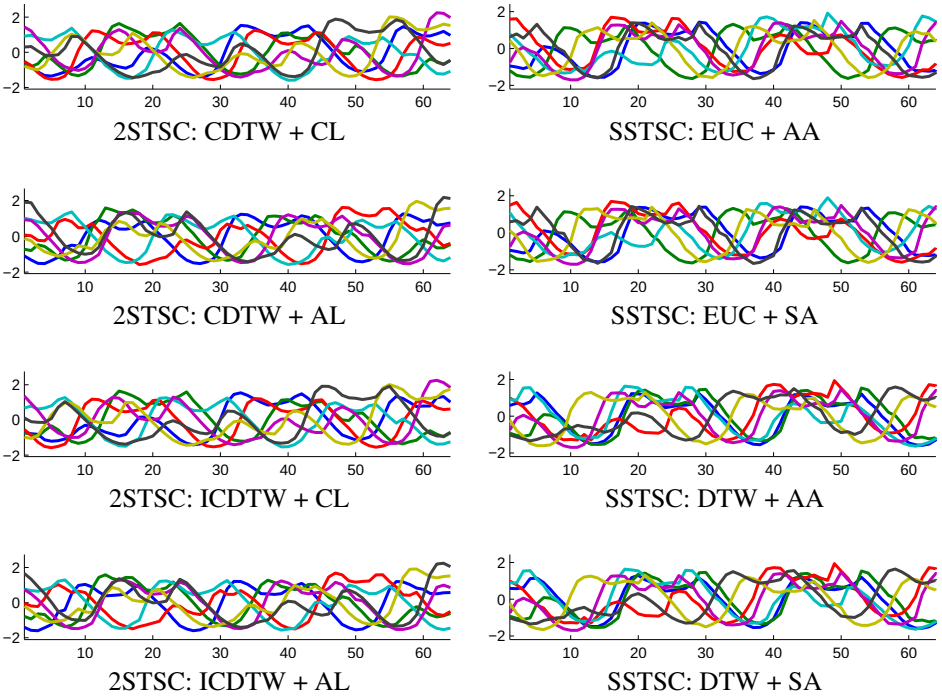

Figure A.22: Cluster representatives of AEM2 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 7$ and $w = 64$.

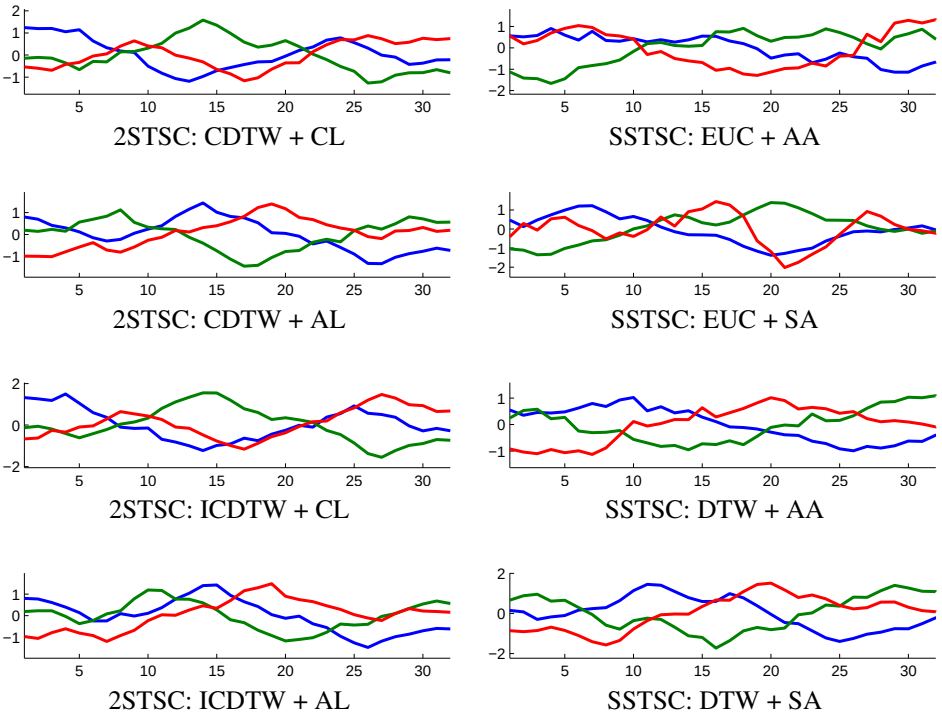

Figure A.23: Cluster representatives of buoy1 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 32$.

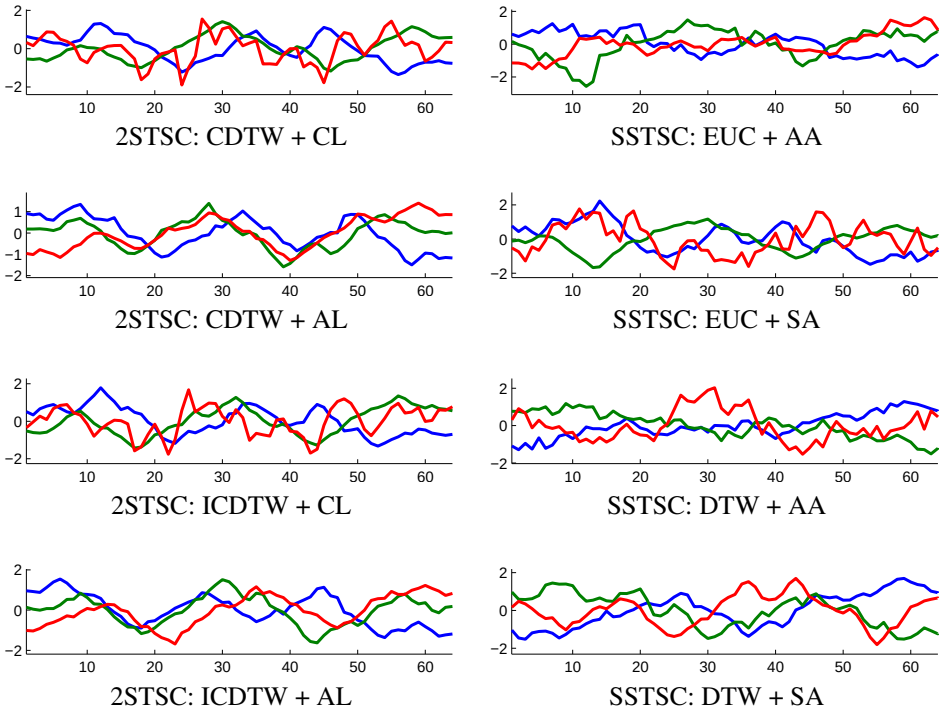

Figure A.24: Cluster representatives of buoy1 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 64$.

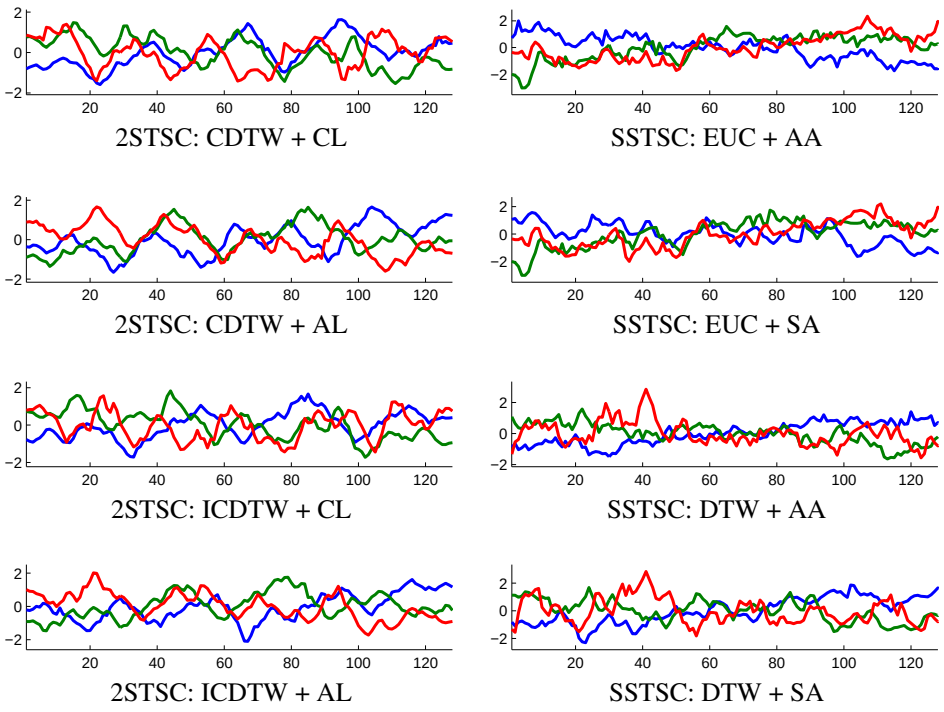

Figure A.25: Cluster representatives of buoy1 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 128$.

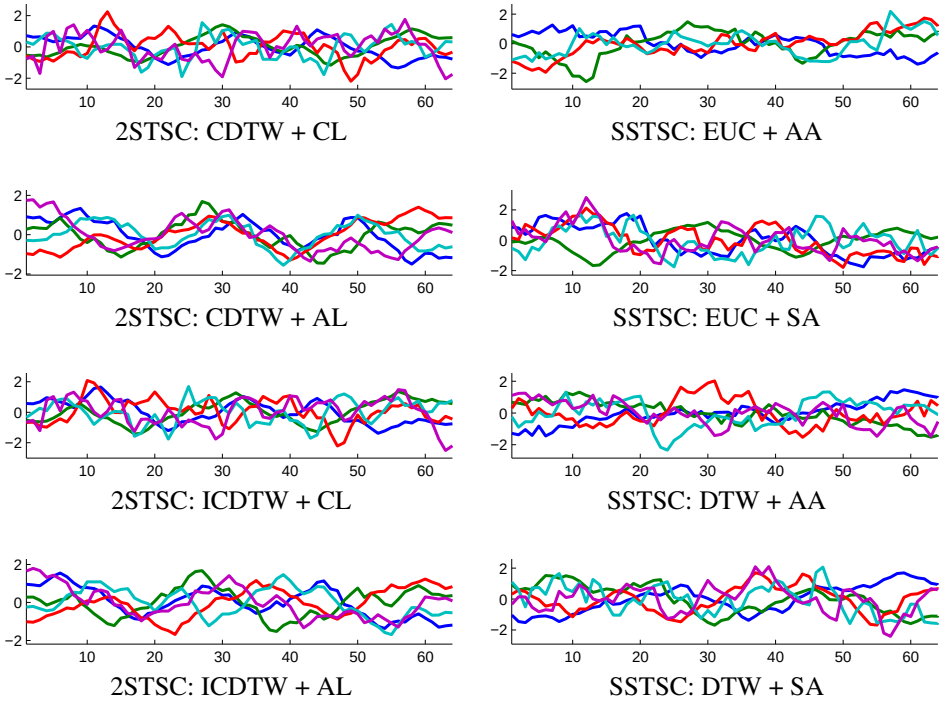

Figure A.26: Cluster representatives of buoy1 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 5$ and $w = 64$.

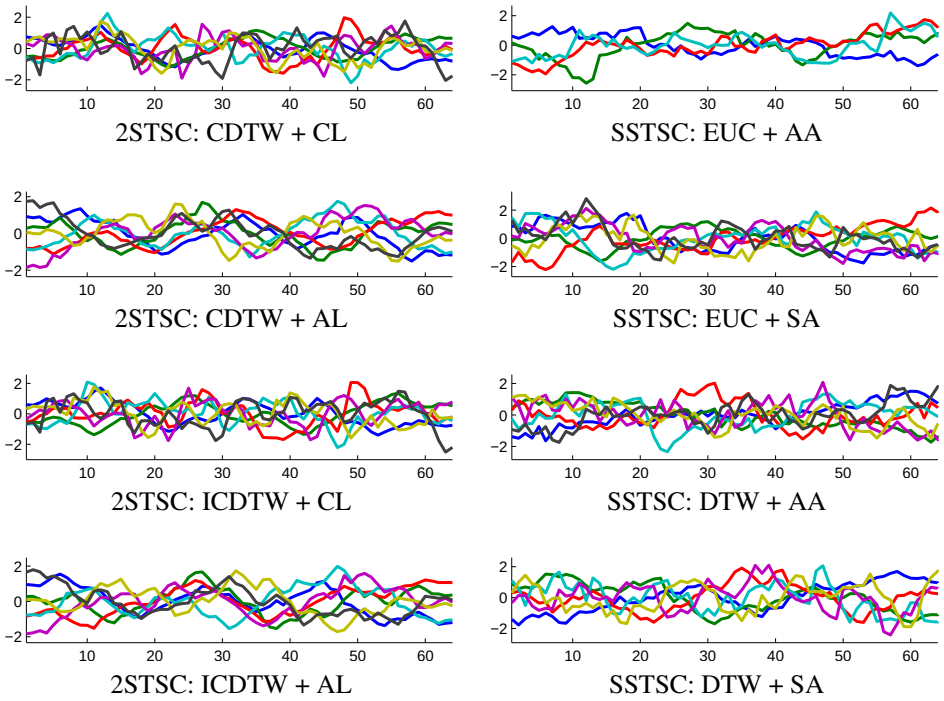

Figure A.27: Cluster representatives of buoy1 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 7$ and $w = 64$.

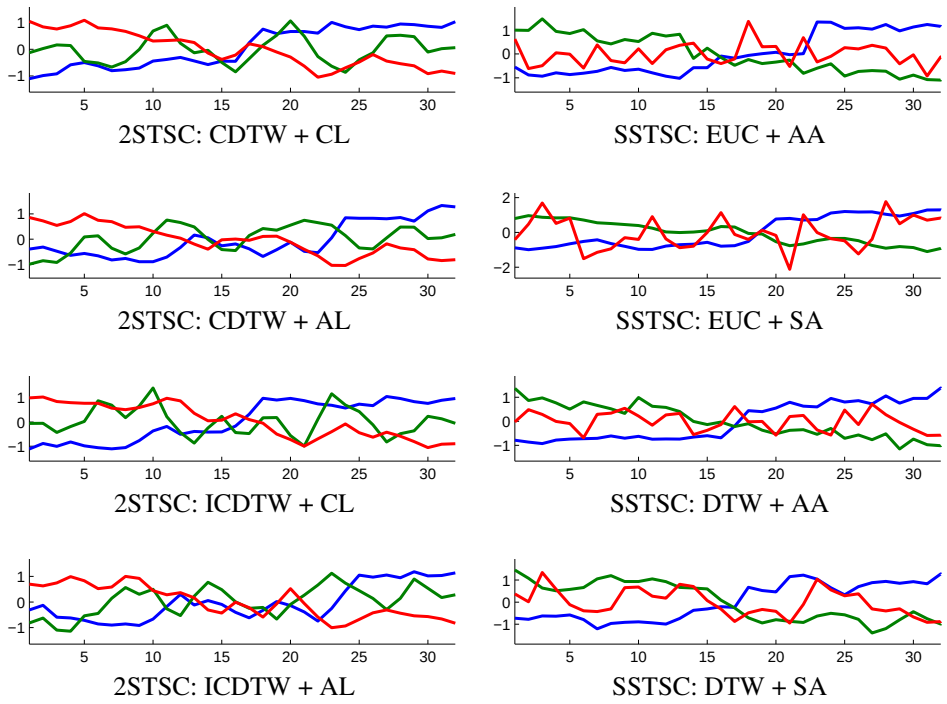

Figure A.28: Cluster representatives of CBF dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 32$.

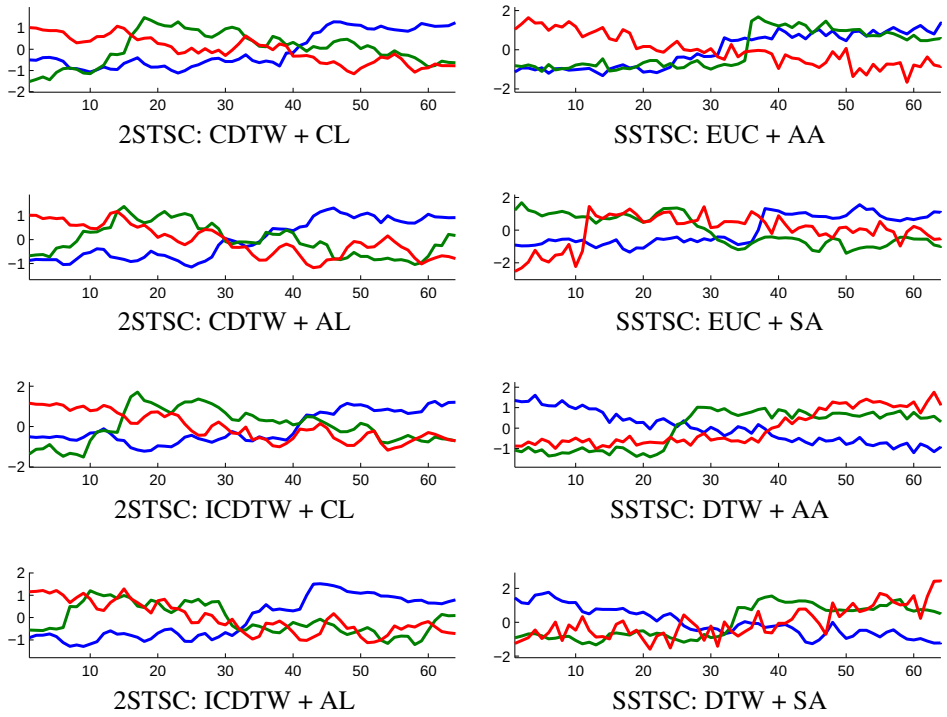

Figure A.29: Cluster representatives of CBF dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 64$.

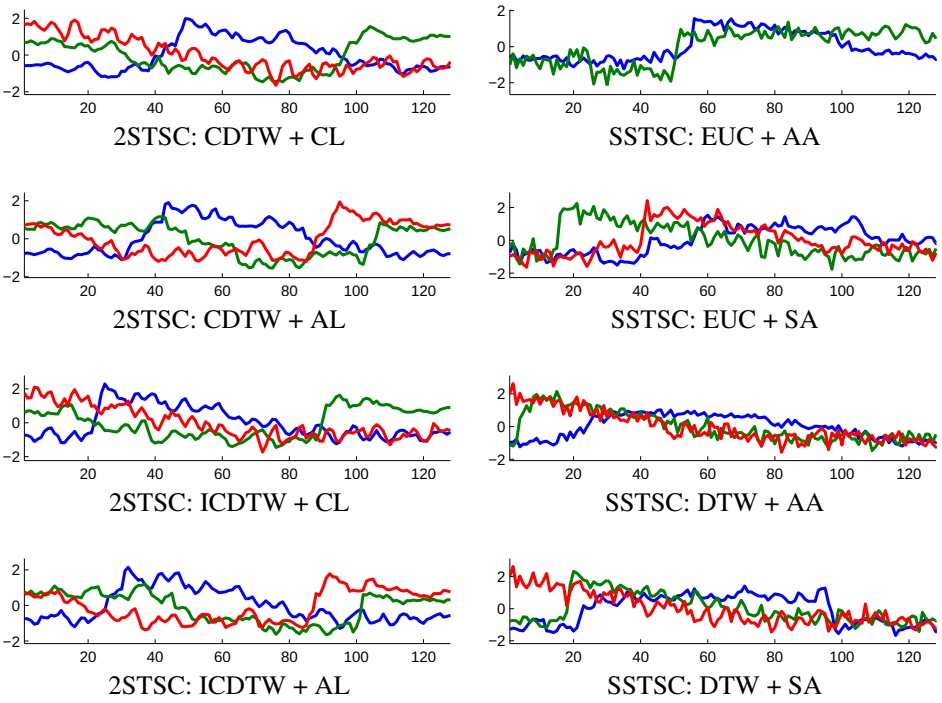

Figure A.30: Cluster representatives of CBF dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 128$.

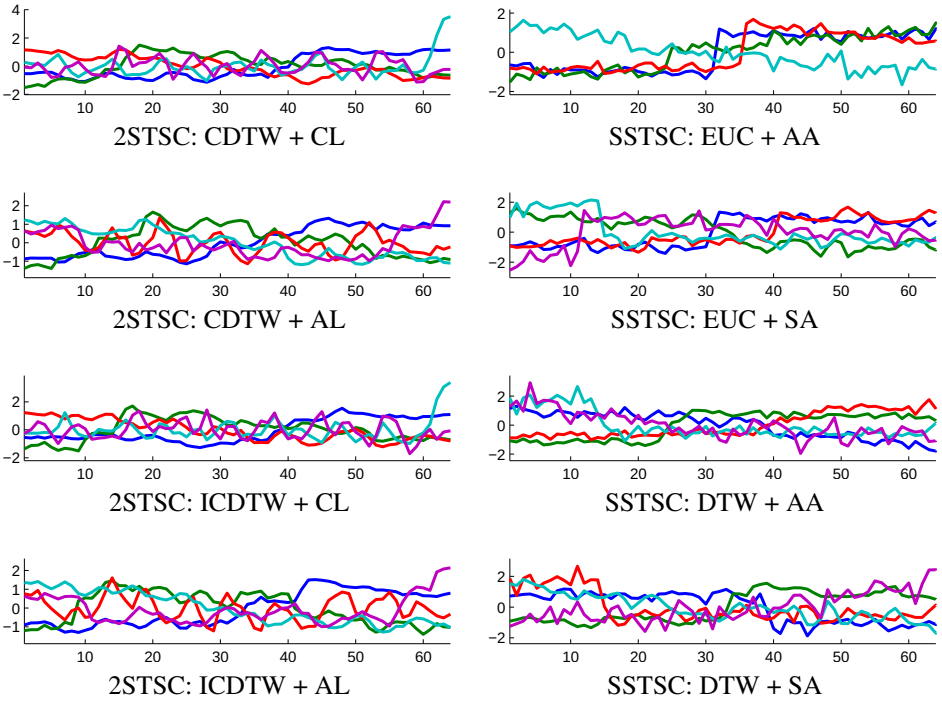

Figure A.31: Cluster representatives of CBF dataset from variations of 2STSC (left) and SSTSC (right) with $k = 5$ and $w = 64$.

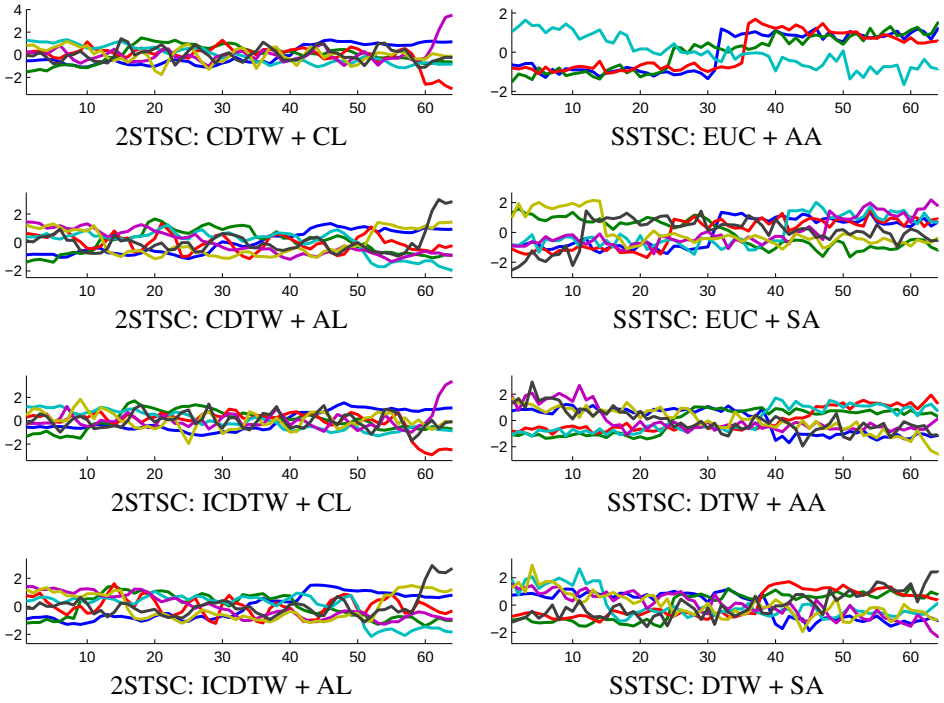

Figure A.32: Cluster representatives of CBF dataset from variations of 2STSC (left) and SSTSC (right) with $k = 7$ and $w = 64$.

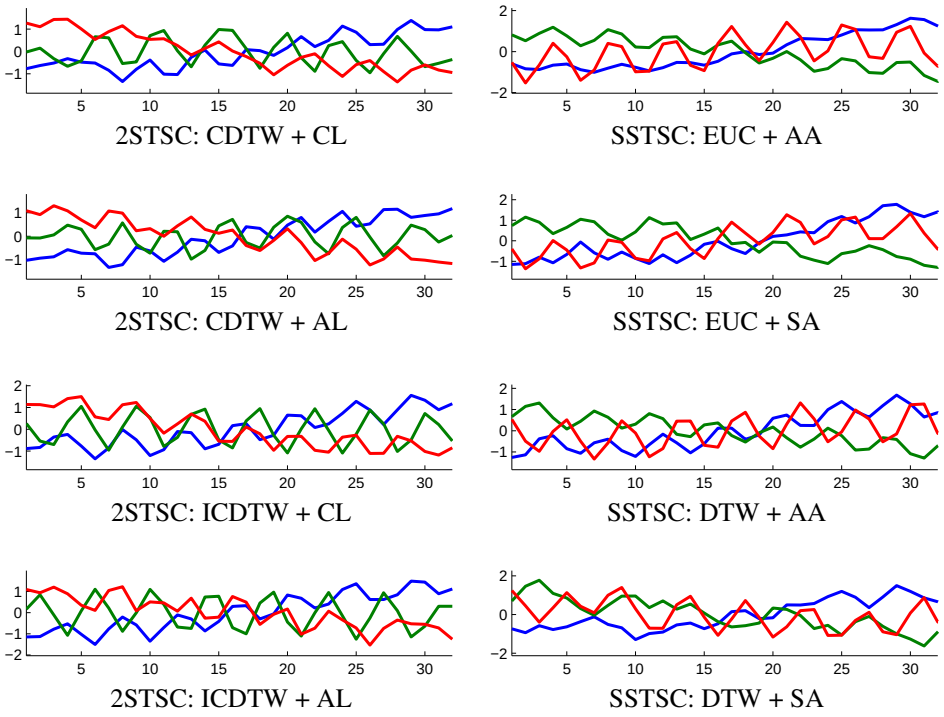

Figure A.33: Cluster representatives of ERP dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 32$.

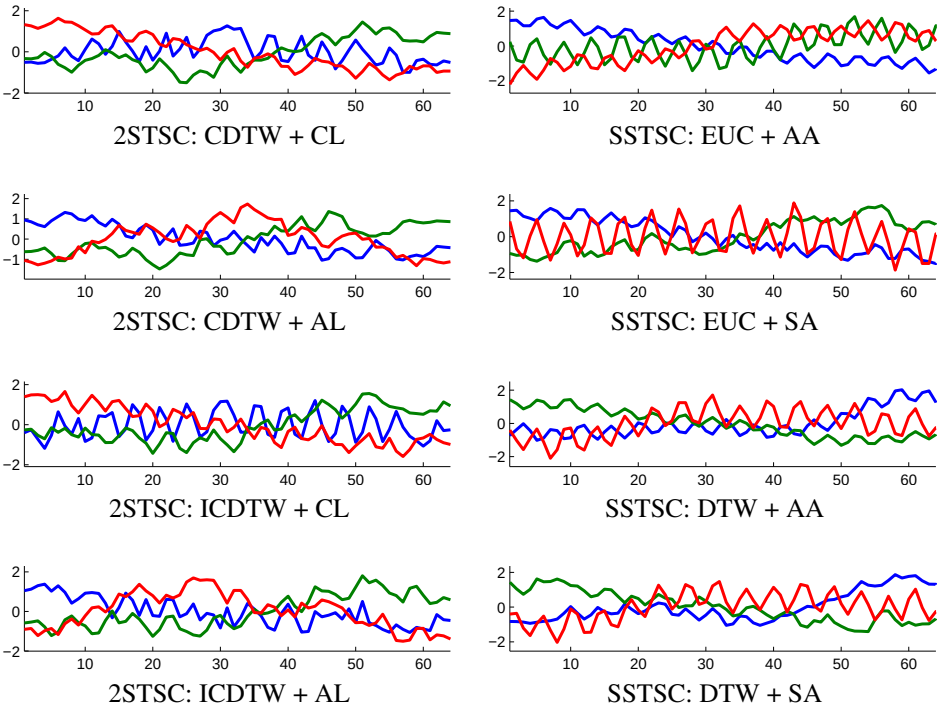

Figure A.34: Cluster representatives of ERP dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 64$.

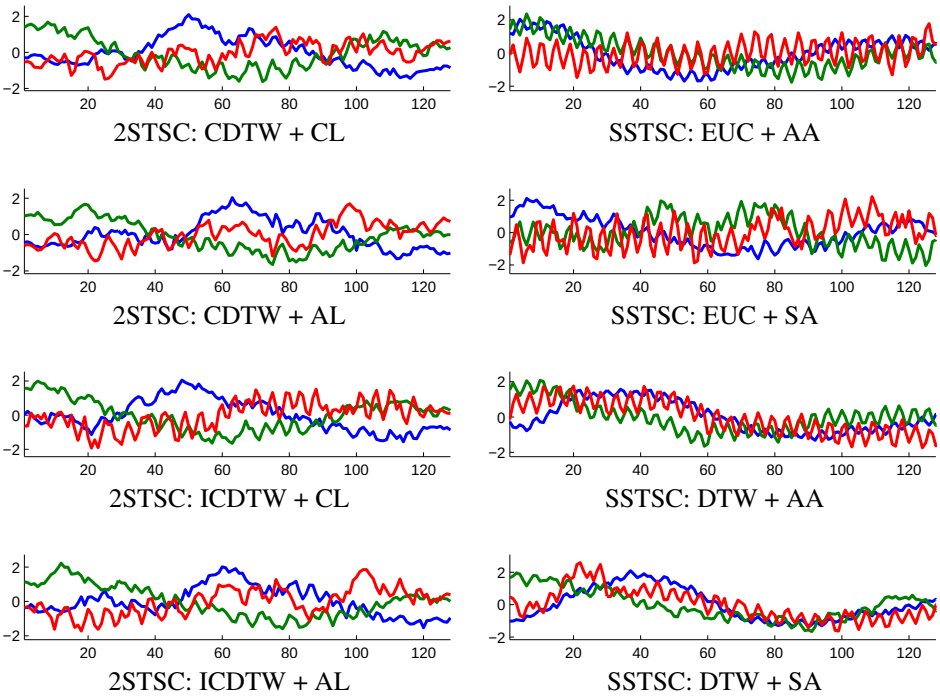

Figure A.35: Cluster representatives of ERP dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 128$.

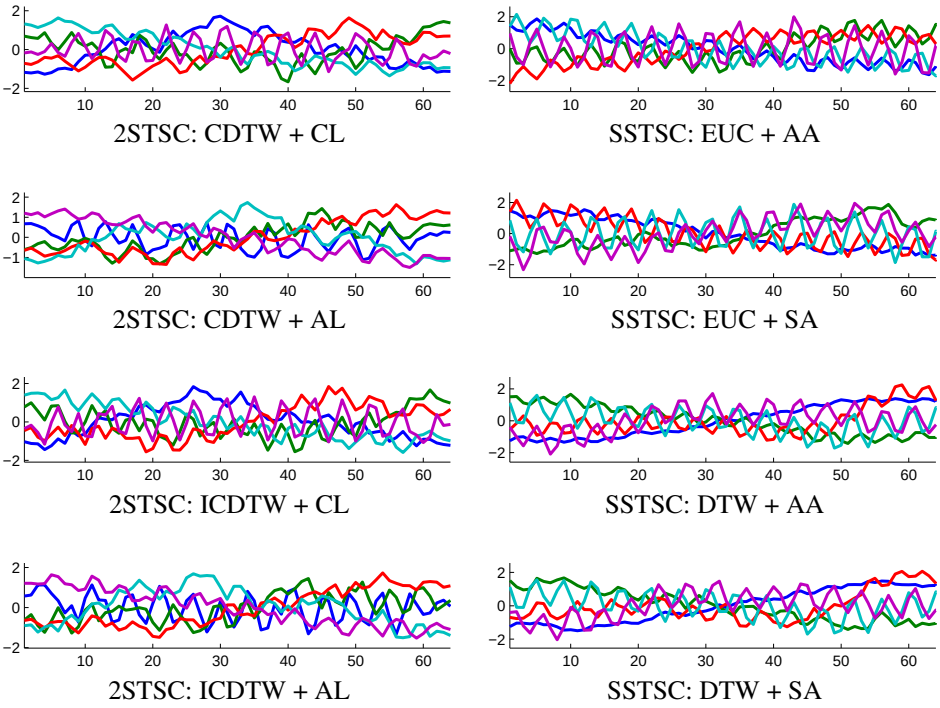

Figure A.36: Cluster representatives of ERP dataset from variations of 2STSC (left) and SSTSC (right) with $k = 5$ and $w = 64$.

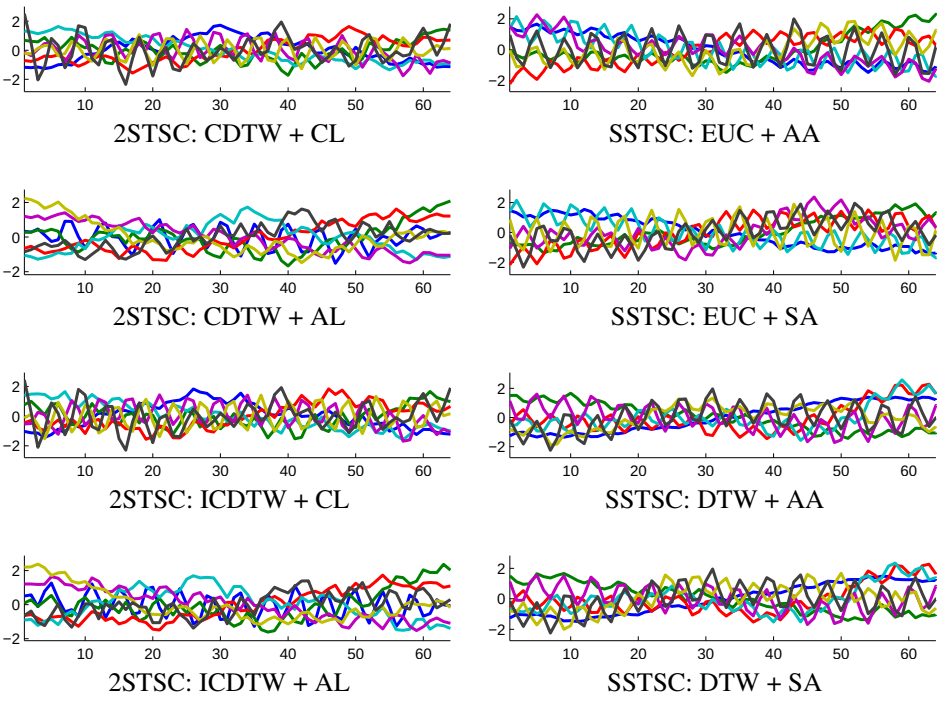

Figure A.37: Cluster representatives of ERP dataset from variations of 2STSC (left) and SSTSC (right) with $k = 7$ and $w = 64$.

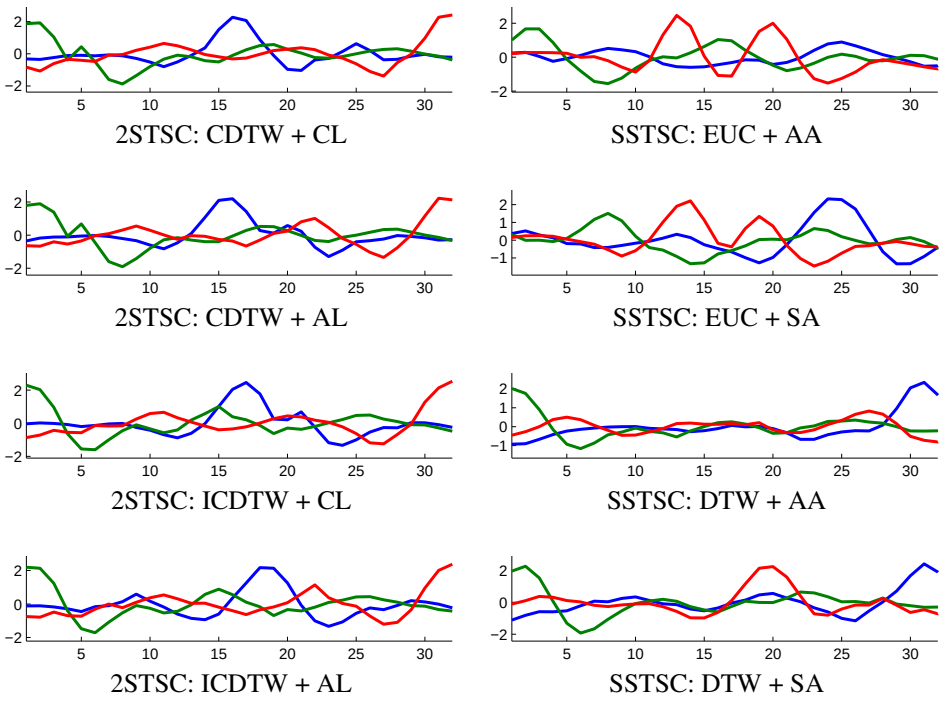

Figure A.38: Cluster representatives of Field4 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 32$.

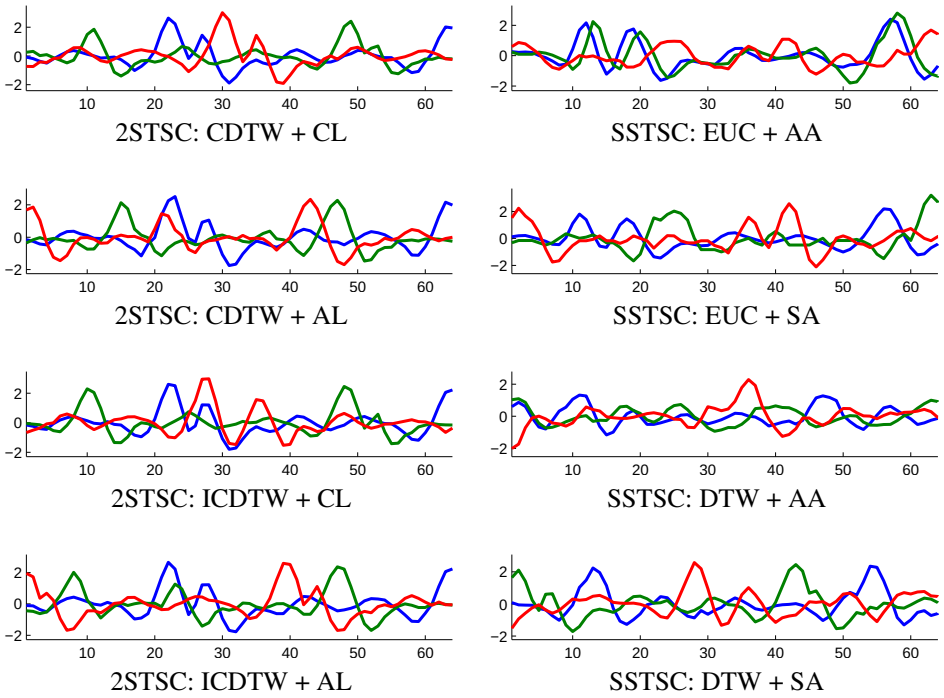

Figure A.39: Cluster representatives of Field4 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 64$.

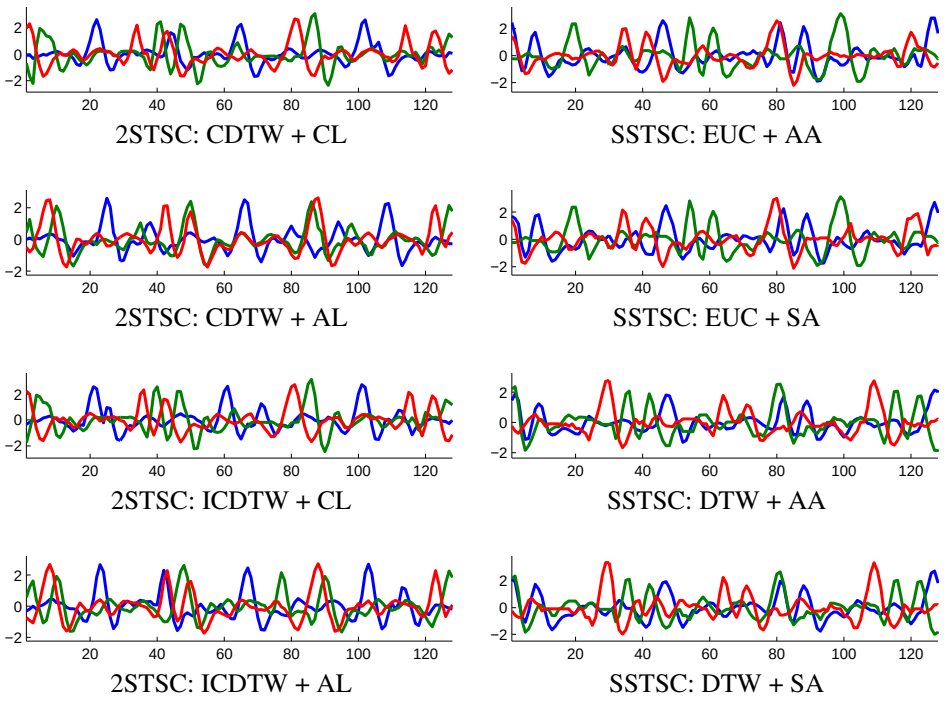

Figure A.40: Cluster representatives of Field4 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 128$.

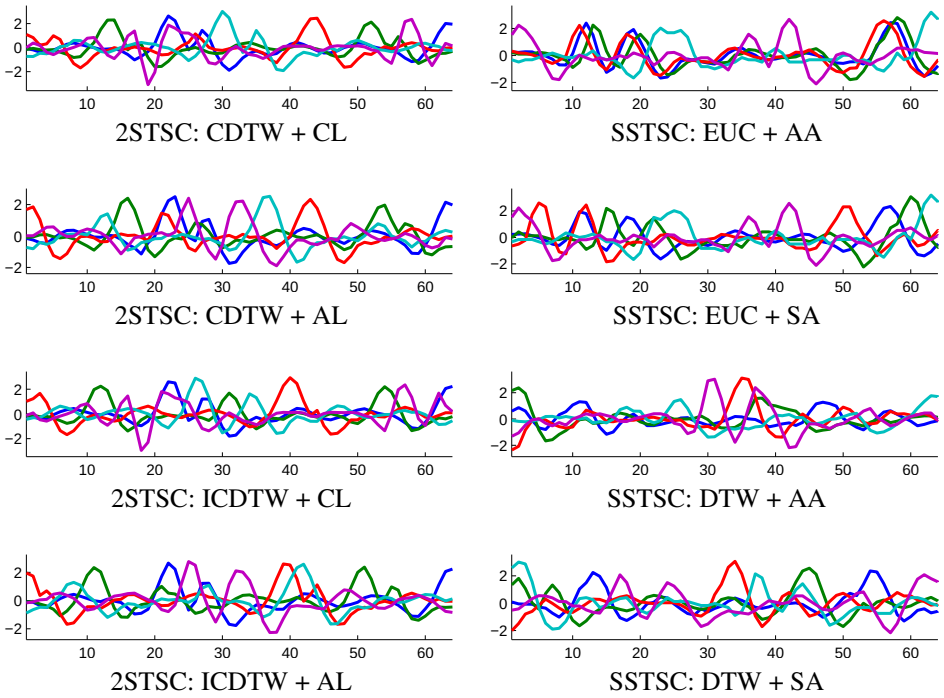

Figure A.41: Cluster representatives of Field4 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 5$ and $w = 64$.

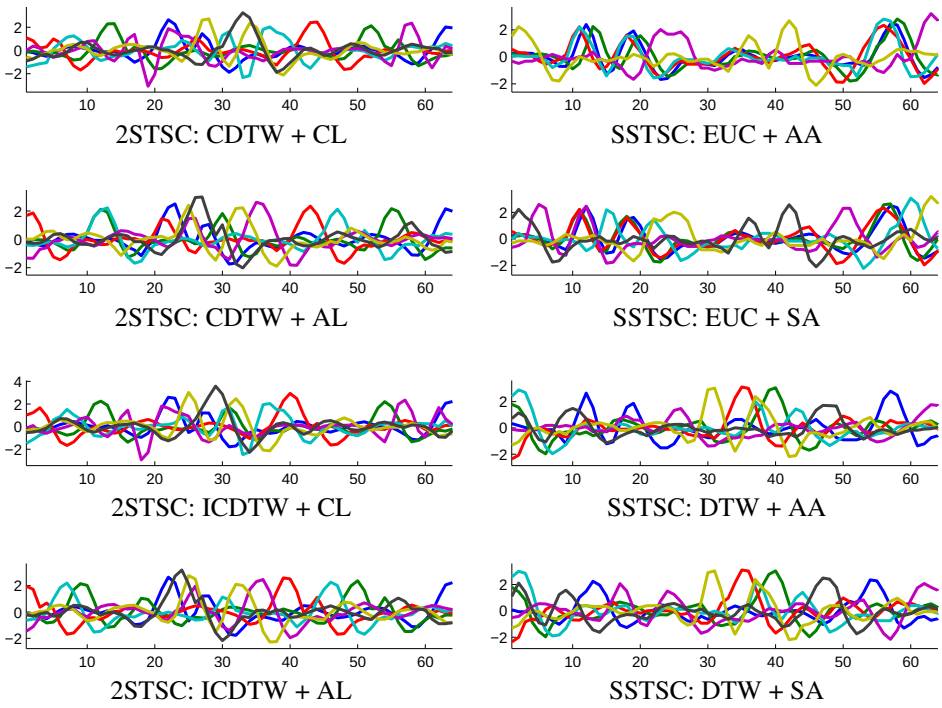

Figure A.42: Cluster representatives of Field4 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 7$ and $w = 64$.

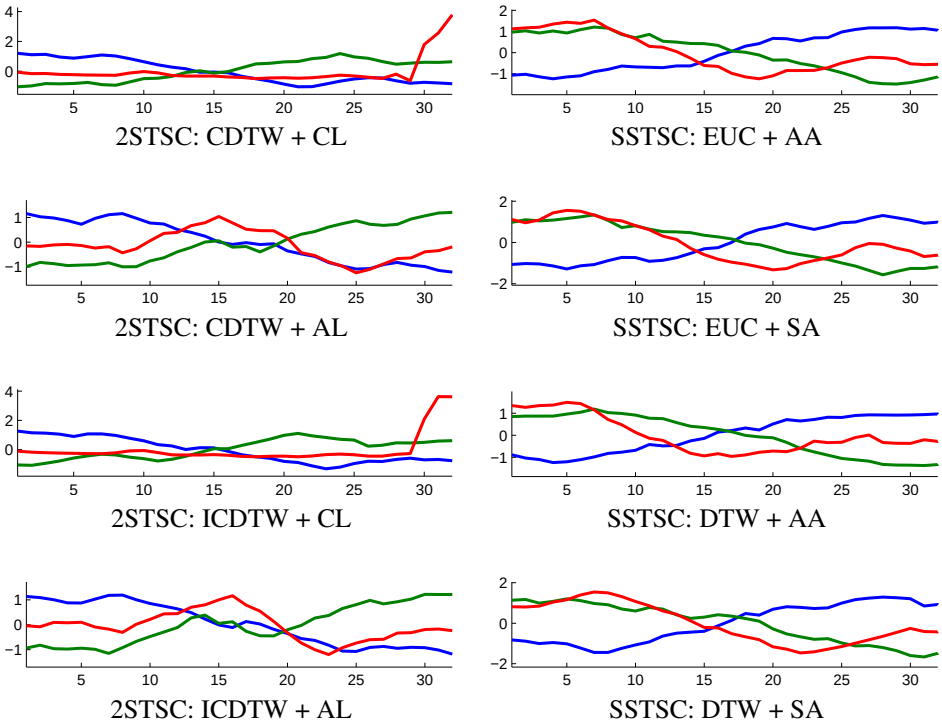

Figure A.43: Cluster representatives of Fortune5004 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 32$.

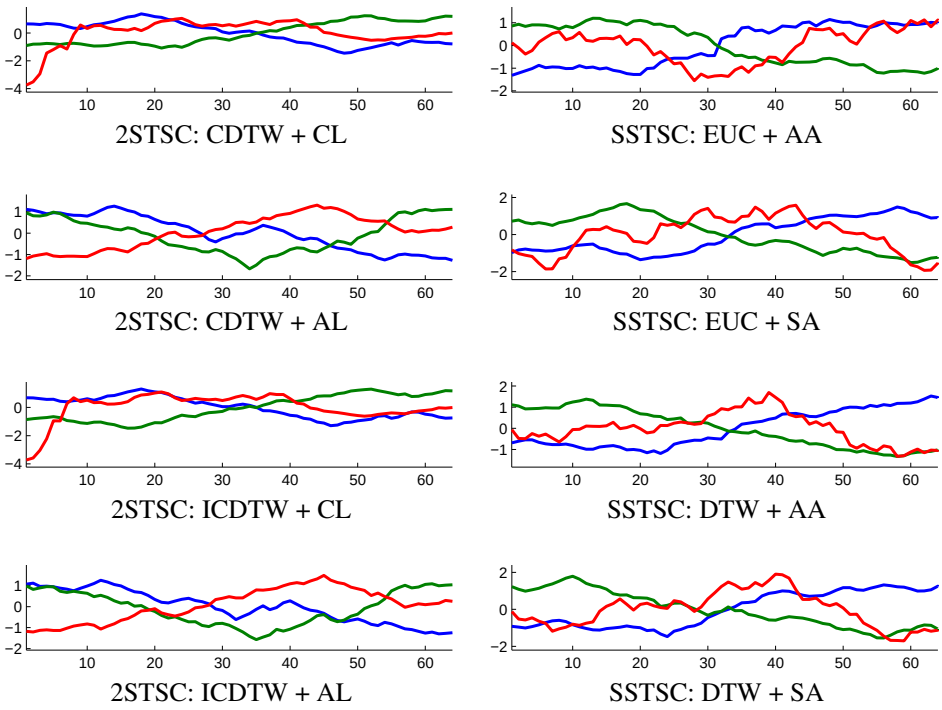

Figure A.44: Cluster representatives of Fortune5004 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 64$.

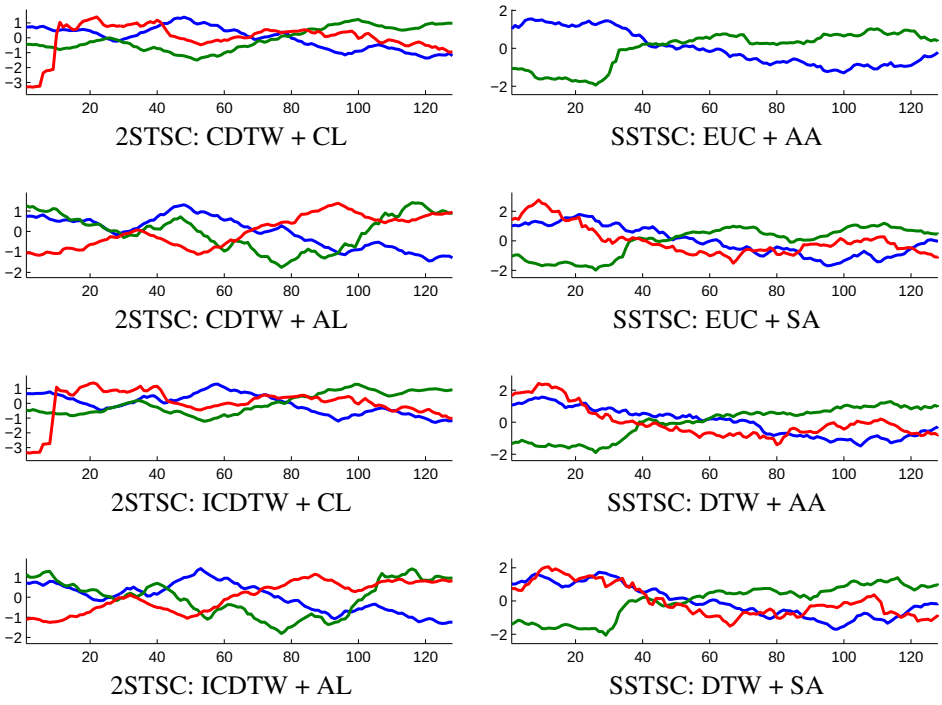

Figure A.45: Cluster representatives of Fortune5004 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 128$.

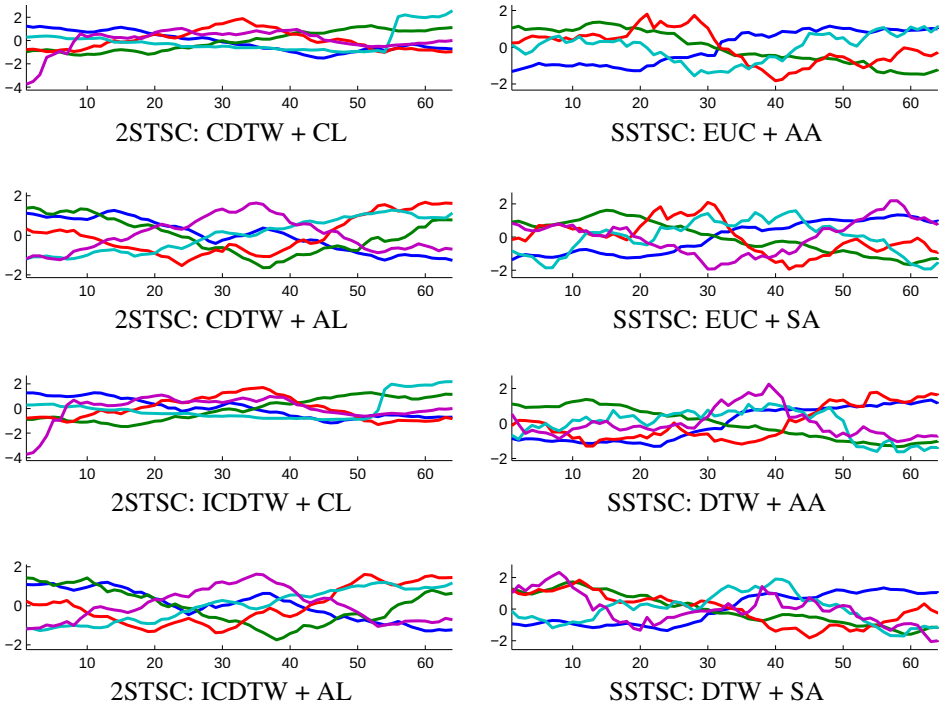

Figure A.46: Cluster representatives of Fortune5004 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 5$ and $w = 64$.

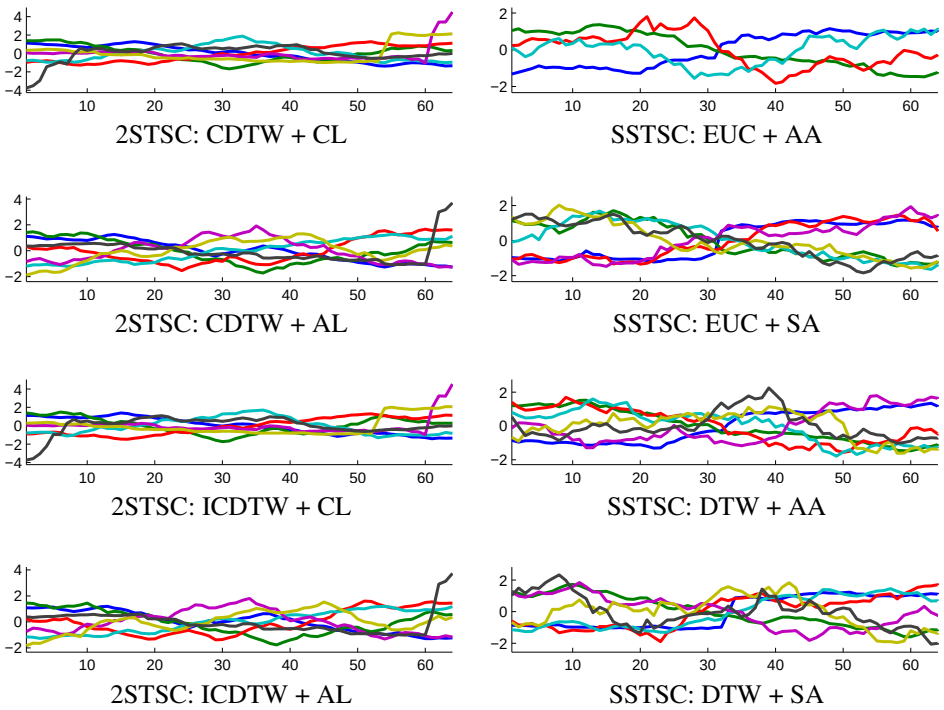

Figure A.47: Cluster representatives of Fortune5004 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 7$ and $w = 64$.

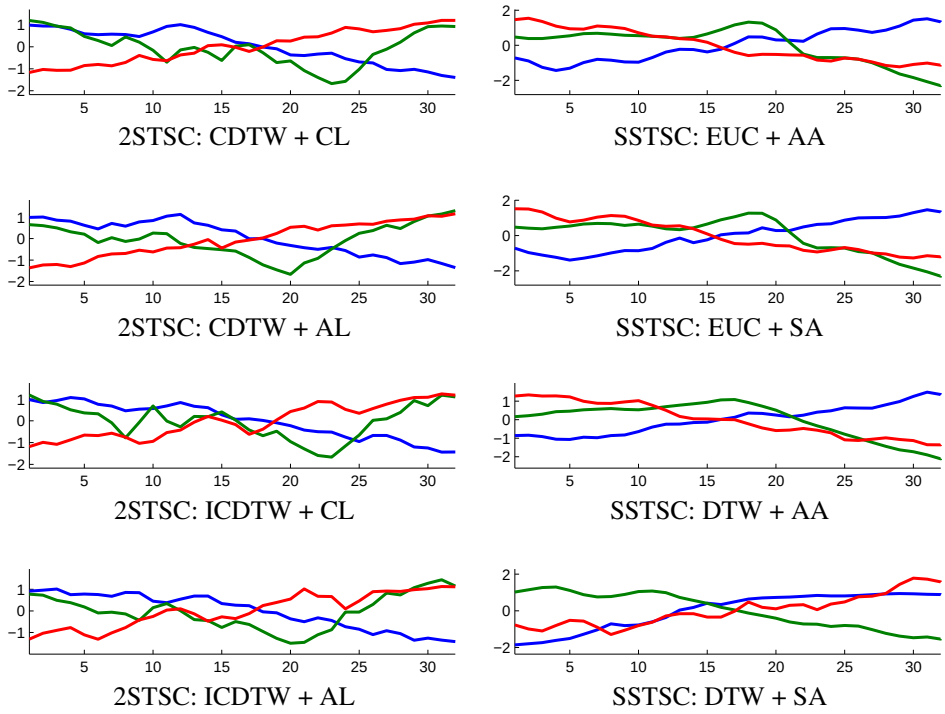

Figure A.48: Cluster representatives of MITDBX108 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 32$.

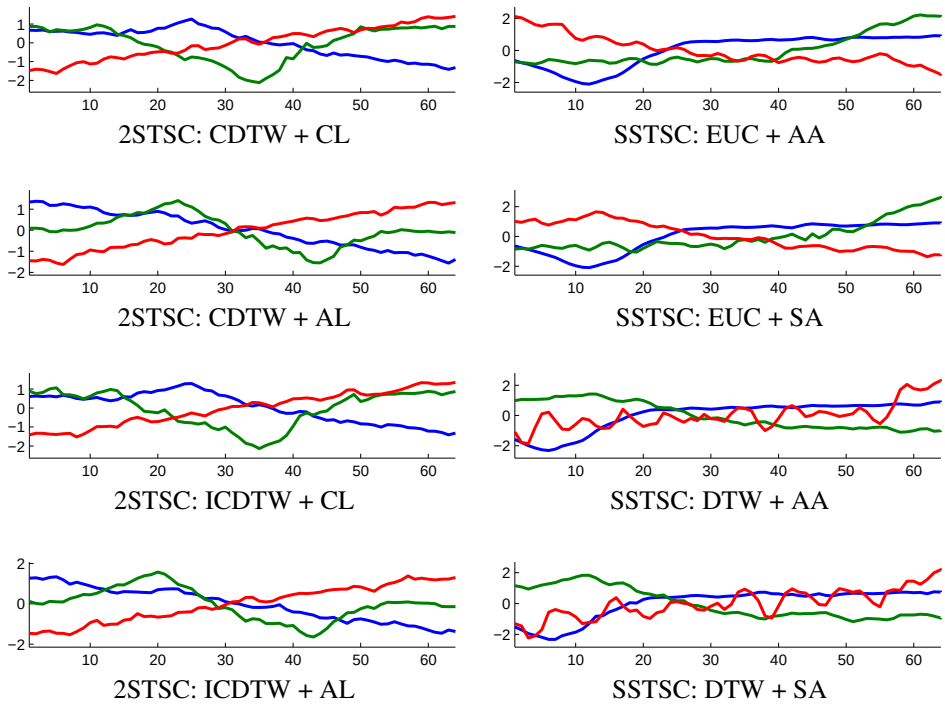

Figure A.49: Cluster representatives of MITDBX108 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 64$.

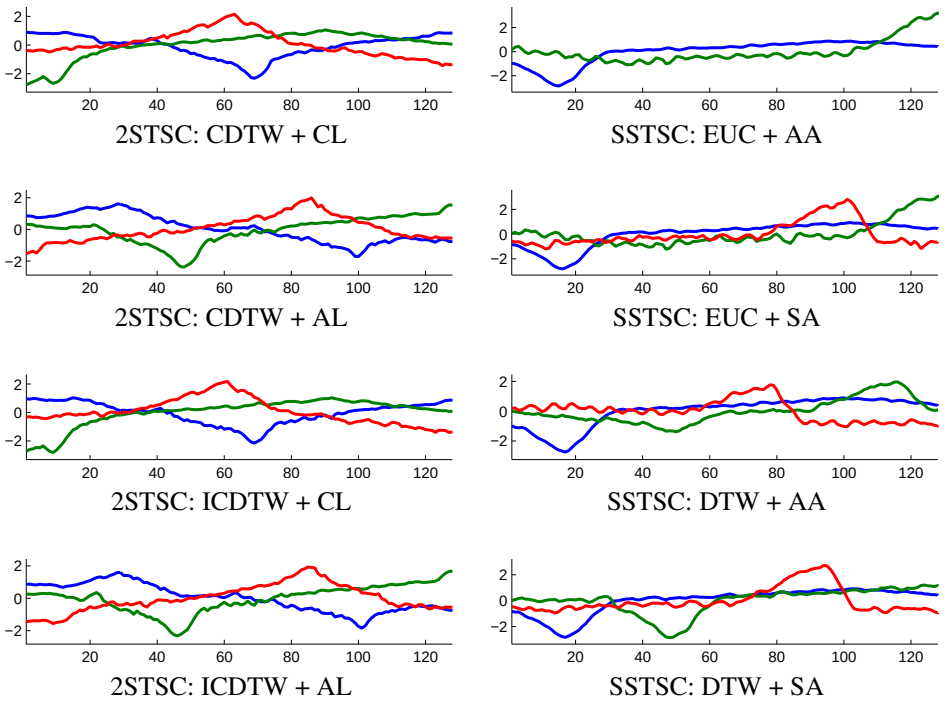

Figure A.50: Cluster representatives of MITDBX108 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 128$.

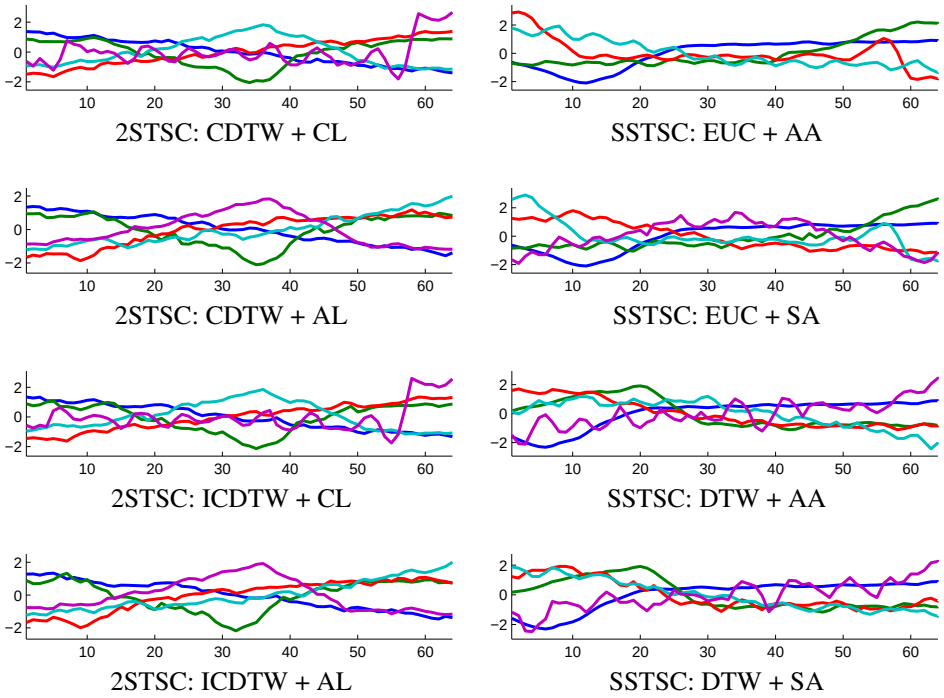

Figure A.51: Cluster representatives of MITDBX108 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 5$ and $w = 64$.

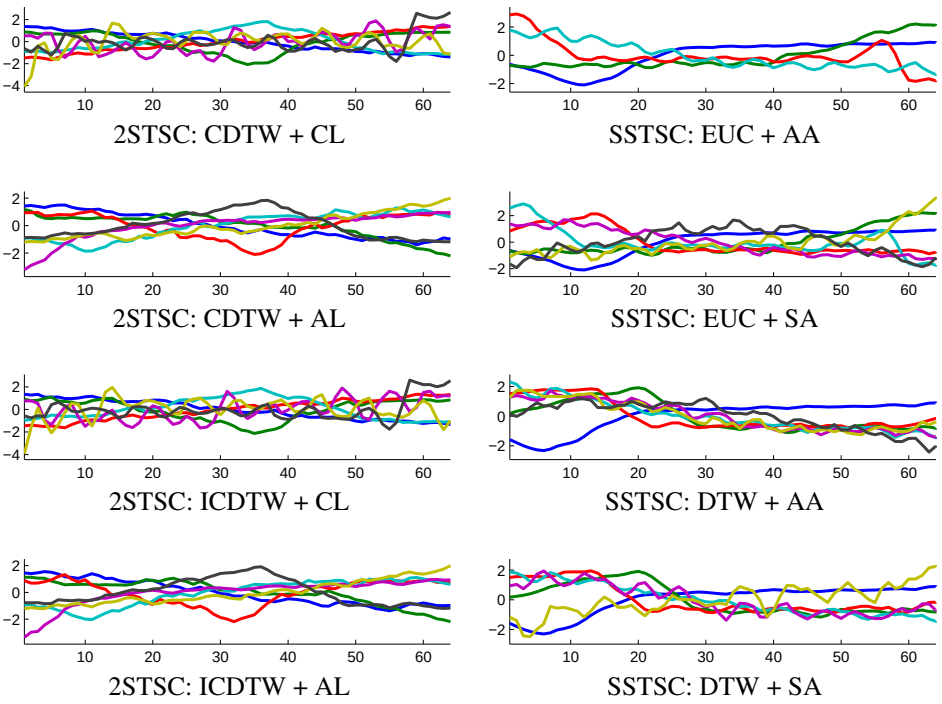

Figure A.52: Cluster representatives of MITDBX108 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 7$ and $w = 64$.

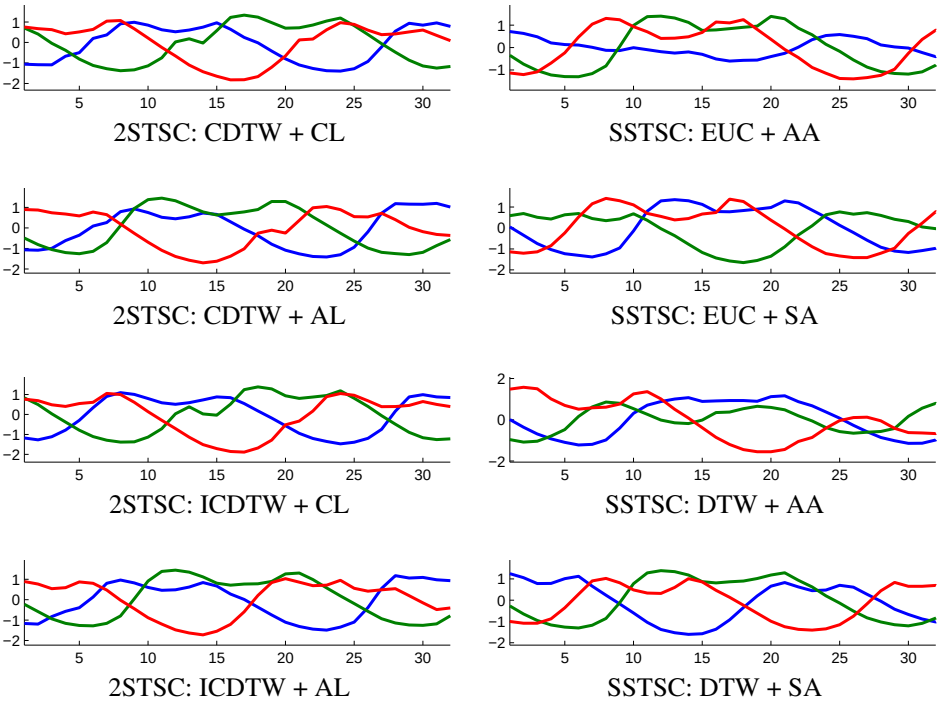

Figure A.53: Cluster representatives of TOR96 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 32$.

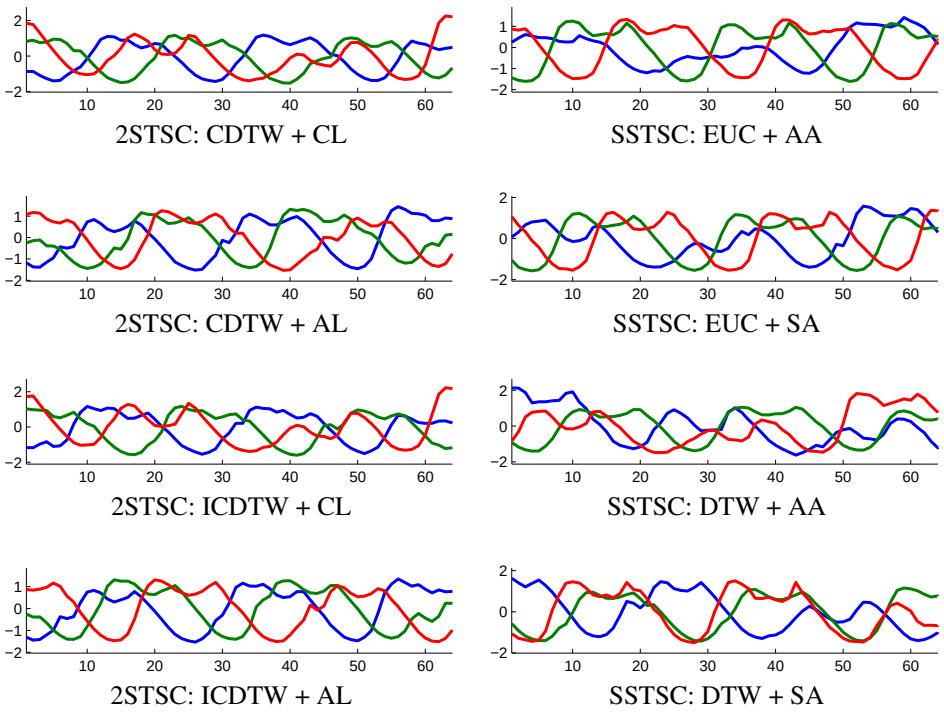

Figure A.54: Cluster representatives of TOR96 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 64$.

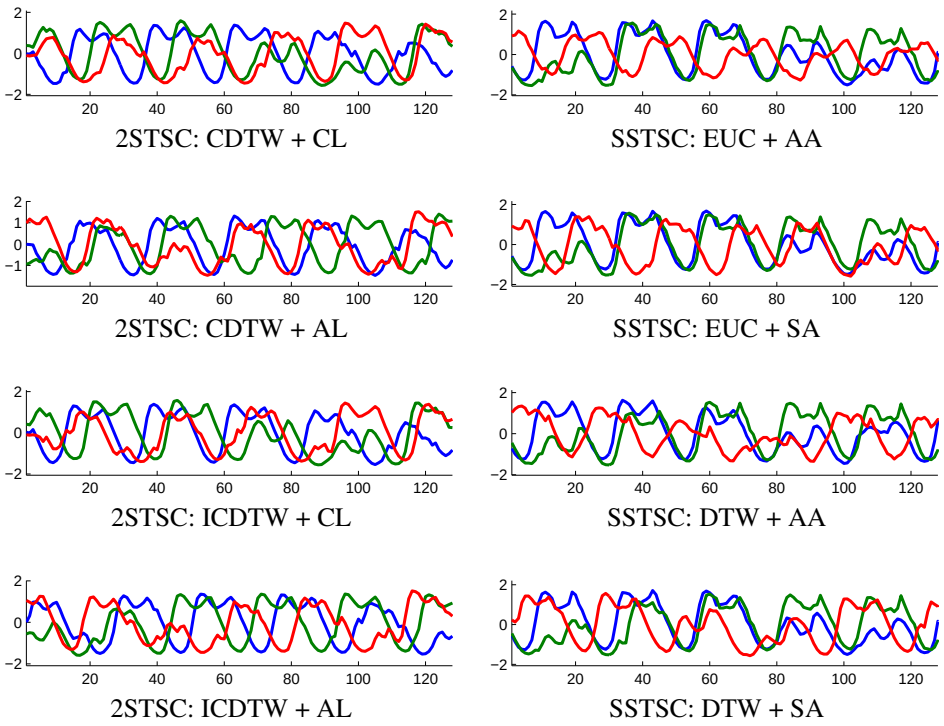

Figure A.55: Cluster representatives of TOR96 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 3$ and $w = 128$.

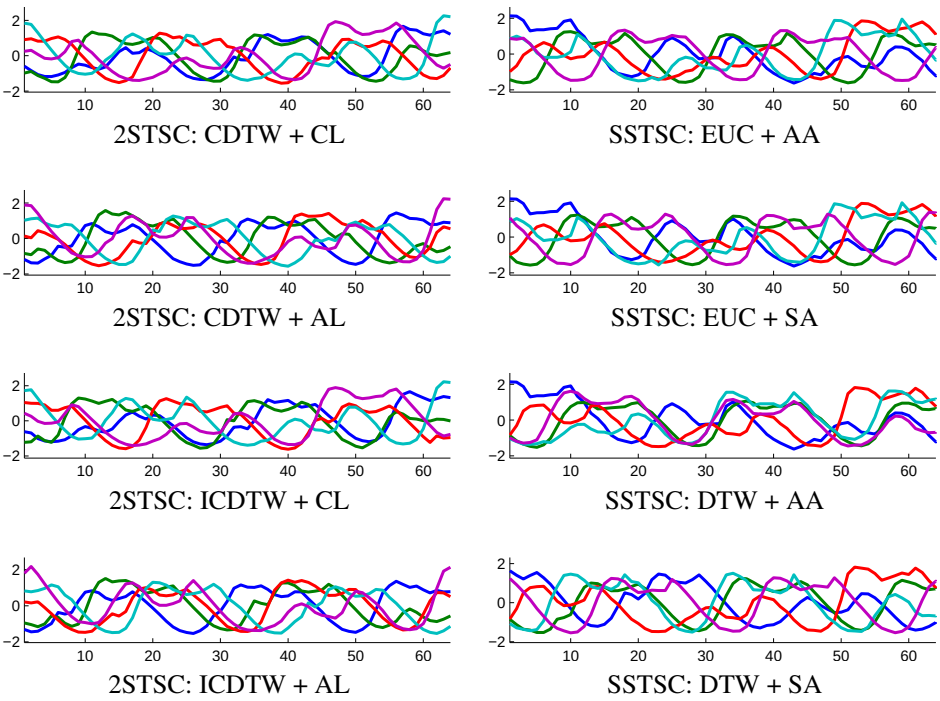

Figure A.56: Cluster representatives of TOR96 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 5$ and $w = 64$.

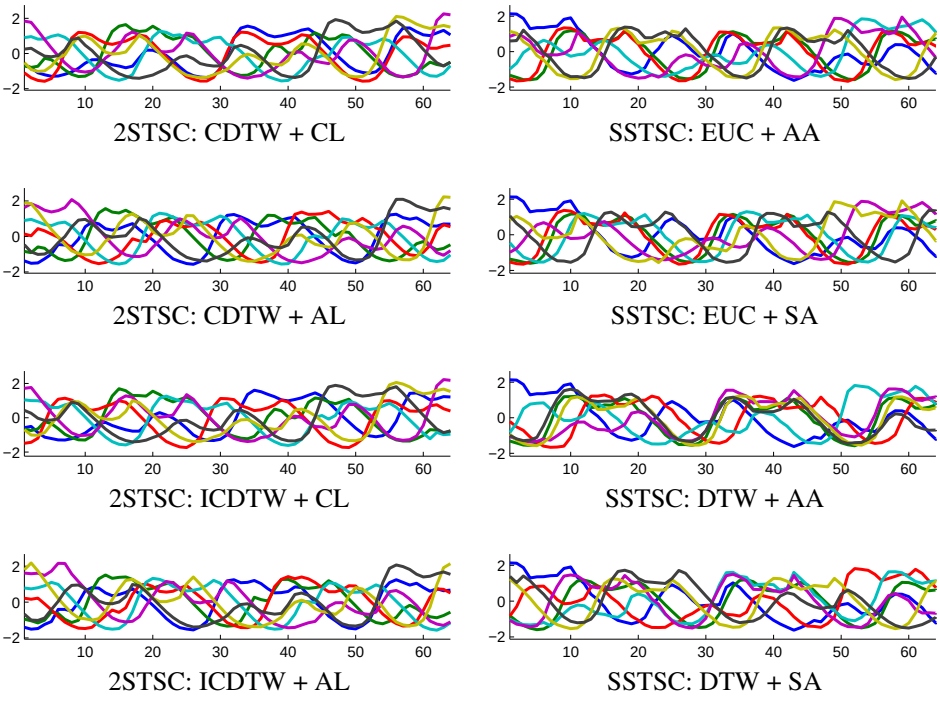

Figure A.57: Cluster representatives of TOR96 dataset from variations of 2STSC (left) and SSTSC (right) with $k = 7$ and $w = 64$.

APPENDICES B

COMPLETE EXPERIMENTAL RESULTS OF THE EXPERIMENT IN SECTION 3.5.4 WHEN SCALING FACTOR IS SET TO 1

Table B.1: Rand Index (RI) from all algorithms on all datasets when the scaling factor (f) is 1 and number of clusters (k) is set to the number of classes in each dataset

Dataset	Rand Index											
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L		D-AA-Z		D-SA-Z	
	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.49	0.52	0.49	0.49	0.49	0.49	0.49	0.49	0.50	**0.56**	0.48	0.53
SonyAIBORobotSurfaceII	0.80	0.78	0.73	0.72	0.73	0.59	**0.81**	0.72	0.48	0.46	0.56	0.57
SonyAIBORobotSurface	0.48	0.49	0.60	0.54	0.77	0.75	0.60	0.56	0.46	0.50	0.47	**1.00**
DistalPhalanxOutlineCorrect	**0.49**	0.48	**0.49**	**0.49**	**0.49**	0.48	**0.49**	**0.49**	**0.49**	0.48	**0.49**	0.48
MiddlePhalanxOutlineCorrect	0.48	0.48	0.49	0.49	0.48	0.48	0.49	0.49	**0.50**	**0.50**	**0.50**	**0.50**
PhalangesOutlinesCorrect	0.48	0.48	**0.49**	**0.49**	**0.49**	**0.49**	0.47	0.47	0.48	0.48	**0.49**	0.49
ProximalPhalanxOutlineCorrect	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**
DistalPhalanxOutlineAgeGroup	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.62	0.62	**0.80**	**0.80**
MiddlePhalanxOutlineAgeGroup	**0.60**	**0.60**	**0.60**	**0.60**	**0.60**	**0.60**	**0.60**	**0.60**	0.45	0.45	**0.60**	**0.60**
ProximalPhalanxOutlineAgeGroup	**0.65**	**0.65**	**0.65**	**0.65**	**0.65**	**0.65**	**0.65**	**0.65**	0.62	0.62	0.62	0.62
TwoLeadECG	**0.61**	**0.61**	0.56	0.56	0.52	0.52	0.56	0.56	0.49	0.48	0.47	0.47
MoteStrain	0.52	0.47	0.49	0.49	0.49	0.47	0.49	0.46	0.50	0.46	0.68	**1.00**
ECG200	0.52	0.54	0.56	0.56	0.49	0.51	0.56	0.59	0.52	**0.60**	0.48	0.48
CBF	0.61	0.60	0.60	0.62	0.61	0.48	0.53	0.39	0.69	**0.73**	0.63	0.39
Two_Patterns	0.65	**1.00**	0.62	**1.00**	0.58	0.00	0.59	0.80	0.57	0.63	0.81	0.91
ECGFiveDays	**0.54**	**0.54**	**0.54**	**0.54**	0.47	0.47	0.47	0.47	0.47	0.47	0.49	0.50
ECG5000	0.78	0.79	0.81	0.81	0.63	0.59	0.80	0.80	0.79	**0.96**	0.75	0.82
Gun_Point	0.52	0.50	**0.61**	0.59	0.49	0.49	**0.61**	0.59	0.52	0.52	0.52	0.52
wafer	0.65	0.66	0.56	0.58	0.72	**0.77**	0.49	0.50	0.49	0.51	0.49	0.50
ChlorineConcentration	0.60	0.60	**0.60**	**0.60**	0.60	0.60	0.51	0.51	0.56	0.56	0.54	0.54
Wine	0.61	0.61	**0.73**	**0.73**	0.61	0.61	**0.73**	**0.73**	0.61	0.61	0.61	0.61
Strawberry	0.49	0.49	**0.56**	**0.56**	0.49	0.49	0.49	0.49	0.49	0.49	**0.56**	**0.56**
ArrowHead	**0.66**	**0.66**	**0.66**	**0.66**	0.63	0.61	0.64	0.64	0.46	0.43	0.53	0.52
Trace	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.69	0.81	**0.84**	**0.84**
ToeSegmentation1	0.49	**0.67**	0.47	0.47	0.45	0.50	0.49	0.45	0.49	0.44	0.48	0.56
Coffee	0.81	0.81	0.73	0.73	0.81	0.81	0.81	0.81	**0.90**	**0.90**	**0.90**	**0.90**
ToeSegmentation2	0.50	**0.80**	0.49	0.47	0.47	0.46	0.56	0.54	0.51	0.52	0.49	0.49
FaceFour	0.65	0.33	0.76	0.71	0.53	**1.00**	0.69	0.65	0.60	0.33	0.67	0.17
yoga	**0.52**	0.51	0.48	0.48	**0.52**	0.51	0.48	0.47	-	-	-	-
Ham	0.47	0.47	0.48	0.48	0.48	**0.50**	0.49	0.49	-	-	-	-
Meat	**0.94**	**0.94**	0.81	0.81	0.81	0.81	0.68	0.68	-	-	-	-
Beef	**0.76**	**0.76**	**0.76**	**0.76**	0.69	0.69	0.70	0.70	-	-	-	-
FordA	0.56	**1.00**	0.52	0.43	0.49	0.40	0.53	0.52	-	-	-	-
FordB	0.52	**1.00**	0.48	0.44	0.51	0.71	0.48	0.47	-	-	-	-
ShapeletSim	0.56	**1.00**	0.46	**1.00**	0.49	0.50	0.47	0.43	-	-	-	-
BeetleFly	0.45	0.33	0.48	0.46	0.47	0.50	0.49	**0.51**	-	-	-	-
BirdChicken	0.47	0.47	**0.48**	**0.48**	0.47	0.47	0.47	0.47	-	-	-	-
Earthquakes	0.59	**1.00**	0.59	0.50	0.52	**1.00**	0.48	0.50	-	-	-	-
Herring	**0.49**	**0.49**	**0.49**	**0.49**	**0.49**	**0.49**	**0.49**	**0.49**	-	-	-	-
OliveOil	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	-	-	-	-
Car	**0.73**	**0.73**	**0.73**	**0.73**	0.70	0.70	**0.73**	**0.73**	-	-	-	-
Lighting2	0.47	0.33	0.51	**1.00**	0.52	0.00	0.47	0.60	-	-	-	-
Computers	0.48	0.45	0.47	0.47	**0.51**	0.43	0.47	0.45	-	-	-	-
LargeKitchenAppliances	0.50	0.54	0.43	0.38	0.54	**0.64**	0.50	0.56	-	-	-	-
RefrigerationDevices	0.51	**1.00**	0.45	0.38	0.50	0.00	0.38	0.44	-	-	-	-
ScreenType	0.53	0.40	0.58	0.56	0.55	**0.81**	0.57	0.57	-	-	-	-
SmallKitchenAppliances	**0.58**	0.33	0.54	0.52	0.54	0.50	0.47	0.43	-	-	-	-
WormsTwoClass	0.56	**0.60**	0.50	0.50	0.52	0.47	0.47	0.44	-	-	-	-
Worms	0.64	**1.00**	0.72	0.67	0.67	**1.00**	0.61	0.53	-	-	-	-
StarLightCurves	**0.81**	**0.81**	**0.81**	**0.81**	0.56	0.68	**0.81**	**0.81**	-	-	-	-
Haptics	0.67	0.67	**0.73**	**0.73**	0.60	0.50	0.67	0.67	-	-	-	-
CinC_ECG_torso	0.54	0.60	**0.62**	**0.62**	0.53	0.59	0.56	0.54	-	-	-	-
HandOutlines	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**	-	-	-	-

Table B.2: Precision from all algorithms on all datasets when the scaling factor (f) is 1 and number of clusters (k) is set to the number of classes in each dataset

Dataset	Precision											
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L		D-AA-Z		D-SA-Z	
	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.48	0.49	0.47	0.47	0.48	0.47	0.48	0.48	0.47	**0.55**	0.47	0.50
SonyAIBORobotSurfaceII	0.77	0.73	0.70	0.67	0.70	0.54	**0.79**	0.67	0.44	0.44	0.53	0.54
SonyAIBORobotSurface	0.47	0.61	0.60	0.67	0.77	0.76	0.57	0.56	0.50	**1.00**	0.48	**1.00**
DistalPhalanxOutlineCorrect	**0.48**	0.48	**0.48**	**0.48**	**0.48**	0.48	**0.48**	**0.48**	**0.48**	0.48	**0.48**	0.48
MiddlePhalanxOutlineCorrect	0.46	0.46	0.47	0.47	0.46	0.46	0.47	0.47	**0.48**	**0.48**	**0.48**	**0.48**
PhalangesOutlinesCorrect	0.45	0.45	0.47	0.47	0.47	0.47	0.45	0.45	0.45	0.45	**0.48**	0.48
ProximalPhalanxOutlineCorrect	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**
DistalPhalanxOutlineAgeGroup	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.41	0.41	**0.66**	**0.66**
MiddlePhalanxOutlineAgeGroup	**0.39**	**0.39**	**0.39**	**0.39**	**0.39**	**0.39**	**0.39**	**0.39**	0.31	0.31	**0.39**	**0.39**
ProximalPhalanxOutlineAgeGroup	**0.45**	**0.45**	**0.45**	**0.45**	**0.45**	**0.45**	**0.45**	**0.45**	0.42	0.42	0.42	0.42
TwoLeadECG	**0.58**	**0.58**	0.53	0.53	0.49	0.49	0.53	0.53	0.48	0.48	0.46	0.46
MoteStrain	0.50	0.47	0.47	0.46	0.47	0.46	0.48	0.46	0.48	0.46	0.63	**1.00**
ECG200	0.50	0.51	0.52	0.52	0.47	0.49	0.53	0.56	0.49	**0.57**	0.45	0.45
CBF	0.38	0.33	0.36	0.46	0.37	0.40	0.28	0.29	0.50	**1.00**	0.42	0.45
Two_Patterns	0.22	**1.00**	0.19	0.00	0.26	0.00	0.22	0.33	0.28	0.29	0.53	0.68
ECGFiveDays	**0.52**	**0.52**	**0.52**	**0.52**	0.46	0.46	0.46	0.46	0.45	0.44	0.47	0.47
ECG5000	0.37	0.43	0.44	0.44	0.19	0.23	0.40	0.40	0.42	**0.88**	0.33	0.75
Gun_Point	0.50	0.49	**0.57**	0.55	0.47	0.47	**0.57**	0.55	0.49	0.49	0.49	0.49
wafer	0.63	0.64	0.52	0.54	0.69	**0.75**	0.47	0.48	0.47	0.49	0.47	0.47
ChlorineConcentration	0.33	0.33	**0.36**	**0.36**	0.33	0.33	0.33	0.33	0.29	0.29	0.30	0.30
Wine	0.57	0.57	**0.70**	**0.70**	0.57	0.57	**0.70**	**0.70**	0.57	0.57	0.57	0.57
Strawberry	0.48	0.48	**0.53**	**0.53**	0.48	0.48	0.48	0.48	0.48	0.48	**0.53**	**0.53**
ArrowHead	**0.45**	**0.45**	**0.45**	**0.45**	0.42	0.42	0.43	0.43	0.33	0.33	0.34	0.34
Trace	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.38	0.53	**0.58**	**0.58**
ToeSegmentation1	0.45	**0.86**	0.44	0.42	0.47	0.50	0.48	0.45	0.46	0.44	0.45	0.52
Coffee	0.79	0.79	0.70	0.70	0.79	0.79	0.79	0.79	**0.89**	**0.89**	**0.89**	**0.89**
ToeSegmentation2	0.48	**0.75**	0.48	0.46	0.45	0.44	0.53	0.52	0.48	0.47	0.47	0.46
FaceFour	0.29	0.33	0.45	0.50	0.19	**1.00**	0.33	0.43	0.15	0.00	0.31	0.17
yoga	**0.50**	0.49	0.46	0.46	0.49	0.48	0.47	0.46	-	-	-	-
Ham	0.45	0.44	0.45	0.45	0.45	**0.47**	0.47	0.47	-	-	-	-
Meat	**0.89**	**0.89**	0.67	0.67	0.65	0.65	0.47	0.47	-	-	-	-
Beef	**0.33**	**0.33**	**0.33**	**0.33**	0.19	0.19	0.22	0.22	-	-	-	-
FordA	0.52	**1.00**	0.49	0.38	0.46	0.40	0.50	0.52	-	-	-	-
FordB	0.49	**1.00**	0.46	0.46	0.48	0.67	0.48	0.47	-	-	-	-
ShapeletSim	0.52	**1.00**	0.45	0.00	0.45	0.33	0.46	0.43	-	-	-	-
BeetleFly	0.47	0.33	0.45	0.46	0.43	**0.50**	0.47	0.48	-	-	-	-
BirdChicken	0.45	0.45	**0.46**	**0.46**	0.45	0.44	0.46	0.46	-	-	-	-
Earthquakes	0.57	0.00	0.57	0.33	0.47	**1.00**	0.48	0.50	-	-	-	-
Herring	**0.48**	**0.48**	**0.48**	**0.48**	**0.48**	**0.48**	**0.48**	**0.48**	-	-	-	-
OliveOil	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	-	-	-	-
Car	**0.39**	**0.39**	**0.39**	**0.39**	0.35	0.35	**0.39**	**0.39**	-	-	-	-
Lighting2	0.44	0.00	0.50	**1.00**	0.47	0.00	0.46	0.60	-	-	-	-
Computers	0.47	0.52	0.45	0.46	0.50	**0.69**	0.45	0.45	-	-	-	-
LargeKitchenAppliances	0.24	0.38	0.28	0.33	0.36	**0.45**	0.29	0.43	-	-	-	-
RefrigerationDevices	0.21	0.00	**0.31**	0.31	0.30	0.00	0.30	0.27	-	-	-	-
ScreenType	0.24	0.00	0.31	0.38	0.27	**0.57**	0.35	0.40	-	-	-	-
SmallKitchenAppliances	**0.34**	0.33	0.26	0.17	0.32	0.25	0.29	0.23	-	-	-	-
WormsTwoClass	0.52	**0.75**	0.47	0.45	0.47	0.43	0.45	0.44	-	-	-	-
Worms	0.08	0.00	0.23	**0.25**	0.11	0.00	0.23	0.19	-	-	-	-
StarLightCurves	**0.65**	**0.65**	**0.65**	**0.65**	0.38	0.48	**0.65**	**0.65**	-	-	-	-
Haptics	0.18	0.00	**0.22**	**0.22**	0.10	0.00	0.15	0.15	-	-	-	-
CinC_ECG_torso	0.22	0.24	0.25	0.25	0.22	**0.26**	0.25	0.23	-	-	-	-
HandOutlines	**0.70**	**0.70**	**0.70**	**0.70**	**0.70**	**0.70**	**0.70**	**0.70**	-	-	-	-

Table B.3: Recall from all algorithms on all datasets when the scaling factor (f) is 1 and number of clusters (k) is set to the number of classes in each dataset

Dataset	Recall											
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L		D-AA-Z		D-SA-Z	
	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.67	0.87	0.56	0.56	0.67	0.84	0.82	0.82	0.64	0.63	**0.89**	0.86
SonyAIBORobotSurfaceII	0.83	**0.89**	0.77	0.84	0.77	0.81	0.82	0.84	0.44	0.54	0.55	0.54
SonyAIBORobotSurface	0.45	0.48	0.60	0.50	0.77	0.76	0.87	0.56	0.43	0.50	0.48	**1.00**
DistalPhalanxOutlineCorrect	**0.82**	0.48	**0.82**	**0.82**	**0.82**	0.48	**0.82**	**0.82**	**0.82**	0.48	**0.82**	0.48
MiddlePhalanxOutlineCorrect	0.59	0.59	0.56	0.56	0.59	0.59	0.56	0.56	**0.81**	**0.81**	**0.81**	**0.81**
PhalangesOutlinesCorrect	0.46	0.46	0.49	0.49	0.56	0.56	0.47	0.47	0.46	0.46	0.82	**0.90**
ProximalPhalanxOutlineCorrect	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**
DistalPhalanxOutlineAgeGroup	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.62	0.62	**0.67**	**0.67**
MiddlePhalanxOutlineAgeGroup	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	**0.68**	**0.68**	0.59	0.59
ProximalPhalanxOutlineAgeGroup	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**	0.68	0.68	0.68	0.68
TwoLeadECG	0.63	0.63	0.56	0.56	0.50	0.50	0.56	0.56	**0.82**	0.48	0.64	0.64
MoteStrain	0.63	0.47	0.49	0.55	0.49	0.72	0.67	0.46	0.79	0.46	0.73	**1.00**
ECG200	0.63	0.70	**0.73**	**0.73**	0.56	0.60	0.56	0.60	0.50	0.61	0.46	0.44
CBF	0.47	0.50	0.42	0.58	0.33	0.44	0.37	**0.80**	0.55	0.43	0.50	0.31
Two_Patterns	0.31	**1.00**	0.24	0.00	0.50	0.00	0.37	**1.00**	0.65	0.61	0.67	**1.00**
ECGFiveDays	0.53	0.53	0.53	0.53	0.57	0.57	0.57	0.57	0.48	0.57	0.58	**0.64**
ECG5000	0.44	0.47	0.56	0.56	0.40	0.60	0.48	0.48	0.68	**0.88**	0.56	0.43
Gun_Point	0.63	**0.89**	0.72	0.75	0.56	0.88	0.72	0.75	0.54	0.54	0.54	0.54
wafer	0.63	0.69	0.73	**0.84**	0.74	0.75	0.49	0.54	0.49	0.53	0.47	0.50
ChlorineConcentration	0.35	0.35	0.40	0.40	0.35	0.35	**0.60**	**0.60**	0.33	0.33	0.40	0.40
Wine	0.72	0.72	**0.77**	**0.77**	0.72	0.72	**0.77**	**0.77**	0.72	0.72	0.72	0.72
Strawberry	**0.82**	**0.82**	0.62	0.62	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	0.62	0.62
ArrowHead	0.61	0.61	0.61	0.61	0.56	0.74	0.60	0.60	0.77	**0.87**	0.60	0.65
Trace	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	**0.88**	0.85	0.85
ToeSegmentation1	0.51	0.60	0.44	0.43	0.78	0.50	**0.90**	0.45	0.46	0.39	0.46	0.52
Coffee	0.82	0.82	0.77	0.77	0.82	0.82	0.82	0.82	**0.90**	**0.90**	**0.90**	**0.90**
ToeSegmentation2	0.58	**0.86**	0.67	0.55	0.53	0.54	0.53	0.51	0.52	0.53	0.72	0.72
FaceFour	0.45	**1.00**	0.52	0.62	0.36	**1.00**	0.45	0.68	0.15	0.00	0.40	**1.00**
yoga	0.63	0.60	0.59	0.59	0.54	0.52	**0.72**	0.70	-	-	-	-
Ham	0.47	0.44	**0.50**	**0.50**	**0.50**	0.47	0.49	0.49	-	-	-	-
Meat	**0.90**	**0.90**	0.75	0.75	0.81	0.81	0.49	0.49	-	-	-	-
Beef	**0.23**	**0.23**	**0.23**	**0.23**	0.17	0.17	0.19	0.19	-	-	-	-
FordA	0.56	**1.00**	0.61	0.50	0.65	0.40	0.84	0.52	-	-	-	-
FordB	0.51	**1.00**	0.62	0.72	0.47	0.67	0.48	0.47	-	-	-	-
ShapeletSim	0.56	**1.00**	0.71	0.00	0.53	0.50	0.73	0.43	-	-	-	-
BeetleFly	0.47	0.33	0.50	0.46	0.45	0.50	**0.80**	0.76	-	-	-	-
BirdChicken	0.52	0.59	0.59	0.59	0.47	0.48	**0.64**	**0.64**	-	-	-	-
Earthquakes	0.55	0.00	0.55	0.50	0.53	**1.00**	0.48	0.50	-	-	-	-
Herring	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	-	-	-	-
OliveOil	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	-	-	-	-
Car	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	-	-	-	-
Lighting2	0.57	0.00	0.58	**1.00**	0.53	0.00	0.65	**1.00**	-	-	-	-
Computers	0.45	0.45	0.53	**0.68**	0.48	0.43	0.59	0.45	-	-	-	-
LargeKitchenAppliances	0.30	0.67	0.59	0.75	0.67	**0.82**	0.49	0.52	-	-	-	-
RefrigerationDevices	0.21	0.00	0.68	**0.82**	0.33	0.00	0.80	0.60	-	-	-	-
ScreenType	0.26	0.00	0.32	0.38	0.33	**0.80**	0.50	0.57	-	-	-	-
SmallKitchenAppliances	0.38	0.33	0.30	0.17	**0.50**	0.33	0.44	0.33	-	-	-	-
WormsTwoClass	0.52	0.50	**0.64**	0.63	0.47	0.43	0.51	0.39	-	-	-	-
Worms	0.10	0.00	0.32	0.33	0.13	0.00	**0.59**	0.43	-	-	-	-
StarLightCurves	0.81	0.81	0.81	0.81	0.75	**0.88**	0.81	0.81	-	-	-	-
Haptics	**0.30**	0.00	0.30	0.30	0.19	0.00	0.22	0.22	-	-	-	-
CinC_ECG_torso	0.45	0.41	0.40	0.40	0.50	0.42	0.53	**0.53**	-	-	-	-
HandOutlines	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	**0.77**	-	-	-	-

Table B.4: F1-score from all algorithms on all datasets when the scaling factor (f) is 1 and number of clusters (k) is set to the number of classes in each dataset

Dataset	F1-Measure											
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L		D-AA-Z		D-SA-Z	
	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.56	0.63	0.51	0.51	0.56	0.60	0.61	0.61	0.54	0.59	0.62	**0.63**
SonyAIBORobotSurfaceII	0.80	0.80	0.73	0.74	0.73	0.65	**0.80**	0.74	0.44	0.48	0.54	0.54
SonyAIBORobotSurface	0.46	0.53	0.60	0.57	0.77	0.76	0.68	0.56	0.46	0.67	0.48	**1.00**
DistalPhalanxOutlineCorrect	**0.61**	0.48	**0.61**	**0.61**	**0.61**	0.48	**0.61**	**0.61**	**0.61**	0.48	**0.61**	0.48
MiddlePhalanxOutlineCorrect	0.52	0.52	0.51	0.51	0.52	0.52	0.51	0.51	**0.60**	**0.60**	**0.60**	**0.60**
PhalangesOutlinesCorrect	0.45	0.45	0.48	0.48	0.51	0.51	0.46	0.46	0.45	0.45	0.61	**0.62**
ProximalPhalanxOutlineCorrect	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**
DistalPhalanxOutlineAgeGroup	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	**0.66**	**0.66**
MiddlePhalanxOutlineAgeGroup	**0.47**	**0.47**	**0.47**	**0.47**	**0.47**	**0.47**	**0.47**	**0.47**	0.43	0.43	**0.47**	**0.47**
ProximalPhalanxOutlineAgeGroup	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	0.52	0.52	0.52	0.52
TwoLeadECG	0.60	0.60	0.54	0.54	0.50	0.50	0.54	0.54	**0.61**	0.48	0.54	0.54
MoteStrain	0.56	0.47	0.48	0.50	0.48	0.56	0.56	0.46	0.59	0.46	0.68	**1.00**
ECG200	0.56	0.59	**0.61**	**0.61**	0.51	0.54	0.54	0.58	0.50	0.59	0.45	0.44
CBF	0.42	0.40	0.39	0.51	0.35	0.42	0.32	0.42	0.52	**0.60**	0.46	0.37
Two_Patterns	0.26	**1.00**	0.21	0.00	0.34	0.00	0.27	0.50	0.39	0.39	0.59	0.81
ECGFiveDays	0.52	0.52	0.52	0.52	0.51	0.51	0.51	0.51	0.46	0.50	0.52	**0.54**
ECG5000	0.40	0.45	0.49	0.49	0.26	0.33	0.44	0.44	0.52	**0.88**	0.42	0.55
Gun_Point	0.56	0.63	0.63	**0.64**	0.51	0.61	0.63	**0.64**	0.52	0.52	0.52	0.52
wafer	0.63	0.66	0.61	0.66	0.71	**0.75**	0.48	0.51	0.48	0.51	0.47	0.48
ChlorineConcentration	0.34	0.34	0.38	0.38	0.34	0.34	**0.43**	**0.43**	0.31	0.31	0.34	0.34
Wine	0.63	0.63	**0.73**	**0.73**	0.63	0.63	**0.73**	**0.73**	0.63	0.63	0.63	0.63
Strawberry	**0.61**	**0.61**	0.57	0.57	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**	0.57	0.57
ArrowHead	0.52	0.52	0.52	0.52	0.48	**0.54**	0.50	0.50	0.46	0.48	0.44	0.45
Trace	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.49	0.66	**0.69**	**0.69**
ToeSegmentation1	0.48	**0.71**	0.44	0.43	0.58	0.50	0.62	0.45	0.46	0.41	0.45	0.52
Coffee	0.80	0.80	0.73	0.73	0.80	0.80	0.80	0.80	**0.90**	**0.90**	**0.90**	**0.90**
ToeSegmentation2	0.52	**0.80**	0.56	0.50	0.49	0.48	0.53	0.51	0.50	0.50	0.57	0.57
FaceFour	0.35	0.50	0.48	0.55	0.25	**1.00**	0.38	0.53	0.15	0.00	0.35	0.29
yoga	0.56	0.54	0.52	0.52	0.52	0.50	**0.57**	0.56	-	-	-	-
Ham	0.46	0.44	0.48	0.48	0.48	0.47	**0.48**	**0.48**	-	-	-	-
Meat	**0.90**	**0.90**	0.71	0.71	0.72	0.72	0.48	0.48	-	-	-	-
Beef	**0.27**	**0.27**	**0.27**	**0.27**	0.18	0.18	0.21	0.21	-	-	-	-
FordA	0.54	**1.00**	0.55	0.43	0.54	0.40	0.63	0.52	-	-	-	-
FordB	0.50	**1.00**	0.53	0.57	0.47	0.67	0.48	0.47	-	-	-	-
ShapeletSim	0.54	**1.00**	0.55	0.00	0.49	0.40	0.56	0.43	-	-	-	-
BeetleFly	0.47	0.33	0.48	0.46	0.44	0.50	**0.60**	0.59	-	-	-	-
BirdChicken	0.48	0.51	0.52	0.52	0.46	0.46	**0.54**	**0.54**	-	-	-	-
Earthquakes	0.56	0.00	0.56	0.40	0.50	**1.00**	0.48	0.50	-	-	-	-
Herring	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	-	-	-	-
OliveOil	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**	-	-	-	-
Car	**0.44**	**0.44**	**0.44**	**0.44**	0.41	0.41	**0.44**	**0.44**	-	-	-	-
Lighting2	0.49	0.00	0.54	**1.00**	0.50	0.00	0.54	0.75	-	-	-	-
Computers	0.46	0.48	0.49	**0.55**	0.49	0.53	0.51	0.45	-	-	-	-
LargeKitchenAppliances	0.27	0.48	0.38	0.46	0.47	**0.58**	0.37	0.47	-	-	-	-
RefrigerationDevices	0.21	0.00	0.43	**0.45**	0.32	0.00	0.44	0.38	-	-	-	-
ScreenType	0.25	0.00	0.31	0.38	0.30	**0.67**	0.41	0.47	-	-	-	-
SmallKitchenAppliances	0.36	0.33	0.28	0.17	**0.39**	0.29	0.35	0.27	-	-	-	-
WormsTwoClass	0.52	**0.60**	0.54	0.53	0.47	0.43	0.48	0.41	-	-	-	-
Worms	0.09	0.00	0.27	0.29	0.12	0.00	**0.33**	0.26	-	-	-	-
StarLightCurves	**0.72**	**0.72**	**0.72**	**0.72**	0.50	0.63	**0.72**	**0.72**	-	-	-	-
Haptics	0.23	0.00	**0.25**	**0.25**	0.13	0.00	0.18	0.18	-	-	-	-
CinC_ECG_torso	0.29	0.30	0.30	0.30	0.31	0.32	**0.34**	0.32	-	-	-	-
HandOutlines	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**	**0.73**	-	-	-	-

Table B.5: AoR from all algorithms on all datasets when the scaling factor (f) is 1 and number of clusters (k) is set to the number of classes in each dataset

Dataset	Accuracy on Retrieval (AoR)											
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L		D-AA-Z		D-SA-Z	
	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	**1.00**	0.60	**1.00**	**1.00**	**1.00**	0.55	**1.00**	**1.00**	0.90	0.55	0.90	0.50
SonyAIBORobotSurfaceII	0.95	0.45	**1.00**	0.65	**1.00**	0.60	**1.00**	0.65	0.90	0.40	0.85	0.40
SonyAIBORobotSurface	**1.00**	0.81	**1.00**	0.81	**1.00**	0.94	**1.00**	0.88	0.81	0.25	0.88	0.19
DistalPhalanxOutlineCorrect	**1.00**	0.90	**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**	**1.00**	0.90	**1.00**	0.90
MiddlePhalanxOutlineCorrect	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	0.90	0.90	0.90
PhalangesOutlinesCorrect	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.95
ProximalPhalanxOutlineCorrect	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
DistalPhalanxOutlineAgeGroup	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
MiddlePhalanxOutlineAgeGroup	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
ProximalPhalanxOutlineAgeGroup	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
TwoLeadECG	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**
MoteStrain	**1.00**	0.60	**1.00**	0.85	**1.00**	0.70	**1.00**	0.70	0.80	0.40	0.80	0.30
ECG200	**1.00**	0.95	**1.00**	**1.00**	**1.00**	0.95	**1.00**	0.95	**1.00**	0.80	**1.00**	0.80
CBF	0.76	0.29	**0.90**	0.52	0.71	0.33	**0.90**	0.43	0.81	0.29	0.81	0.38
Two_Patterns	0.60	0.05	0.65	0.10	0.70	0.10	0.70	0.25	**1.00**	0.70	0.95	0.60
ECGFiveDays	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.84	**1.00**	0.95
ECG5000	**1.00**	0.83	**1.00**	**1.00**	**1.00**	0.78	**1.00**	**1.00**	0.94	0.56	**1.00**	0.44
Gun_Point	**1.00**	0.80	**1.00**	0.95	**1.00**	0.75	**1.00**	0.95	**1.00**	**1.00**	**1.00**	**1.00**
wafer	0.95	0.75	**1.00**	0.75	0.95	0.80	**1.00**	0.80	**1.00**	0.85	**1.00**	0.80
ChlorineConcentration	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Wine	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Strawberry	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
ArrowHead	0.95	0.95	0.95	0.95	**1.00**	0.86	0.95	0.95	0.90	0.86	**1.00**	0.95
Trace	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**
ToeSegmentation1	0.75	0.30	0.95	0.60	0.55	0.20	0.95	0.55	0.95	0.45	**1.00**	0.55
Coffee	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
ToeSegmentation2	0.90	0.50	**1.00**	0.75	0.85	0.40	**1.00**	0.80	0.95	0.60	0.95	0.65
FaceFour	0.94	0.17	**1.00**	0.78	0.83	0.11	**1.00**	0.67	0.61	0.17	0.78	0.22
yoga	**1.00**	0.95	**1.00**	**1.00**	**1.00**	0.95	**1.00**	0.95	-	-	-	-
Ham	**1.00**	0.85	**1.00**	**1.00**	**1.00**	0.80	**1.00**	**1.00**	-	-	-	-
Meat	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	-	-	-	-
Beef	0.60	0.60	0.60	0.60	**0.80**	**0.80**	0.75	0.75	-	-	-	-
FordA	0.55	0.25	0.75	0.40	0.75	0.25	**0.90**	0.60	-	-	-	-
FordB	0.75	0.20	**0.90**	0.45	0.70	0.35	0.80	0.50	-	-	-	-
ShapeletSim	0.55	0.15	0.70	0.10	0.65	0.20	**0.75**	0.40	-	-	-	-
BeetleFly	0.60	0.15	**1.00**	0.65	0.75	0.20	0.95	0.70	-	-	-	-
BirdChicken	0.95	0.85	**1.00**	**1.00**	**1.00**	0.85	**1.00**	**1.00**	-	-	-	-
Earthquakes	0.60	0.10	0.60	0.20	0.60	0.15	**0.80**	0.20	-	-	-	-
Herring	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	-	-	-	-
OliveOil	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	-	-	-	-
Car	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	-	-	-	-
Lighting2	0.60	0.15	**0.85**	0.15	0.60	0.10	0.80	0.25	-	-	-	-
Computers	0.80	0.55	**0.85**	0.60	**0.85**	0.40	**0.85**	0.55	-	-	-	-
LargeKitchenAppliances	0.86	0.38	**0.90**	0.48	**0.90**	0.52	**0.90**	0.57	-	-	-	-
RefrigerationDevices	0.62	0.10	**1.00**	0.52	0.62	0.14	0.90	0.43	-	-	-	-
ScreenType	0.71	0.24	**0.95**	0.48	0.71	0.33	0.86	0.33	-	-	-	-
SmallKitchenAppliances	0.76	0.19	**0.81**	0.33	**0.81**	0.24	0.76	0.38	-	-	-	-
WormsTwoClass	0.55	0.25	**0.90**	0.45	0.60	0.30	0.80	0.45	-	-	-	-
Worms	0.55	0.15	**0.85**	0.30	0.50	0.10	**0.85**	0.45	-	-	-	-
StarLightCurves	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**	-	-	-	-
Haptics	**1.00**	0.30	0.95	0.95	0.95	0.25	0.95	0.95	-	-	-	-
CinC_ECG_torso	**1.00**	0.85	**1.00**	**1.00**	0.95	0.75	**1.00**	0.90	-	-	-	-
HandOutlines	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	-	-	-	-

Table B.6: AoD from all algorithms on all datasets when the scaling factor (f) is 1 and number of clusters (k) is set to the number of classes in each dataset

Dataset	Accuracy on Detection (AoD)											
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L		D-AA-Z		D-SA-Z	
	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.86	**0.95**	0.92	0.92	0.85	0.95	0.92	0.92	0.82	0.93	0.81	0.94
SonyAIBORobotSurfaceII	0.77	0.92	0.83	0.92	0.83	0.92	0.84	**0.93**	0.73	0.90	0.77	0.93
SonyAIBORobotSurface	0.91	0.96	0.89	0.97	0.92	0.95	0.92	**0.98**	0.69	0.94	0.66	0.96
DistalPhalanxOutlineCorrect	0.96	0.99	**0.99**	**0.99**	0.96	0.99	**0.99**	**0.99**	0.96	0.98	0.96	0.98
MiddlePhalanxOutlineCorrect	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**	**0.99**	**0.99**	**0.99**
PhalangesOutlinesCorrect	**0.99**	**0.99**	0.98	0.98	**0.99**	**0.99**	0.98	0.98	0.99	0.99	0.96	0.99
ProximalPhalanxOutlineCorrect	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**	**0.99**	**0.99**	**0.99**
DistalPhalanxOutlineAgeGroup	0.99	0.99	0.99	0.99	**0.99**	**0.99**	**0.99**	**0.99**	0.99	0.99	0.99	0.99
MiddlePhalanxOutlineAgeGroup	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**	**0.99**	**0.99**	**0.99**
ProximalPhalanxOutlineAgeGroup	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**	**0.99**	0.99	0.99
TwoLeadECG	0.98	0.98	**0.98**	**0.98**	0.98	0.98	0.98	0.98	0.94	0.98	0.98	0.98
MoteStrain	0.81	0.97	0.91	0.95	0.85	0.95	0.89	0.96	0.75	0.95	0.75	**0.97**
ECG200	0.93	0.94	0.94	0.94	0.93	0.94	0.94	**0.95**	0.85	0.90	0.86	0.91
CBF	0.68	0.91	0.82	**0.94**	0.69	0.94	0.77	0.93	0.70	0.89	0.72	0.90
Two_Patterns	0.62	**1.00**	0.60	0.86	0.64	0.94	0.67	0.94	0.85	0.94	0.82	0.93
ECGFiveDays	**0.98**	**0.98**	**0.98**	**0.98**	0.98	0.98	0.98	0.98	0.93	0.98	0.96	0.98
ECG5000	0.87	0.92	0.95	0.95	0.85	0.93	**0.96**	**0.96**	0.82	0.94	0.79	0.95
Gun_Point	0.91	0.97	0.96	0.97	0.90	0.97	0.96	0.97	0.98	0.98	**0.99**	**0.99**
wafer	0.88	0.96	0.91	**0.97**	0.88	0.94	0.90	0.95	0.91	0.96	0.89	0.96
ChlorineConcentration	0.97	0.97	0.99	0.99	0.97	0.97	**0.99**	**0.99**	0.97	0.97	0.96	0.96
Wine	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Strawberry	0.95	0.95	**0.96**	**0.96**	0.95	0.95	**0.96**	**0.96**	0.95	0.95	0.94	0.94
ArrowHead	0.95	0.95	**0.95**	**0.95**	0.89	0.95	0.95	0.95	0.89	0.95	0.93	0.94
Trace	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.94	**0.97**	0.96	0.96
ToeSegmentation1	0.70	0.92	0.81	0.92	0.68	0.88	0.78	0.92	0.80	0.93	0.82	**0.93**
Coffee	0.99	0.99	1.00	1.00	1.00	1.00	**1.00**	**1.00**	0.99	0.99	0.99	0.99
ToeSegmentation2	0.80	**0.94**	0.87	0.94	0.77	0.94	0.86	0.91	0.82	0.93	0.84	0.94
FaceFour	0.68	0.86	0.87	0.92	0.64	**0.97**	0.84	0.93	0.67	0.89	0.71	0.90
yoga	0.95	0.96	0.97	0.97	0.95	0.96	0.97	**0.98**	-	-	-	-
Ham	0.92	**0.98**	0.98	0.98	0.88	0.96	0.97	0.97	-	-	-	-
Meat	**1.00**	**1.00**	**1.00**	**1.00**	1.00	1.00	1.00	1.00	-	-	-	-
Beef	1.00	1.00	**1.00**	**1.00**	1.00	1.00	**1.00**	**1.00**	-	-	-	-
FordA	0.72	**0.95**	0.75	0.90	0.72	0.87	0.81	0.94	-	-	-	-
FordB	0.66	0.89	0.78	**0.93**	0.72	0.91	0.77	0.90	-	-	-	-
ShapeletSim	0.71	0.90	0.63	**0.98**	0.69	0.93	0.74	0.94	-	-	-	-
BeetleFly	0.64	0.93	0.84	0.94	0.65	**0.97**	0.83	0.92	-	-	-	-
BirdChicken	0.87	0.93	0.94	0.94	0.89	0.94	**0.95**	**0.95**	-	-	-	-
Earthquakes	0.68	0.90	0.68	**0.99**	0.67	0.88	0.65	0.96	-	-	-	-
Herring	0.99	0.99	1.00	1.00	1.00	1.00	**1.00**	**1.00**	-	-	-	-
OliveOil	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	-	-	-	-
Car	0.97	0.97	0.97	0.97	0.97	0.97	**0.97**	**0.97**	-	-	-	-
Lighting2	0.66	0.85	0.66	0.90	0.64	0.85	0.70	**0.96**	-	-	-	-
Computers	0.81	0.91	0.82	0.92	0.76	0.92	0.80	**0.94**	-	-	-	-
LargeKitchenAppliances	0.73	**0.94**	0.75	0.90	0.78	0.94	0.77	0.90	-	-	-	-
RefrigerationDevices	0.62	0.91	0.83	0.94	0.67	**0.95**	0.77	0.93	-	-	-	-
ScreenType	0.69	0.89	0.78	**0.94**	0.73	0.91	0.72	0.92	-	-	-	-
SmallKitchenAppliances	0.67	**0.95**	0.75	0.92	0.70	0.94	0.76	0.93	-	-	-	-
WormsTwoClass	0.69	0.89	0.77	**0.93**	0.71	0.92	0.76	0.92	-	-	-	-
Worms	0.68	0.90	0.69	0.89	0.64	**0.91**	0.75	0.90	-	-	-	-
StarLightCurves	0.99	0.99	0.99	0.99	0.94	0.98	**0.99**	**0.99**	-	-	-	-
Haptics	0.77	0.92	0.97	0.97	0.73	0.93	**0.97**	**0.97**	-	-	-	-
CinC_ECG_torso	0.91	0.95	0.97	0.97	0.88	0.95	0.94	**0.98**	-	-	-	-
HandOutlines	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	-	-	-	-

Table B.7: Excess Rate from all algorithms on all datasets when the scaling factor (f) is 1 and number of clusters (k) is set to the number of classes in each dataset

Dataset	Excess Rate											
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L		D-AA-Z		D-SA-Z	
	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.09	0.45	**0.00**	**0.00**	0.09	0.50	**0.00**	**0.00**	0.18	0.50	0.18	0.55
SonyAIBORobotSurfaceII	0.14	0.59	**0.00**	0.35	0.09	0.45	**0.00**	0.35	0.25	0.67	0.26	0.65
SonyAIBORobotSurface	0.16	0.32	**0.00**	0.19	0.16	0.21	**0.00**	0.13	0.28	0.78	0.22	0.83
DistalPhalanxOutlineCorrect	**0.00**	0.10	**0.00**	**0.00**	**0.00**	0.10	**0.00**	**0.00**	**0.00**	0.10	**0.00**	0.10
MiddlePhalanxOutlineCorrect	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
PhalangesOutlinesCorrect	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.05	0.10
ProximalPhalanxOutlineCorrect	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
DistalPhalanxOutlineAgeGroup	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
MiddlePhalanxOutlineAgeGroup	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
ProximalPhalanxOutlineAgeGroup	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
TwoLeadECG	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.10	**0.00**	**0.00**
MoteStrain	0.05	0.43	**0.00**	0.15	0.09	0.36	**0.00**	0.30	0.24	0.62	0.27	0.73
ECG200	**0.00**	0.05	**0.00**	**0.00**	**0.00**	0.05	**0.00**	0.05	0.09	0.27	0.09	0.27
CBF	0.30	0.74	**0.17**	0.52	0.35	0.70	**0.17**	0.61	0.26	0.74	0.26	0.65
Two_Patterns	0.40	0.95	0.41	0.91	0.36	0.91	0.39	0.78	**0.09**	0.36	0.14	0.45
ECGFiveDays	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.14	0.27	0.17	0.22
ECG5000	0.05	0.21	**0.00**	**0.00**	0.10	0.30	**0.00**	**0.00**	0.15	0.50	0.10	0.60
Gun_Point	0.05	0.24	**0.00**	0.05	0.05	0.29	**0.00**	0.05	**0.00**	**0.00**	**0.00**	**0.00**
wafer	0.05	0.25	**0.00**	0.25	0.05	0.20	**0.00**	0.20	**0.00**	0.15	**0.00**	0.20
ChlorineConcentration	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.05	0.05	**0.00**	**0.00**	**0.00**	**0.00**
Wine	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Strawberry	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
ArrowHead	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.14	**0.00**	**0.00**	0.10	0.14	**0.00**	0.05
Trace	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.10	**0.00**	**0.00**
ToeSegmentation1	0.38	0.75	0.14	0.45	0.48	0.81	0.14	0.50	0.14	0.59	**0.13**	0.52
Coffee	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
ToeSegmentation2	0.22	0.57	0.05	0.29	0.26	0.65	**0.00**	0.20	0.17	0.48	0.21	0.46
FaceFour	0.19	0.86	**0.00**	0.22	0.29	0.90	**0.00**	0.33	0.42	0.84	0.26	0.79
yoga	0.05	0.10	**0.00**	**0.00**	0.09	0.14	0.05	0.10	-	-	-	-
Ham	0.05	0.19	**0.00**	**0.00**	**0.00**	0.20	**0.00**	**0.00**	-	-	-	-
Meat	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	-	-	-	-
Beef	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	-	-	-	-
FordA	0.50	0.77	0.29	0.62	0.32	0.77	**0.25**	0.50	-	-	-	-
FordB	0.32	0.82	**0.22**	0.61	0.39	0.70	0.27	0.55	-	-	-	-
ShapeletSim	0.45	0.85	**0.33**	0.90	0.41	0.82	0.35	0.65	-	-	-	-
BeetleFly	0.45	0.86	0.13	0.43	0.38	0.83	**0.05**	0.30	-	-	-	-
BirdChicken	0.17	0.26	**0.00**	**0.00**	0.17	0.29	**0.00**	**0.00**	-	-	-	-
Earthquakes	0.43	0.90	0.45	0.82	0.43	0.86	**0.30**	0.83	-	-	-	-
Herring	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	-	-	-	-
OliveOil	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	-	-	-	-
Car	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	-	-	-	-
Lighting2	0.43	0.86	**0.15**	0.85	0.43	0.90	0.27	0.77	-	-	-	-
Computers	0.33	0.54	0.23	0.45	0.26	0.65	**0.19**	0.48	-	-	-	-
LargeKitchenAppliances	0.25	0.67	**0.17**	0.57	0.24	0.56	0.21	0.50	-	-	-	-
RefrigerationDevices	0.43	0.91	**0.16**	0.56	0.43	0.87	0.21	0.63	-	-	-	-
ScreenType	0.32	0.77	**0.17**	0.58	0.35	0.70	0.18	0.68	-	-	-	-
SmallKitchenAppliances	0.30	0.83	0.32	0.72	**0.29**	0.79	0.33	0.67	-	-	-	-
WormsTwoClass	0.48	0.76	**0.18**	0.59	0.40	0.70	0.30	0.61	-	-	-	-
Worms	0.42	0.84	**0.19**	0.71	0.50	0.90	**0.19**	0.57	-	-	-	-
StarLightCurves	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.10	**0.00**	**0.00**	-	-	-	-
Haptics	0.05	0.71	**0.00**	**0.00**	**0.00**	0.74	**0.00**	**0.00**	-	-	-	-
CinC_ECG_torso	0.09	0.23	**0.00**	**0.00**	0.14	0.32	0.05	0.14	-	-	-	-
HandOutlines	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	-	-	-	-

Table B.8: Rand Index (RI) from all algorithms on all datasets when the scaling factor (f) is 1 and number of clusters (k) is chosen by the SSTSC algorithms

Dataset	Rand Index											
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L		D-AA-Z		D-SA-Z	
	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.54	0.52	0.49	0.49	0.51	0.49	0.52	0.52	0.50	**0.56**	0.48	0.53
SonyAIBORobotSurfaceII	0.71	0.68	0.66	0.60	0.66	0.53	0.63	0.55	0.58	**0.90**	0.61	0.80
SonyAIBORobotSurface	0.61	0.47	0.63	0.54	0.61	0.47	0.56	0.51	0.45	0.33	0.49	**1.00**
DistalPhalanxOutlineCorrect	**0.56**	0.56	0.54	0.54	0.49	0.48	0.52	0.52	0.49	0.48	0.54	0.54
MiddlePhalanxOutlineCorrect	0.48	0.48	0.49	0.49	0.48	0.48	0.49	0.49	**0.55**	**0.55**	**0.55**	**0.55**
PhalangesOutlinesCorrect	0.48	0.48	0.48	0.48	0.47	0.47	0.47	0.47	0.48	0.48	**0.49**	0.48
ProximalPhalanxOutlineCorrect	**0.56**	**0.56**	0.54	0.54	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**
DistalPhalanxOutlineAgeGroup	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.62	0.62	**0.77**	**0.77**
MiddlePhalanxOutlineAgeGroup	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	**0.61**	**0.61**	**0.61**	**0.61**
ProximalPhalanxOutlineAgeGroup	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**
TwoLeadECG	**0.58**	**0.58**	**0.58**	**0.58**	0.50	0.50	**0.58**	**0.58**	0.49	0.48	0.58	0.58
MoteStrain	0.65	0.83	0.64	0.66	0.56	0.61	0.62	0.73	0.59	0.86	0.56	**1.00**
ECG200	0.59	**0.62**	0.59	0.59	0.49	0.50	0.52	0.53	0.54	0.59	0.56	0.56
CBF	0.61	0.60	0.61	0.58	0.69	0.67	0.70	**0.83**	0.69	0.73	0.64	0.71
Two_Patterns	0.42	0.00	0.62	**1.00**	0.54	0.00	0.68	0.67	0.82	0.86	0.86	0.97
ECGFiveDays	0.68	0.68	0.68	0.68	**0.70**	**0.70**	**0.70**	**0.70**	0.53	0.56	0.49	0.47
ECG5000	0.82	0.80	**0.88**	**0.88**	0.80	0.79	0.74	0.74	0.79	0.86	0.75	0.82
Gun_Point	0.61	0.65	0.65	**0.65**	0.57	0.63	0.65	**0.65**	0.57	0.57	0.57	0.57
wafer	0.62	**0.67**	0.55	0.61	0.60	0.62	0.57	0.62	0.59	0.62	0.49	0.50
ChlorineConcentration	0.64	0.64	**0.66**	**0.66**	0.64	0.64	0.34	0.34	0.62	0.62	0.61	0.61
Wine	0.61	0.61	**0.73**	**0.73**	0.61	0.61	**0.73**	**0.73**	0.57	0.57	0.57	0.57
Strawberry	0.60	0.60	0.56	0.56	0.68	0.68	0.55	0.55	0.54	0.54	**0.74**	**0.74**
ArrowHead	0.66	0.66	0.65	0.65	0.63	0.61	**0.74**	**0.74**	0.49	0.47	0.53	0.52
Trace	0.81	0.81	0.81	0.81	**0.83**	**0.83**	0.81	0.81	0.63	0.73	0.74	0.74
ToeSegmentation1	0.50	0.33	0.56	0.49	0.53	**1.00**	0.56	0.55	0.56	0.50	0.50	0.55
Coffee	0.76	0.76	0.68	0.68	0.75	0.75	**0.83**	**0.83**	0.70	0.70	0.72	0.72
ToeSegmentation2	0.50	0.57	0.58	0.59	0.53	**0.62**	0.54	0.54	0.53	0.60	0.56	0.60
FaceFour	0.70	0.33	**0.86**	0.85	0.71	0.00	0.84	0.83	0.49	0.33	0.67	0.17
yoga	0.50	0.50	0.48	0.48	**0.51**	0.49	0.49	0.48	-	-	-	-
Ham	0.49	0.49	0.48	0.48	**0.51**	0.50	0.49	0.49	-	-	-	-
Meat	0.77	0.77	**0.81**	**0.81**	0.81	0.81	0.75	0.75	-	-	-	-
Beef	0.65	0.65	**0.69**	**0.69**	0.62	0.62	0.62	0.62	-	-	-	-
FordA	0.49	**1.00**	0.52	0.43	0.53	0.00	0.53	0.53	-	-	-	-
FordB	0.52	**1.00**	0.52	0.47	0.61	**1.00**	0.56	0.17	-	-	-	-
ShapeletSim	0.56	**1.00**	0.46	**1.00**	0.53	**1.00**	0.53	**1.00**	-	-	-	-
BeetleFly	0.52	0.33	0.53	0.58	0.53	**1.00**	0.57	0.59	-	-	-	-
BirdChicken	0.49	0.49	0.57	0.57	0.49	0.50	**0.58**	**0.58**	-	-	-	-
Earthquakes	0.59	**1.00**	0.59	0.50	0.52	**1.00**	0.48	0.50	-	-	-	-
Herring	**0.54**	**0.54**	0.49	0.49	**0.54**	**0.54**	**0.54**	**0.54**	-	-	-	-
OliveOil	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	-	-	-	-
Car	0.73	0.73	0.73	0.73	0.79	0.79	**0.80**	**0.80**	-	-	-	-
Lighting2	0.47	0.33	0.51	**1.00**	0.52	0.00	0.50	**1.00**	-	-	-	-
Computers	0.44	0.39	0.50	0.50	**0.54**	0.40	0.50	0.53	-	-	-	-
LargeKitchenAppliances	0.65	**0.71**	0.57	0.56	0.69	0.69	0.63	0.61	-	-	-	-
RefrigerationDevices	0.43	0.67	0.45	0.31	0.56	**1.00**	0.72	0.70	-	-	-	-
ScreenType	0.50	0.40	**0.65**	0.61	0.50	0.43	0.53	0.62	-	-	-	-
SmallKitchenAppliances	0.58	0.33	**0.65**	0.50	**0.65**	0.33	0.63	0.50	-	-	-	-
WormsTwoClass	0.51	0.50	0.53	**0.64**	0.52	0.47	0.52	0.57	-	-	-	-
Worms	0.65	**0.83**	0.72	0.67	0.35	0.67	0.62	0.53	-	-	-	-
StarLightCurves	**0.84**	**0.84**	**0.84**	**0.84**	0.82	0.84	**0.84**	**0.84**	-	-	-	-
Haptics	**0.77**	0.67	0.69	0.69	0.77	0.50	0.67	0.67	-	-	-	-
CinC_ECG_torso	0.79	0.79	0.73	0.73	0.80	**0.80**	0.80	0.80	-	-	-	-
HandOutlines	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	-	-	-	-

Table B.9: Precision from all algorithms on all datasets when the scaling factor (f) is 1 and number of clusters (k) is chosen by the SSTSC algorithms

Dataset	Precision											
	Algorithm 1		Algorithm 2		Algorithm 3		Algorithm 4		Algorithm 5		Algorithm 6	
	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.51	0.49	0.45	0.45	0.48	0.47	0.48	0.48	0.47	**0.55**	0.47	0.50
SonyAIBORobotSurfaceII	0.91	**1.00**	0.84	0.77	0.93	0.60	0.87	0.57	0.89	**1.00**	**1.00**	**1.00**
SonyAIBORobotSurface	0.78	0.80	0.69	0.67	0.78	0.80	0.58	0.57	0.80	**1.00**	0.50	**1.00**
DistalPhalanxOutlineCorrect	**0.56**	0.55	0.54	0.54	0.48	0.48	0.55	0.55	0.48	0.48	0.52	0.52
MiddlePhalanxOutlineCorrect	0.46	0.46	0.47	0.47	0.46	0.46	0.47	0.47	**0.52**	**0.52**	0.52	0.52
PhalangesOutlinesCorrect	0.39	0.39	0.41	0.41	0.44	0.44	0.41	0.41	0.45	0.45	**0.48**	**0.48**
ProximalPhalanxOutlineCorrect	**0.53**	**0.53**	0.51	0.51	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**
DistalPhalanxOutlineAgeGroup	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.40	0.40	**0.64**	**0.64**
MiddlePhalanxOutlineAgeGroup	**0.39**	**0.39**	**0.39**	**0.39**	**0.39**	**0.39**	**0.39**	**0.39**	0.39	0.39	0.39	0.39
ProximalPhalanxOutlineAgeGroup	**0.41**	**0.41**	**0.41**	**0.41**	**0.41**	**0.41**	**0.41**	**0.41**	**0.41**	**0.41**	**0.41**	**0.41**
TwoLeadECG	0.59	0.59	0.59	0.59	0.45	0.45	0.59	0.59	0.48	0.48	**0.76**	**0.76**
MoteStrain	0.83	0.95	0.79	0.78	0.64	0.78	0.85	0.95	0.65	**1.00**	0.56	**1.00**
ECG200	0.65	**0.72**	0.62	0.62	0.44	0.46	0.48	0.51	0.55	0.63	0.59	0.57
CBF	0.38	0.33	0.33	0.36	0.50	**1.00**	0.48	**1.00**	0.50	**1.00**	0.39	**1.00**
Two_Patterns	0.21	0.00	0.19	0.00	0.27	0.00	0.00	0.00	0.88	**1.00**	0.76	**1.00**
ECGFiveDays	0.81	0.81	0.81	0.81	**0.86**	**0.86**	**0.86**	**0.86**	0.50	0.67	0.46	0.47
ECG5000	0.46	0.45	0.64	0.64	0.31	0.33	0.33	0.33	0.35	**0.75**	0.33	**0.75**
Gun_Point	0.68	0.70	**0.81**	0.81	0.62	0.67	**0.81**	0.81	0.56	0.56	0.56	0.56
wafer	0.90	**1.00**	0.58	0.73	0.93	0.92	0.74	0.92	0.82	0.88	0.37	0.42
ChlorineConcentration	0.35	0.35	**0.38**	**0.38**	0.35	0.35	0.30	0.30	0.27	0.27	0.24	0.24
Wine	0.57	0.57	**0.70**	**0.70**	0.57	0.57	**0.70**	**0.70**	0.56	0.56	0.56	0.56
Strawberry	0.71	0.71	0.64	0.64	**0.89**	**0.89**	0.57	0.57	0.53	0.53	0.84	0.84
ArrowHead	0.46	0.46	0.45	0.45	0.42	0.42	**0.62**	**0.62**	0.31	0.31	0.34	0.34
Trace	0.55	0.55	0.55	0.55	**0.62**	**0.62**	0.55	0.55	0.34	0.44	0.44	0.44
ToeSegmentation1	0.29	0.00	0.60	0.33	0.60	**1.00**	0.67	0.50	0.80	0.00	0.44	0.50
Coffee	0.88	0.88	0.73	0.73	0.96	0.96	**1.00**	**1.00**	0.97	0.97	0.97	0.97
ToeSegmentation2	0.41	0.50	0.68	0.80	0.50	0.67	0.56	0.56	0.50	**1.00**	0.58	0.71
FaceFour	0.32	0.33	0.77	0.80	0.30	0.00	0.85	**0.90**	0.22	0.33	0.31	0.17
yoga	**0.46**	**0.46**	0.42	0.42	**0.46**	0.44	0.42	0.42	-	-	-	-
Ham	0.44	0.45	0.45	0.45	**0.47**	0.47	0.42	0.42	-	-	-	-
Meat	0.56	0.56	**0.67**	**0.67**	0.65	0.65	0.61	0.61	-	-	-	-
Beef	0.18	0.18	**0.19**	**0.19**	0.19	0.19	0.19	0.19	-	-	-	-
FordA	0.40	**1.00**	0.49	0.25	0.43	0.00	0.50	0.50	-	-	-	-
FordB	0.49	**1.00**	0.48	0.60	0.67	**1.00**	**1.00**	**1.00**	-	-	-	-
ShapeletSim	0.52	**1.00**	0.45	0.00	0.45	**1.00**	0.40	**1.00**	-	-	-	-
BeetleFly	0.57	0.33	0.50	0.60	0.47	**1.00**	0.89	0.86	-	-	-	-
BirdChicken	0.36	0.37	0.65	0.65	0.38	0.39	**0.78**	**0.78**	-	-	-	-
Earthquakes	0.57	0.00	0.57	0.33	0.47	**1.00**	0.46	0.50	-	-	-	-
Herring	**0.52**	**0.52**	0.48	0.48	**0.52**	**0.52**	**0.52**	**0.52**	-	-	-	-
OliveOil	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	**0.56**	-	-	-	-
Car	0.39	0.39	0.39	0.39	0.50	0.50	**0.52**	**0.52**	-	-	-	-
Lighting2	0.44	0.00	0.50	**1.00**	0.47	0.00	0.33	**1.00**	-	-	-	-
Computers	0.31	0.40	0.42	0.40	**0.52**	0.50	0.39	0.43	-	-	-	-
LargeKitchenAppliances	0.30	**0.67**	0.31	0.50	0.42	**0.67**	0.36	0.55	-	-	-	-
RefrigerationDevices	0.25	0.00	0.31	0.31	0.21	**1.00**	0.67	0.00	-	-	-	-
ScreenType	0.30	0.14	0.36	0.25	0.27	0.27	0.35	**0.47**	-	-	-	-
SmallKitchenAppliances	**0.34**	0.33	0.30	0.33	0.33	0.33	0.33	0.33	-	-	-	-
WormsTwoClass	0.38	**1.00**	0.50	0.64	0.47	0.43	0.40	0.50	-	-	-	-
Worms	0.18	0.00	0.23	**0.25**	0.13	0.00	0.21	0.19	-	-	-	-
StarLightCurves	**0.84**	**0.84**	**0.84**	**0.84**	0.82	0.82	**0.84**	**0.84**	-	-	-	-
Haptics	**0.28**	0.00	0.19	0.19	0.22	0.00	0.15	0.15	-	-	-	-
CinC_ECG_torso	0.50	0.55	0.30	0.30	0.60	**0.63**	0.54	0.54	-	-	-	-
HandOutlines	**0.80**	**0.80**	**0.80**	**0.80**	**0.80**	**0.80**	**0.80**	**0.80**	-	-	-	-

Table B.10: Recall from all algorithms on all datasets when the scaling factor (f) is 1 and number of clusters (k) is chosen by the SSTSC algorithms

Dataset	Recall											
	Algorithm 1		Algorithm 2		Algorithm 3		Algorithm 4		Algorithm 5		Algorithm 6	
	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.44	0.87	0.29	0.29	0.38	0.84	0.29	0.29	0.64	0.63	**0.89**	0.86
SonyAIBORobotSurfaceII	0.42	0.44	0.34	0.26	0.31	0.11	0.25	0.13	0.14	**0.75**	0.17	0.50
SonyAIBORobotSurface	0.30	0.44	0.45	0.50	0.30	0.44	0.42	0.47	0.10	0.33	0.30	**1.00**
DistalPhalanxOutlineCorrect	0.37	0.44	0.52	0.52	**0.82**	0.48	0.51	0.51	**0.82**	0.48	0.47	0.47
MiddlePhalanxOutlineCorrect	0.59	0.59	0.56	0.56	0.59	0.59	0.56	0.56	0.50	0.50	**0.67**	**0.67**
PhalangesOutlinesCorrect	0.19	0.19	0.21	0.21	0.33	0.33	0.21	0.21	0.46	0.46	**0.90**	0.48
ProximalPhalanxOutlineCorrect	**0.53**	**0.53**	0.42	0.42	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**
DistalPhalanxOutlineAgeGroup	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	**0.58**	**0.58**	0.53	0.53
MiddlePhalanxOutlineAgeGroup	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**	0.51	0.51	0.51	0.51
ProximalPhalanxOutlineAgeGroup	**0.84**	**0.84**	**0.84**	**0.84**	**0.84**	**0.84**	**0.84**	**0.84**	**0.84**	**0.84**	**0.84**	**0.84**
TwoLeadECG	0.39	0.39	0.39	0.39	0.23	0.23	0.39	0.39	**0.82**	0.48	0.16	0.16
MoteStrain	0.32	0.68	0.32	0.39	0.16	0.23	0.24	0.43	0.27	0.69	0.29	**1.00**
ECG200	0.29	0.32	**0.37**	**0.37**	0.30	0.32	0.24	0.26	0.20	0.29	0.23	0.30
CBF	0.47	**0.50**	0.25	0.26	0.21	0.22	0.20	0.40	0.26	0.43	0.31	0.45
Two_Patterns	0.59	0.00	0.24	0.00	0.53	0.00	0.00	0.00	0.21	0.31	0.50	**0.85**
ECGFiveDays	0.43	0.43	0.43	0.43	0.44	0.44	0.44	0.44	0.06	0.11	**0.54**	0.47
ECG5000	0.44	0.56	0.56	0.56	0.20	0.27	**0.60**	**0.60**	0.32	0.43	0.56	0.43
Gun_Point	0.31	0.46	0.33	0.36	0.26	0.41	0.33	0.36	**0.48**	**0.48**	**0.48**	**0.48**
wafer	0.22	**0.31**	0.20	**0.31**	0.17	0.21	0.16	0.21	0.17	0.22	0.12	0.18
ChlorineConcentration	0.23	0.23	0.23	0.23	0.23	0.23	**0.90**	**0.90**	0.16	0.16	0.14	0.14
Wine	0.72	0.72	**0.77**	**0.77**	0.72	0.72	**0.77**	**0.77**	0.49	0.49	0.49	0.49
Strawberry	0.27	0.27	0.21	0.21	0.37	0.37	0.23	0.23	**0.72**	**0.72**	0.56	0.56
ArrowHead	**0.75**	**0.75**	0.65	0.65	0.56	0.74	0.38	0.38	0.56	0.63	0.60	0.65
Trace	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.80	**1.00**	**1.00**	**1.00**
ToeSegmentation1	0.06	0.00	0.16	0.15	0.11	**1.00**	0.13	0.16	0.10	0.00	0.19	0.20
Coffee	0.58	0.58	0.51	0.51	0.50	0.50	**0.64**	**0.64**	0.37	0.37	0.44	0.44
ToeSegmentation2	0.12	0.08	0.23	**0.26**	0.12	0.22	0.16	0.16	0.07	0.12	0.22	0.24
FaceFour	0.40	**1.00**	0.52	0.62	0.16	0.00	0.33	0.47	0.46	**1.00**	0.40	**1.00**
yoga	0.30	**0.33**	0.24	0.24	0.27	0.30	0.23	0.26	-	-	-	-
Ham	0.29	0.39	**0.50**	**0.50**	0.26	0.35	0.18	0.18	-	-	-	-
Meat	**1.00**	**1.00**	0.75	0.75	0.81	0.81	0.49	0.49	-	-	-	-
Beef	0.33	0.33	0.30	0.30	**0.43**	**0.43**	**0.43**	**0.43**	-	-	-	-
FordA	0.19	**1.00**	0.43	0.17	0.07	0.00	0.07	0.14	-	-	-	-
FordB	0.51	**1.00**	0.25	0.60	0.17	**1.00**	0.13	0.17	-	-	-	-
ShapeletSim	0.56	**1.00**	0.71	0.00	0.20	**1.00**	0.07	**1.00**	-	-	-	-
BeetleFly	0.24	0.33	0.14	0.29	0.19	**1.00**	0.10	0.14	-	-	-	-
BirdChicken	0.11	0.13	**0.19**	**0.19**	0.10	0.13	0.17	0.17	-	-	-	-
Earthquakes	0.55	0.00	0.55	0.50	0.53	**1.00**	0.54	0.50	-	-	-	-
Herring	0.38	0.38	**0.67**	**0.67**	0.38	0.38	0.38	0.38	-	-	-	-
OliveOil	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	**0.67**	-	-	-	-
Car	**0.50**	**0.50**	**0.50**	**0.50**	0.40	0.40	0.42	0.42	-	-	-	-
Lighting2	0.57	0.00	0.58	**1.00**	0.53	0.00	0.08	**1.00**	-	-	-	-
Computers	0.09	0.13	0.27	**0.33**	0.30	**0.33**	0.14	0.19	-	-	-	-
LargeKitchenAppliances	0.13	0.22	0.35	0.50	0.10	0.25	0.35	**0.75**	-	-	-	-
RefrigerationDevices	0.48	0.00	0.68	**1.00**	0.14	**1.00**	0.13	0.00	-	-	-	-
ScreenType	0.46	0.25	0.20	0.11	0.47	0.80	0.67	**1.00**	-	-	-	-
SmallKitchenAppliances	**0.38**	0.33	0.11	0.25	0.14	0.33	0.18	0.25	-	-	-	-
WormsTwoClass	0.12	0.17	0.24	0.44	**0.47**	0.43	0.04	0.11	-	-	-	-
Worms	0.36	0.00	0.32	0.33	**0.67**	0.00	0.53	0.43	-	-	-	-
StarLightCurves	0.57	0.57	0.57	0.57	0.51	**0.60**	0.57	0.57	-	-	-	-
Haptics	**0.30**	0.00	0.30	0.30	0.19	0.00	0.22	0.22	-	-	-	-
CinC_ECG_torso	0.19	0.23	0.20	0.20	0.17	**0.25**	0.19	0.22	-	-	-	-
HandOutlines	**0.40**	**0.40**	**0.40**	**0.40**	**0.40**	**0.40**	**0.40**	**0.40**	-	-	-	-

Table B.11: F1-score from all algorithms on all datasets when the scaling factor (f) is 1 and number of clusters (k) is chosen by the SSTSC algorithms

Dataset	F1-Measure											
	Algorithm 1		Algorithm 2		Algorithm 3		Algorithm 4		Algorithm 5		Algorithm 6	
	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.48	0.63	0.35	0.35	0.42	0.60	0.36	0.36	0.54	0.59	0.62	**0.63**
SonyAIBORobotSurfaceII	0.58	0.61	0.49	0.39	0.46	0.19	0.38	0.21	0.24	**0.86**	0.29	0.67
SonyAIBORobotSurface	0.43	0.57	0.55	0.57	0.43	0.57	0.49	0.52	0.18	0.50	0.38	**1.00**
DistalPhalanxOutlineCorrect	0.44	0.49	0.53	0.53	**0.61**	0.48	0.53	0.53	**0.61**	0.48	0.49	0.49
MiddlePhalanxOutlineCorrect	0.52	0.52	0.51	0.51	0.52	0.52	0.51	0.51	0.51	0.51	**0.58**	**0.58**
PhalangesOutlinesCorrect	0.26	0.26	0.27	0.27	0.38	0.38	0.28	0.28	0.45	0.45	**0.62**	0.48
ProximalPhalanxOutlineCorrect	**0.53**	**0.53**	0.46	0.46	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**
DistalPhalanxOutlineAgeGroup	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.47	0.47	**0.58**	**0.58**
MiddlePhalanxOutlineAgeGroup	**0.47**	**0.47**	**0.47**	**0.47**	**0.47**	**0.47**	**0.47**	**0.47**	0.44	0.44	0.44	0.44
ProximalPhalanxOutlineAgeGroup	**0.55**	**0.55**	**0.55**	**0.55**	**0.55**	**0.55**	**0.55**	**0.55**	**0.55**	**0.55**	**0.55**	**0.55**
TwoLeadECG	0.47	0.47	0.47	0.47	0.31	0.31	0.47	0.47	**0.61**	0.48	0.27	0.27
MoteStrain	0.46	0.79	0.46	0.52	0.25	0.35	0.38	0.59	0.38	0.82	0.38	**1.00**
ECG200	0.40	0.44	**0.46**	**0.46**	0.36	0.38	0.32	0.34	0.29	0.39	0.34	0.39
CBF	0.42	0.40	0.28	0.30	0.30	0.36	0.28	0.57	0.34	0.60	0.35	**0.63**
Two_Patterns	0.31	0.00	0.21	0.00	0.36	0.00	0.00	0.00	0.34	0.48	0.60	**0.92**
ECGFiveDays	0.56	0.56	0.56	0.56	**0.59**	**0.59**	**0.59**	**0.59**	0.10	0.19	0.50	0.47
ECG5000	0.45	0.50	**0.60**	**0.60**	0.24	0.30	0.43	0.43	0.33	0.55	0.42	0.55
Gun_Point	0.43	**0.55**	0.47	0.50	0.36	0.51	0.47	0.50	0.51	0.51	0.51	0.51
wafer	0.36	**0.48**	0.30	0.44	0.29	0.34	0.26	0.34	0.29	0.35	0.19	0.25
ChlorineConcentration	0.28	0.28	0.29	0.29	0.28	0.28	**0.45**	**0.45**	0.20	0.20	0.18	0.18
Wine	0.63	0.63	**0.73**	**0.73**	0.63	0.63	**0.73**	**0.73**	0.52	0.52	0.52	0.52
Strawberry	0.40	0.40	0.32	0.32	0.52	0.52	0.33	0.33	0.61	0.61	**0.67**	**0.67**
ArrowHead	**0.57**	**0.57**	0.54	0.54	0.48	0.54	0.47	0.47	0.40	0.42	0.44	0.45
Trace	0.49	0.49	0.49	0.49	0.52	0.52	0.49	0.49	0.48	0.61	**0.62**	**0.62**
ToeSegmentation1	0.09	0.00	0.25	0.21	0.19	**1.00**	0.21	0.24	0.18	0.00	0.26	0.29
Coffee	0.70	0.70	0.60	0.60	0.66	0.66	**0.78**	**0.78**	0.54	0.54	0.60	0.60
ToeSegmentation2	0.19	0.14	0.35	**0.39**	0.20	0.33	0.24	0.25	0.13	0.21	0.32	0.36
FaceFour	0.36	0.50	0.62	**0.70**	0.21	0.00	0.48	0.62	0.30	0.50	0.35	0.29
yoga	0.36	**0.39**	0.31	0.31	0.34	0.36	0.30	0.32	-	-	-	-
Ham	0.35	0.42	**0.48**	**0.48**	0.33	0.40	0.25	0.25	-	-	-	-
Meat	**0.72**	**0.72**	0.71	0.71	0.72	0.72	0.54	0.54	-	-	-	-
Beef	0.23	0.23	0.23	0.23	**0.26**	**0.26**	**0.26**	**0.26**	-	-	-	-
FordA	0.26	**1.00**	0.46	0.20	0.12	0.00	0.12	0.22	-	-	-	-
FordB	0.50	**1.00**	0.33	0.60	0.27	**1.00**	0.23	0.29	-	-	-	-
ShapeletSim	0.54	**1.00**	0.55	0.00	0.28	**1.00**	0.11	**1.00**	-	-	-	-
BeetleFly	0.33	0.33	0.22	0.39	0.27	**1.00**	0.18	0.24	-	-	-	-
BirdChicken	0.17	0.19	**0.29**	**0.29**	0.16	0.19	0.28	0.28	-	-	-	-
Earthquakes	0.56	0.00	0.56	0.40	0.50	**1.00**	0.50	0.50	-	-	-	-
Herring	0.44	0.44	**0.56**	**0.56**	0.44	0.44	0.44	0.44	-	-	-	-
OliveOil	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**	-	-	-	-
Car	0.44	0.44	0.44	0.44	0.44	0.44	**0.46**	**0.46**	-	-	-	-
Lighting2	0.49	0.00	0.54	**1.00**	0.50	0.00	0.13	**1.00**	-	-	-	-
Computers	0.14	0.19	0.33	0.36	0.38	**0.40**	0.21	0.26	-	-	-	-
LargeKitchenAppliances	0.18	0.33	0.33	0.50	0.16	0.36	0.35	**0.63**	-	-	-	-
RefrigerationDevices	0.33	0.00	0.43	0.47	0.17	**1.00**	0.22	0.00	-	-	-	-
ScreenType	0.36	0.18	0.25	0.15	0.35	0.40	0.46	**0.64**	-	-	-	-
SmallKitchenAppliances	**0.36**	0.33	0.16	0.29	0.20	0.33	0.24	0.29	-	-	-	-
WormsTwoClass	0.18	0.29	0.32	**0.52**	0.47	0.43	0.07	0.18	-	-	-	-
Worms	0.24	0.00	0.27	0.29	0.22	0.00	**0.30**	0.26	-	-	-	-
StarLightCurves	0.68	0.68	0.68	0.68	0.63	**0.69**	0.68	0.68	-	-	-	-
Haptics	**0.29**	0.00	0.23	0.23	0.20	0.00	0.18	0.18	-	-	-	-
CinC_ECG_torso	0.28	0.32	0.24	0.24	0.26	**0.36**	0.29	0.31	-	-	-	-
HandOutlines	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	**0.53**	-	-	-	-

Table B.12: AoR from all algorithms on all datasets when the scaling factor (f) is 1 and number of clusters (k) is chosen by the SSTSC algorithms

Dataset	Accuracy on Retrieval (AoR)											
	Algorithm 1		Algorithm 2		Algorithm 3		Algorithm 4		Algorithm 5		Algorithm 6	
	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	**1.00**	0.60	**1.00**	**1.00**	**1.00**	0.55	**1.00**	**1.00**	0.90	0.55	0.90	0.50
SonyAIBORobotSurfaceII	0.90	0.40	**1.00**	0.65	0.95	0.55	0.95	0.60	0.80	0.25	0.85	0.30
SonyAIBORobotSurface	**1.00**	0.63	**1.00**	0.81	**1.00**	0.63	**1.00**	0.88	0.75	0.19	0.88	0.19
DistalPhalanxOutlineCorrect	**1.00**	0.90	0.80	0.80	**1.00**	0.90	0.75	0.75	**1.00**	0.90	0.90	0.90
MiddlePhalanxOutlineCorrect	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	0.90	0.90	0.90
PhalangesOutlinesCorrect	0.80	0.80	0.90	0.90	0.85	0.85	0.85	0.85	**1.00**	**1.00**	0.95	0.90
ProximalPhalanxOutlineCorrect	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
DistalPhalanxOutlineAgeGroup	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.95	0.95	0.95	0.95
MiddlePhalanxOutlineAgeGroup	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
ProximalPhalanxOutlineAgeGroup	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
TwoLeadECG	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	0.95	0.95
MoteStrain	**1.00**	0.60	0.95	0.85	**1.00**	0.60	**1.00**	0.70	0.80	0.40	0.75	0.25
ECG200	**1.00**	0.95	**1.00**	**1.00**	**1.00**	0.95	**1.00**	0.95	0.95	0.70	0.95	0.70
CBF	0.76	0.29	**0.90**	0.52	0.71	0.33	**0.90**	0.43	0.81	0.29	0.81	0.33
Two_Patterns	0.65	0.10	0.65	0.10	0.65	0.10	0.60	0.15	**0.90**	0.65	**0.90**	0.60
ECGFiveDays	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.95	0.68	**1.00**	0.74
ECG5000	**1.00**	0.78	**1.00**	**1.00**	**1.00**	0.78	**1.00**	**1.00**	0.94	0.50	**1.00**	0.44
Gun_Point	**1.00**	0.80	**1.00**	0.95	**1.00**	0.75	**1.00**	0.95	**1.00**	**1.00**	**1.00**	**1.00**
wafer	0.95	0.75	**1.00**	0.75	0.95	0.80	**1.00**	0.80	0.95	0.85	0.95	0.80
ChlorineConcentration	0.95	0.95	0.95	0.95	0.95	0.95	**1.00**	**1.00**	0.95	0.95	0.95	0.95
Wine	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Strawberry	0.90	0.90	0.85	0.85	**1.00**	**1.00**	**1.00**	**1.00**	0.80	0.80	0.85	0.85
ArrowHead	0.95	0.95	0.90	0.90	**1.00**	0.86	0.81	0.81	0.90	0.86	**1.00**	0.95
Trace	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**
ToeSegmentation1	0.65	0.15	0.80	0.50	0.55	0.05	0.85	0.55	0.95	0.40	**1.00**	0.55
Coffee	0.85	0.85	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.95	0.95	0.85	0.85
ToeSegmentation2	0.90	0.40	0.95	0.70	0.80	0.35	**1.00**	0.80	0.95	0.55	0.95	0.50
FaceFour	0.89	0.17	**1.00**	0.78	0.72	0.00	**1.00**	0.67	0.61	0.17	0.78	0.22
yoga	0.95	0.90	**1.00**	**1.00**	**1.00**	0.85	0.95	0.90	-	-	-	-
Ham	**1.00**	0.85	**1.00**	**1.00**	**1.00**	0.80	**1.00**	**1.00**	-	-	-	-
Meat	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	-	-	-	-
Beef	0.95	0.95	0.95	0.95	**1.00**	**1.00**	**1.00**	**1.00**	-	-	-	-
FordA	0.50	0.05	**0.75**	0.40	0.70	0.00	0.70	0.30	-	-	-	-
FordB	0.75	0.20	**0.85**	0.30	0.40	0.05	0.70	0.20	-	-	-	-
ShapeletSim	0.55	0.15	**0.70**	0.10	0.55	0.05	0.60	0.05	-	-	-	-
BeetleFly	0.60	0.15	**1.00**	0.60	0.65	0.10	0.95	0.70	-	-	-	-
BirdChicken	0.95	0.80	**1.00**	**1.00**	**1.00**	0.80	0.95	0.95	-	-	-	-
Earthquakes	0.60	0.10	0.60	0.20	0.60	0.15	**0.80**	0.20	-	-	-	-
Herring	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	-	-	-	-
OliveOil	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	-	-	-	-
Car	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.95	0.95	-	-	-	-
Lighting2	0.60	0.15	**0.85**	0.15	0.60	0.10	0.75	0.15	-	-	-	-
Computers	0.70	0.40	0.60	0.40	0.70	0.25	**0.75**	0.45	-	-	-	-
LargeKitchenAppliances	0.86	0.38	**0.90**	0.43	**0.90**	0.48	0.81	0.43	-	-	-	-
RefrigerationDevices	0.67	0.14	**1.00**	0.48	0.57	0.05	0.71	0.24	-	-	-	-
ScreenType	0.76	0.29	**0.90**	0.38	0.71	0.33	0.86	0.33	-	-	-	-
SmallKitchenAppliances	**0.76**	0.19	0.67	0.24	0.67	0.14	0.71	0.24	-	-	-	-
WormsTwoClass	0.55	0.25	**0.90**	0.45	0.60	0.30	0.75	0.35	-	-	-	-
Worms	0.70	0.20	**0.85**	0.30	0.60	0.15	0.80	0.45	-	-	-	-
StarLightCurves	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**	-	-	-	-
Haptics	**1.00**	0.30	0.95	0.95	0.95	0.25	0.95	0.95	-	-	-	-
CinC_ECG_torso	0.95	0.80	**1.00**	**1.00**	0.95	0.70	0.95	0.90	-	-	-	-
HandOutlines	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	-	-	-	-

Table B.13: AoD from all algorithms on all datasets when the scaling factor (f) is 1 and number of clusters (k) is chosen by the SSTSC algorithms

Dataset	Accuracy on Detection (AoD)											
	Algorithm 1		Algorithm 2		Algorithm 3		Algorithm 4		Algorithm 5		Algorithm 6	
	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.86	**0.95**	0.92	0.92	0.85	0.95	0.92	0.92	0.82	0.93	0.81	0.94
SonyAIBORobotSurfaceII	0.75	0.91	0.83	0.92	0.80	0.90	0.84	**0.93**	0.70	0.89	0.73	0.91
SonyAIBORobotSurface	0.83	0.97	0.89	0.97	0.83	0.97	0.92	**0.98**	0.66	0.96	0.66	0.96
DistalPhalanxOutlineCorrect	0.96	0.99	0.99	0.99	0.96	0.99	**0.99**	**0.99**	0.96	0.98	0.98	0.98
MiddlePhalanxOutlineCorrect	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**	**0.99**	**0.99**	**0.99**
PhalangesOutlinesCorrect	**0.99**	**0.99**	0.98	0.98	**0.99**	**0.99**	0.98	0.98	0.99	0.99	0.94	0.99
ProximalPhalanxOutlineCorrect	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**	**0.99**	**0.99**	**0.99**
DistalPhalanxOutlineAgeGroup	0.99	0.99	0.99	0.99	**0.99**	**0.99**	**0.99**	**0.99**	0.99	0.99	0.99	0.99
MiddlePhalanxOutlineAgeGroup	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**	**0.99**	**0.99**	**0.99**
ProximalPhalanxOutlineAgeGroup	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	**0.99**	**0.99**	0.99	0.99
TwoLeadECG	0.98	0.98	**0.98**	**0.98**	0.98	0.98	0.98	0.98	0.94	0.98	0.98	0.98
MoteStrain	0.81	0.96	0.92	0.95	0.81	0.95	0.89	0.96	0.75	0.95	0.75	**0.97**
ECG200	0.93	0.94	0.94	0.94	0.93	0.94	0.94	**0.95**	0.83	0.91	0.83	0.90
CBF	0.68	0.91	0.82	**0.94**	0.68	0.91	0.76	0.91	0.70	0.89	0.70	0.89
Two_Patterns	0.63	**0.95**	0.60	0.86	0.65	0.93	0.63	0.89	0.82	0.93	0.82	0.93
ECGFiveDays	**0.98**	**0.98**	**0.98**	**0.98**	0.98	0.98	0.98	0.98	0.87	0.97	0.87	0.98
ECG5000	0.84	0.92	0.95	0.95	0.85	0.93	**0.96**	**0.96**	0.80	0.95	0.79	0.95
Gun_Point	0.90	0.97	0.96	0.97	0.89	0.96	0.96	0.97	0.98	0.98	**0.99**	**0.99**
wafer	0.88	0.96	0.91	**0.97**	0.90	0.94	0.90	0.95	0.93	0.96	0.91	0.96
ChlorineConcentration	0.98	0.98	0.99	0.99	0.98	0.98	**0.99**	**0.99**	0.97	0.97	0.96	0.96
Wine	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Strawberry	0.95	0.95	0.95	0.95	0.95	0.95	**0.96**	**0.96**	0.95	0.95	0.95	0.95
ArrowHead	0.95	0.95	**0.95**	**0.95**	0.89	0.95	0.95	0.95	0.89	0.95	0.93	0.94
Trace	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.94	**0.97**	0.96	0.96
ToeSegmentation1	0.63	0.91	0.78	0.91	0.62	0.84	0.78	0.92	0.79	0.93	0.82	**0.93**
Coffee	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	**1.00**	0.99	0.99	0.99	0.99
ToeSegmentation2	0.77	**0.94**	0.85	0.93	0.75	0.93	0.86	0.91	0.81	0.93	0.79	0.92
FaceFour	0.68	0.86	0.87	0.92	0.62	**0.98**	0.84	0.93	0.67	0.89	0.71	0.90
yoga	0.92	0.96	0.97	0.97	0.91	0.96	0.94	**0.98**	-	-	-	-
Ham	0.92	**0.98**	0.98	0.98	0.88	0.96	0.97	0.97	-	-	-	-
Meat	**1.00**	**1.00**	**1.00**	**1.00**	1.00	1.00	1.00	1.00	-	-	-	-
Beef	0.99	0.99	**0.99**	**0.99**	0.99	0.99	0.99	0.99	-	-	-	-
FordA	0.65	0.89	0.75	0.90	0.65	0.89	0.74	**0.91**	-	-	-	-
FordB	0.66	0.89	0.73	0.91	0.65	0.88	0.69	**0.91**	-	-	-	-
ShapeletSim	0.71	0.90	0.62	**0.97**	0.63	0.92	0.63	0.94	-	-	-	-
BeetleFly	0.64	0.93	0.82	0.92	0.61	**0.96**	0.83	0.92	-	-	-	-
BirdChicken	0.84	0.93	0.94	0.94	0.86	0.94	**0.95**	**0.95**	-	-	-	-
Earthquakes	0.68	0.90	0.68	**0.99**	0.67	0.88	0.64	0.96	-	-	-	-
Herring	0.99	0.99	1.00	1.00	1.00	1.00	**1.00**	**1.00**	-	-	-	-
OliveOil	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	-	-	-	-
Car	0.97	0.97	0.97	0.97	0.97	0.97	**0.97**	**0.97**	-	-	-	-
Lighting2	0.66	0.85	0.66	0.90	0.64	0.85	0.67	**0.98**	-	-	-	-
Computers	0.75	0.89	0.79	0.91	0.70	0.90	0.80	**0.93**	-	-	-	-
LargeKitchenAppliances	0.73	**0.94**	0.74	0.91	0.77	0.93	0.75	0.91	-	-	-	-
RefrigerationDevices	0.63	0.90	0.81	**0.93**	0.62	0.86	0.72	0.92	-	-	-	-
ScreenType	0.69	0.87	0.76	**0.94**	0.73	0.91	0.72	0.92	-	-	-	-
SmallKitchenAppliances	0.67	**0.95**	0.71	0.91	0.68	0.95	0.71	0.93	-	-	-	-
WormsTwoClass	0.68	0.89	0.76	0.92	0.71	**0.92**	0.71	0.92	-	-	-	-
Worms	0.67	0.92	0.69	0.89	0.65	**0.94**	0.76	0.90	-	-	-	-
StarLightCurves	0.99	0.99	0.99	0.99	0.94	0.98	**0.99**	**0.99**	-	-	-	-
Haptics	0.77	0.92	0.97	0.97	0.73	0.93	**0.97**	**0.97**	-	-	-	-
CinC_ECG_torso	0.89	0.95	0.97	0.97	0.85	0.96	0.94	**0.98**	-	-	-	-
HandOutlines	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	-	-	-	-

Table B.14: Excess Rate from all algorithms on all datasets when the scaling factor (f) is 1 and number of clusters (k) is chosen by the SSTSC algorithms

Dataset	Excess Rate											
	Algorithm 1		Algorithm 2		Algorithm 3		Algorithm 4		Algorithm 5		Algorithm 6	
	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.09	0.45	**0.00**	**0.00**	0.09	0.50	**0.00**	**0.00**	0.18	0.50	0.18	0.55
SonyAIBORobotSurfaceII	0.14	0.62	**0.00**	0.35	0.10	0.48	**0.00**	0.37	0.24	0.76	0.26	0.74
SonyAIBORobotSurface	0.16	0.47	**0.00**	0.19	0.16	0.47	**0.00**	0.13	0.29	0.82	0.22	0.83
DistalPhalanxOutlineCorrect	**0.00**	0.10	**0.00**	**0.00**	**0.00**	0.10	**0.00**	**0.00**	**0.00**	0.10	**0.00**	**0.00**
MiddlePhalanxOutlineCorrect	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
PhalangesOutlinesCorrect	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.05	0.10
ProximalPhalanxOutlineCorrect	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
DistalPhalanxOutlineAgeGroup	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
MiddlePhalanxOutlineAgeGroup	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
ProximalPhalanxOutlineAgeGroup	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
TwoLeadECG	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.10	**0.00**	**0.00**
MoteStrain	0.05	0.43	**0.00**	0.11	0.09	0.45	**0.00**	0.30	0.24	0.62	0.29	0.76
ECG200	**0.00**	0.05	**0.00**	**0.00**	**0.00**	0.05	**0.00**	0.05	0.10	0.33	0.10	0.33
CBF	0.30	0.74	**0.17**	0.52	0.35	0.70	**0.17**	0.61	0.26	0.74	0.26	0.70
Two_Patterns	0.38	0.90	0.41	0.91	0.38	0.90	0.43	0.86	**0.14**	0.38	**0.14**	0.43
ECGFiveDays	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.10	0.35	0.17	0.39
ECG5000	**0.00**	0.22	**0.00**	**0.00**	**0.00**	0.22	**0.00**	**0.00**	0.06	0.50	0.10	0.60
Gun_Point	**0.00**	0.20	**0.00**	0.05	**0.00**	0.25	**0.00**	0.05	**0.00**	**0.00**	**0.00**	**0.00**
wafer	0.05	0.25	**0.00**	0.25	**0.00**	0.16	**0.00**	0.20	**0.00**	0.11	**0.00**	0.16
ChlorineConcentration	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.05	0.05	**0.00**	**0.00**	**0.00**	**0.00**
Wine	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Strawberry	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
ArrowHead	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.14	**0.00**	**0.00**	0.10	0.14	**0.00**	0.05
Trace	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.10	**0.00**	**0.00**
ToeSegmentation1	0.41	0.86	0.16	0.47	0.48	0.95	0.15	0.45	0.14	0.64	**0.13**	0.52
Coffee	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
ToeSegmentation2	0.22	0.65	0.05	0.30	0.30	0.70	**0.00**	0.20	0.17	0.52	0.21	0.58
FaceFour	0.20	0.85	**0.00**	0.22	0.28	1.00	**0.00**	0.33	0.42	0.84	0.26	0.79
yoga	0.05	0.10	**0.00**	**0.00**	0.09	0.23	0.05	0.10	-	-	-	-
Ham	0.05	0.19	**0.00**	**0.00**	**0.00**	0.20	**0.00**	**0.00**	-	-	-	-
Meat	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	-	-	-	-
Beef	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	-	-	-	-
FordA	0.52	0.95	**0.29**	0.62	0.33	1.00	0.30	0.70	-	-	-	-
FordB	0.32	0.82	**0.23**	0.73	0.50	0.94	0.30	0.80	-	-	-	-
ShapeletSim	0.45	0.85	**0.33**	0.90	0.45	0.95	0.40	0.95	-	-	-	-
BeetleFly	0.45	0.86	0.13	0.48	0.41	0.91	**0.05**	0.30	-	-	-	-
BirdChicken	0.17	0.30	**0.00**	**0.00**	0.17	0.33	**0.00**	**0.00**	-	-	-	-
Earthquakes	0.43	0.90	0.45	0.82	0.43	0.86	**0.30**	0.83	-	-	-	-
Herring	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	-	-	-	-
OliveOil	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	-	-	-	-
Car	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	-	-	-	-
Lighting2	0.43	0.86	**0.15**	0.85	0.43	0.90	0.25	0.85	-	-	-	-
Computers	0.33	0.62	0.20	0.47	0.30	0.75	**0.12**	0.47	-	-	-	-
LargeKitchenAppliances	0.25	0.67	**0.17**	0.61	0.21	0.58	0.23	0.59	-	-	-	-
RefrigerationDevices	0.39	0.87	**0.09**	0.57	0.45	0.95	0.25	0.75	-	-	-	-
ScreenType	0.30	0.74	**0.14**	0.64	0.35	0.70	0.18	0.68	-	-	-	-
SmallKitchenAppliances	**0.30**	0.83	0.36	0.77	0.33	0.86	0.35	0.78	-	-	-	-
WormsTwoClass	0.48	0.76	**0.18**	0.59	0.40	0.70	0.32	0.68	-	-	-	-
Worms	0.39	0.83	**0.19**	0.71	0.45	0.86	0.20	0.55	-	-	-	-
StarLightCurves	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.10	**0.00**	**0.00**	-	-	-	-
Haptics	0.05	0.71	**0.00**	**0.00**	**0.00**	0.74	**0.00**	**0.00**	-	-	-	-
CinC_ECG_torso	0.10	0.24	**0.00**	**0.00**	0.14	0.36	0.05	0.10	-	-	-	-
HandOutlines	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	-	-	-	-

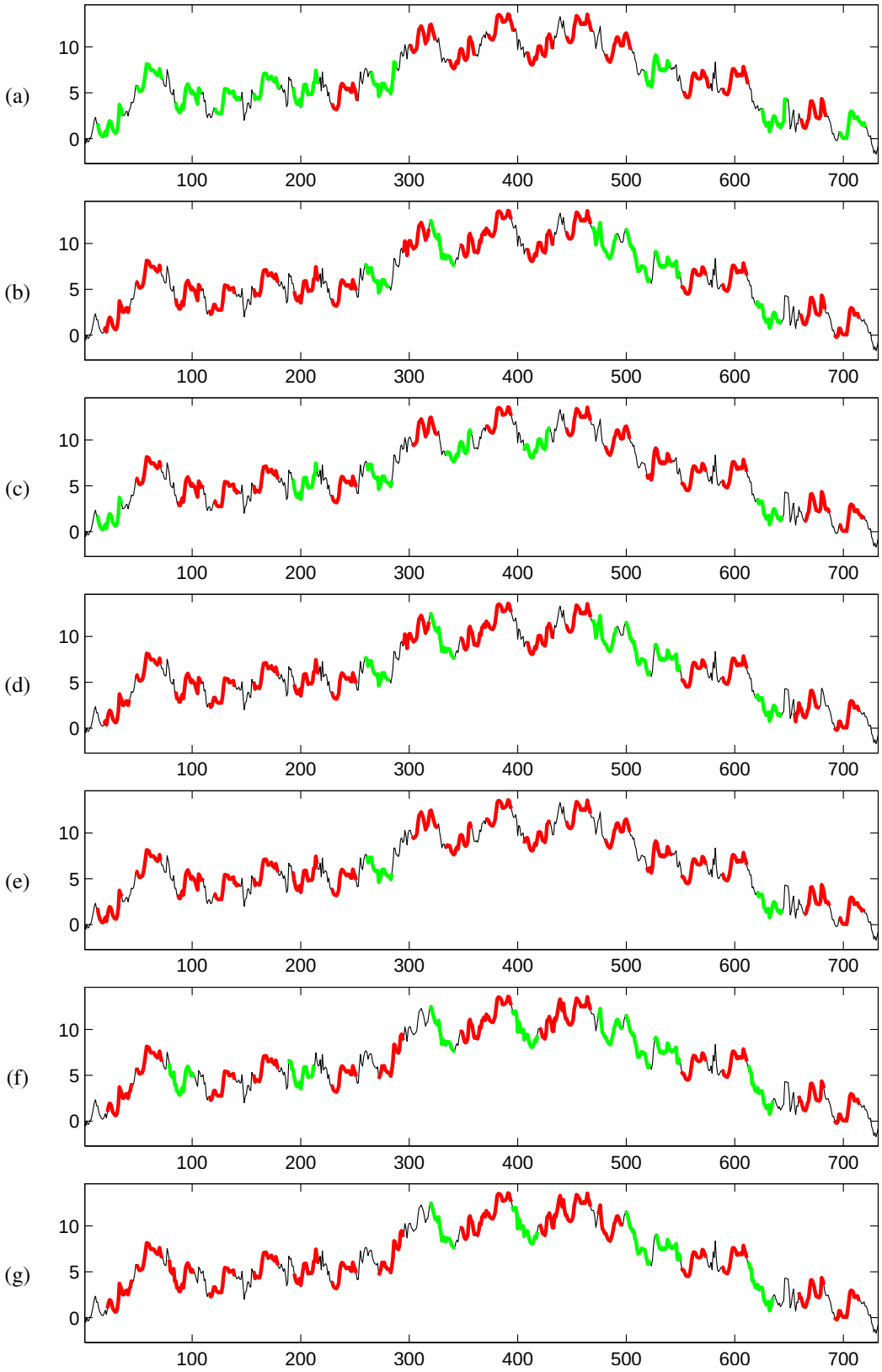

Figure B.1: ItalyPowerDemand dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

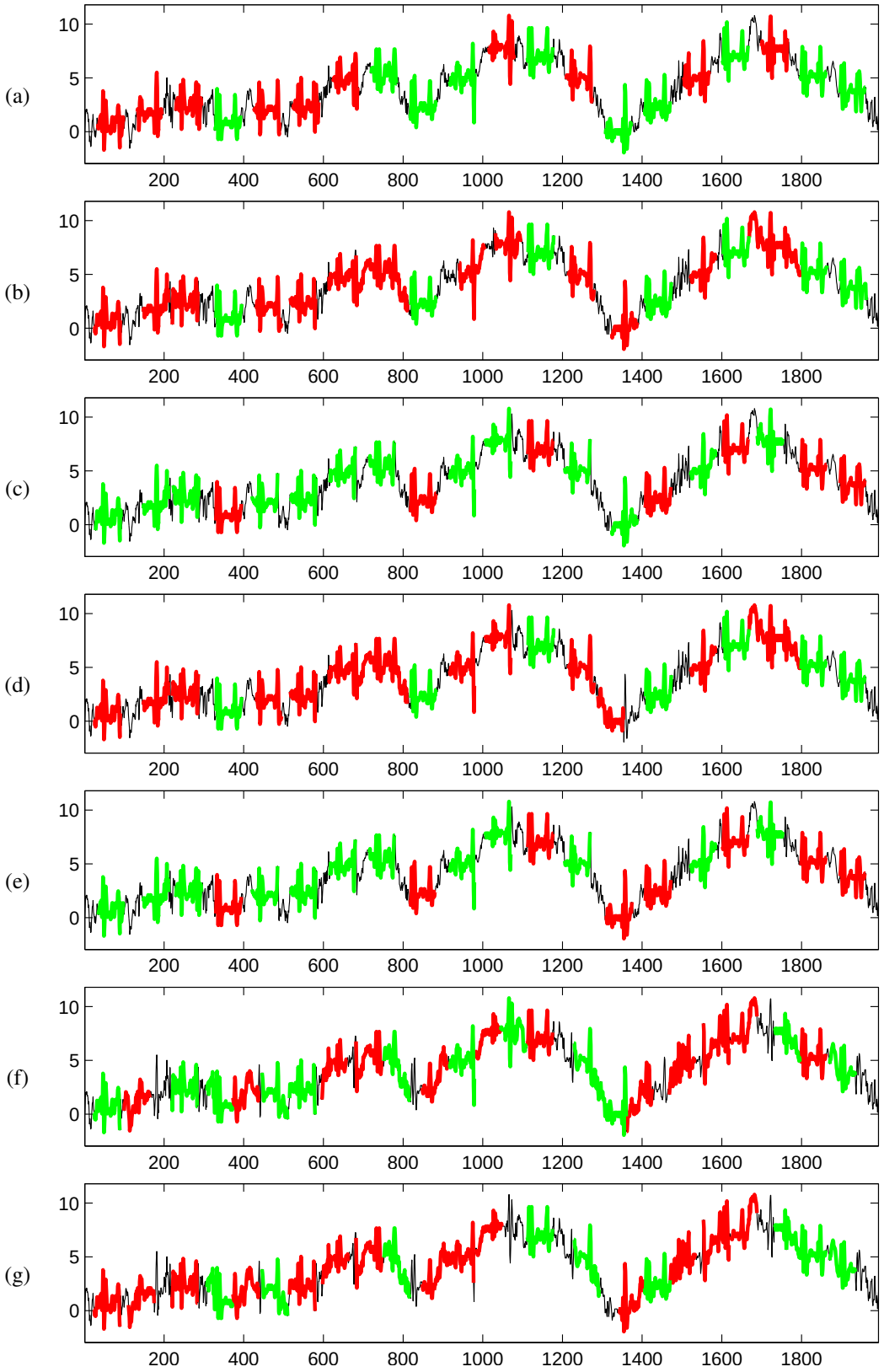

Figure B.2: SonyAIBORobotSurfaceII dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

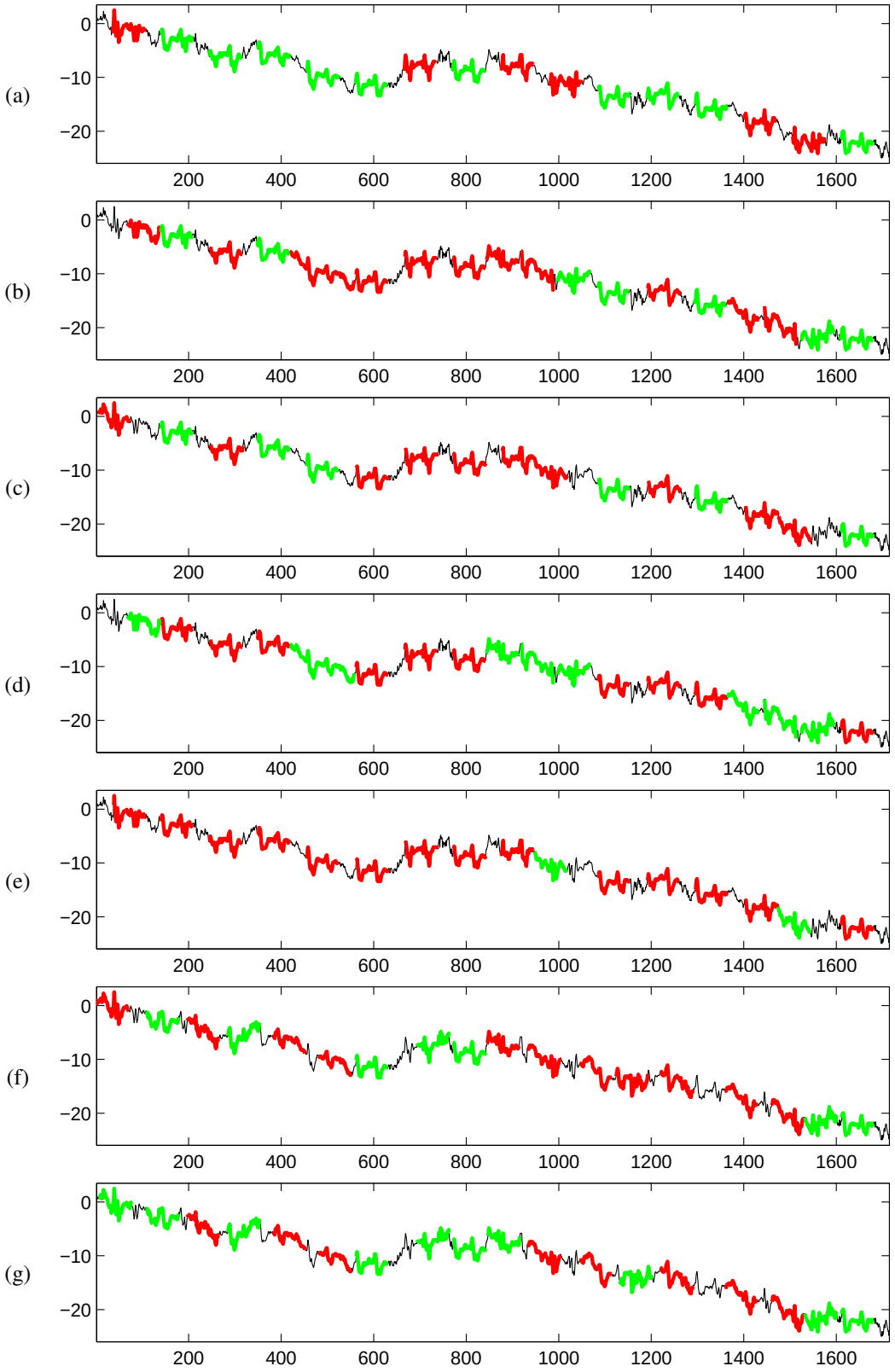

Figure B.3: SonyAIBORobotSurface dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

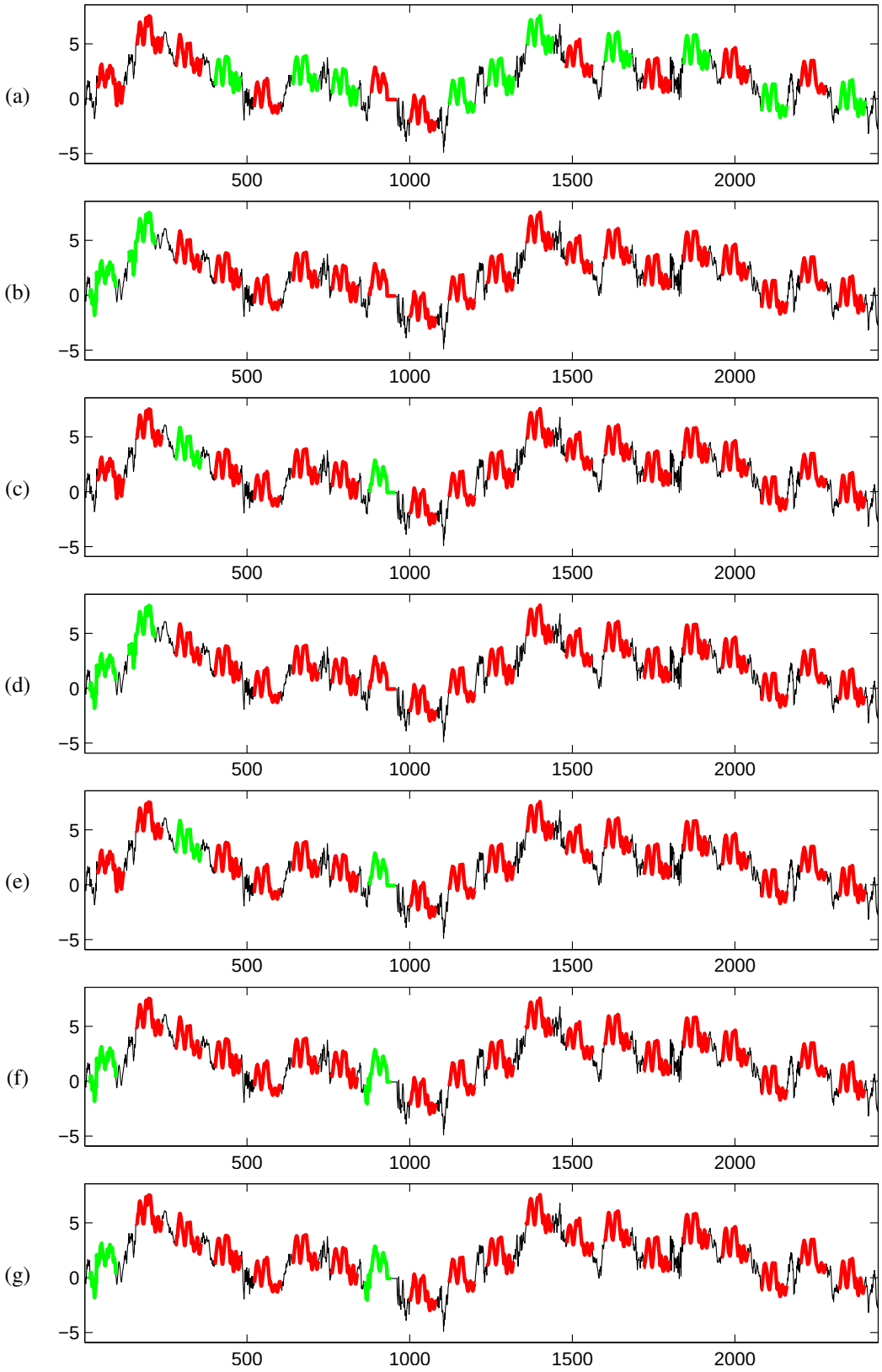

Figure B.4: DistalPhalanxOutlineCorrect dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

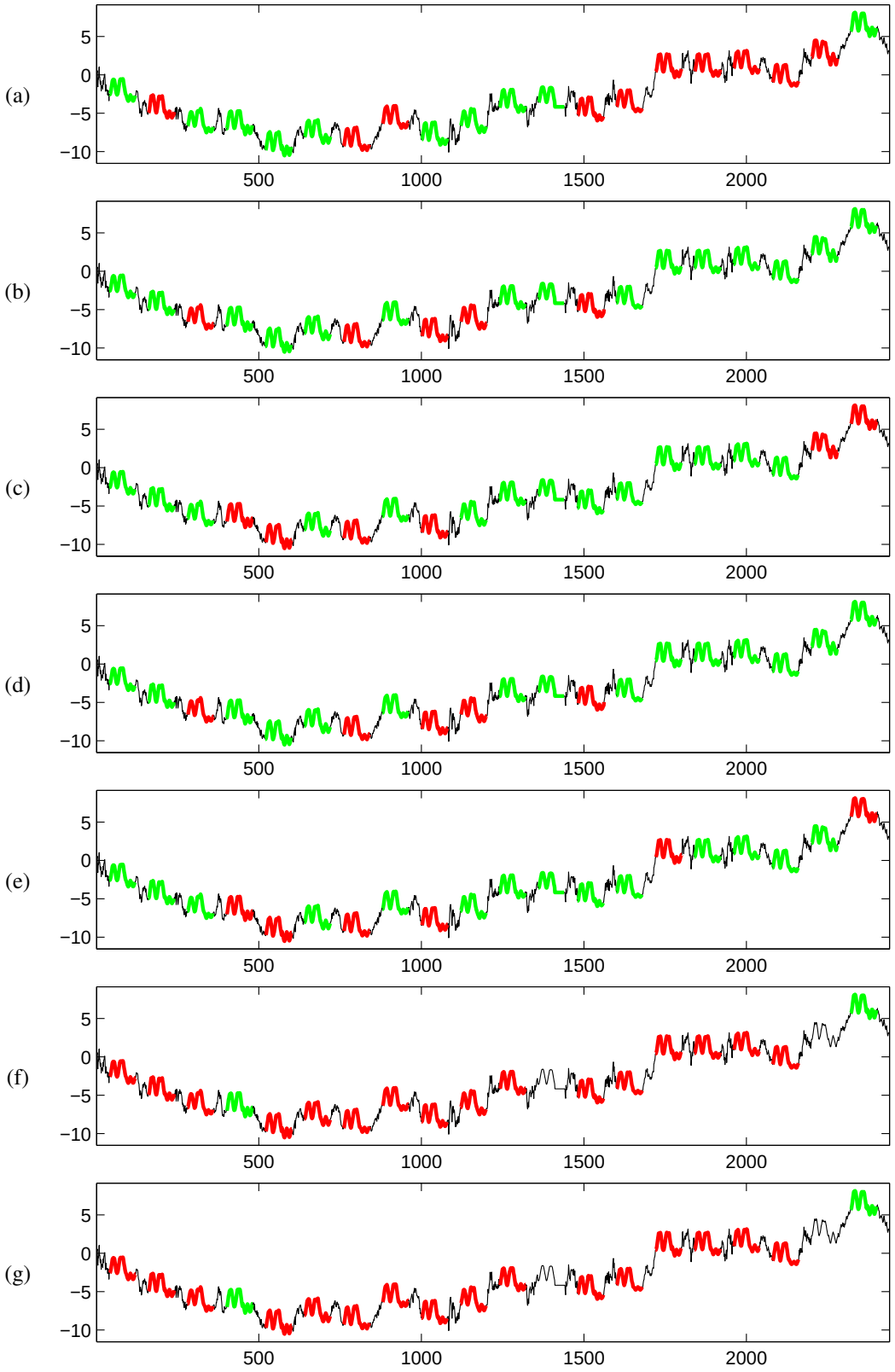

Figure B.5: MiddlePhalanxOutlineCorrect dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

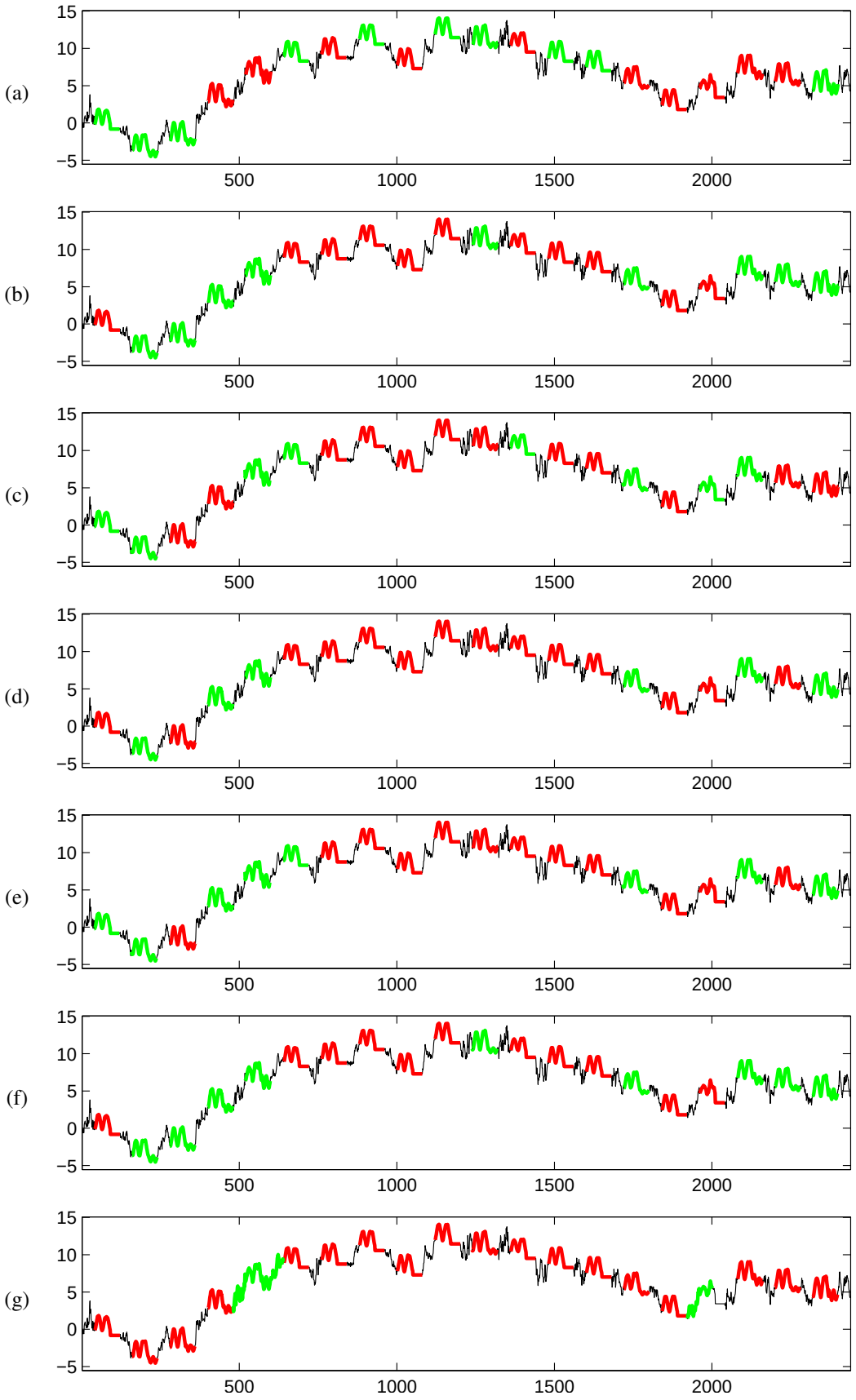

Figure B.6: PhalangesOutlinesCorrect dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

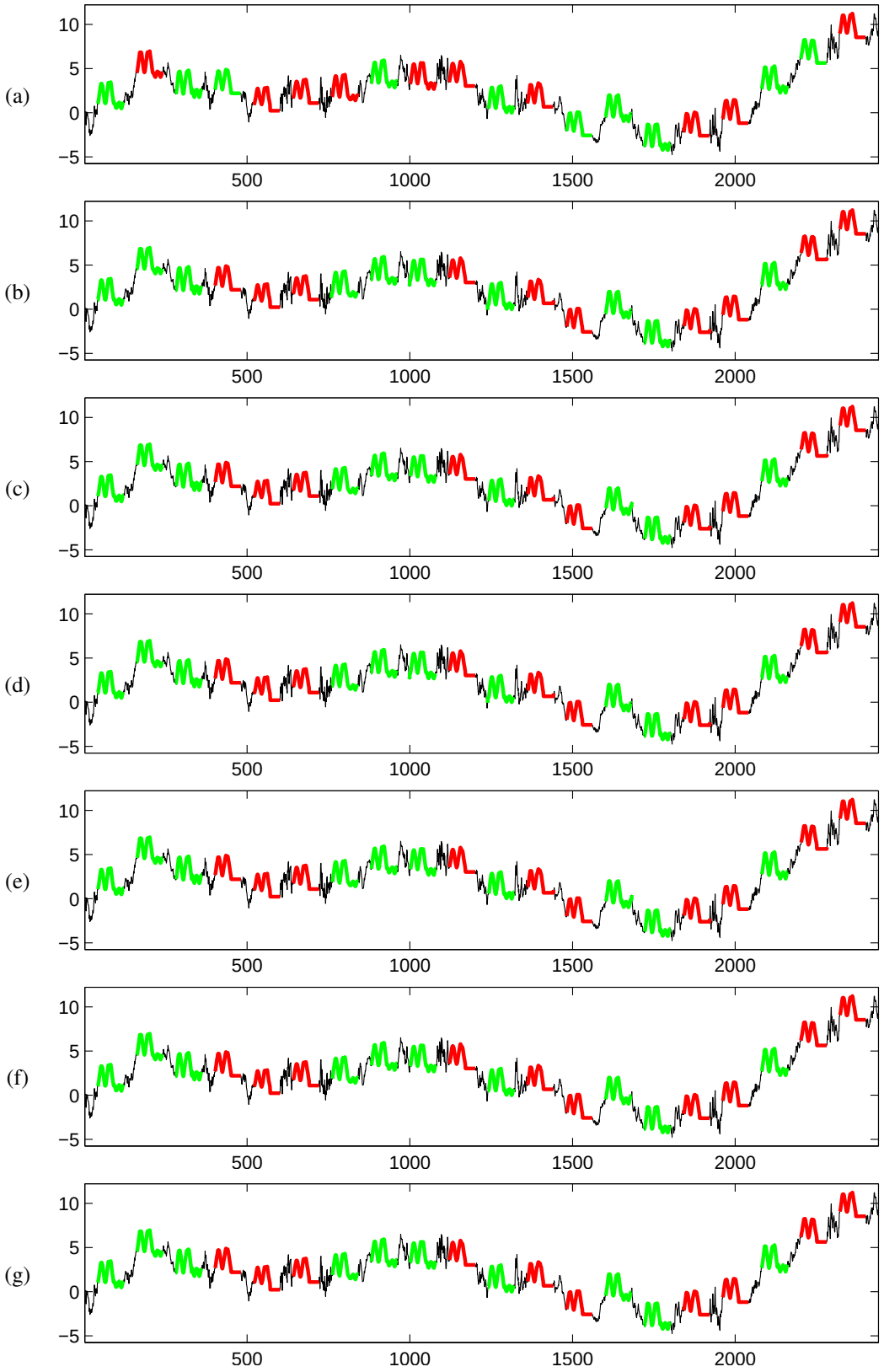

Figure B.7: ProximalPhalanxOutlineCorrect dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

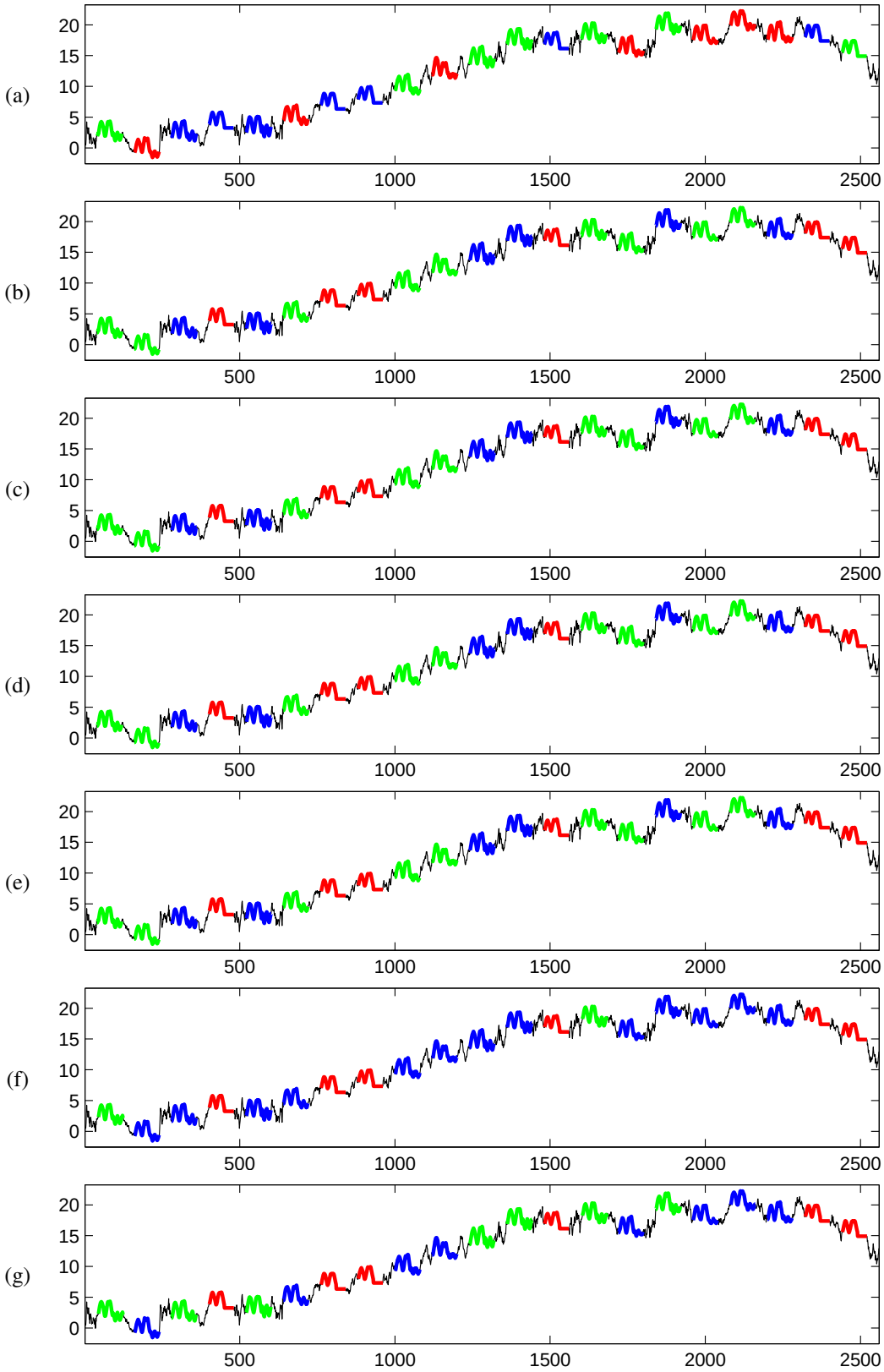

Figure B.8: DistalPhalanxOutlineAgeGroup dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

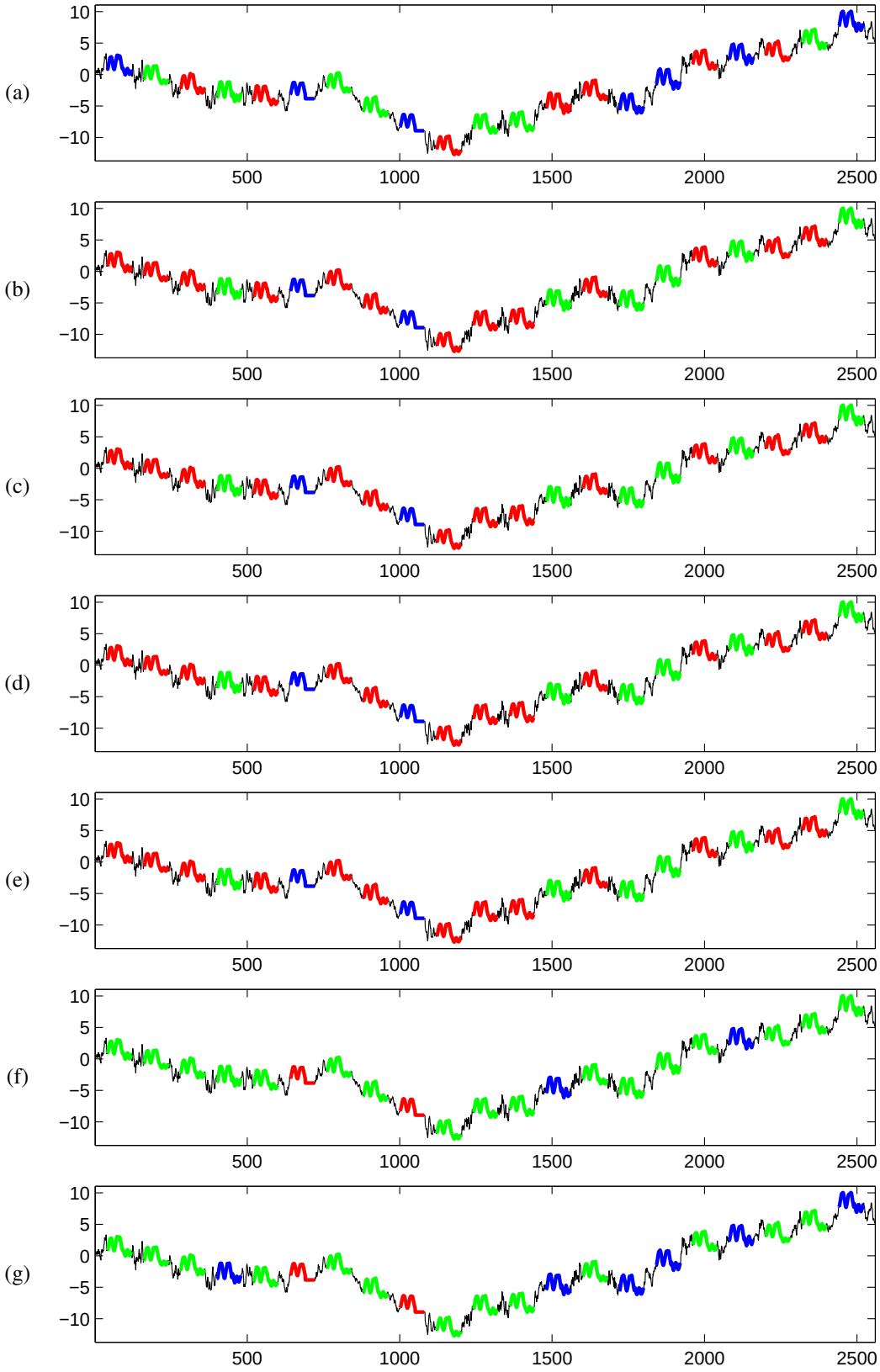

Figure B.9: MiddlePhalanxOutlineAgeGroup dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

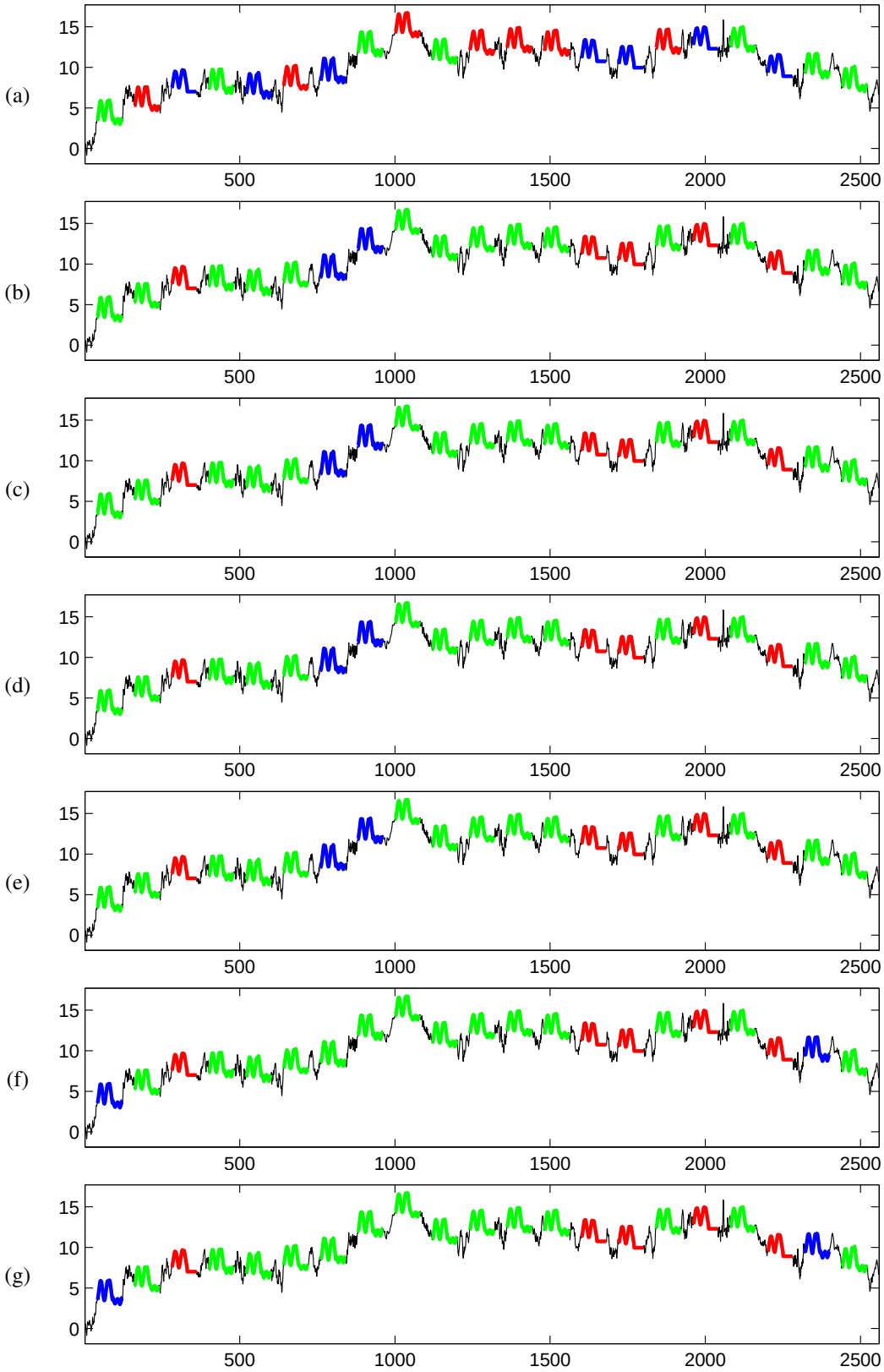

Figure B.10: ProximalPhalanxOutlineAgeGroup dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

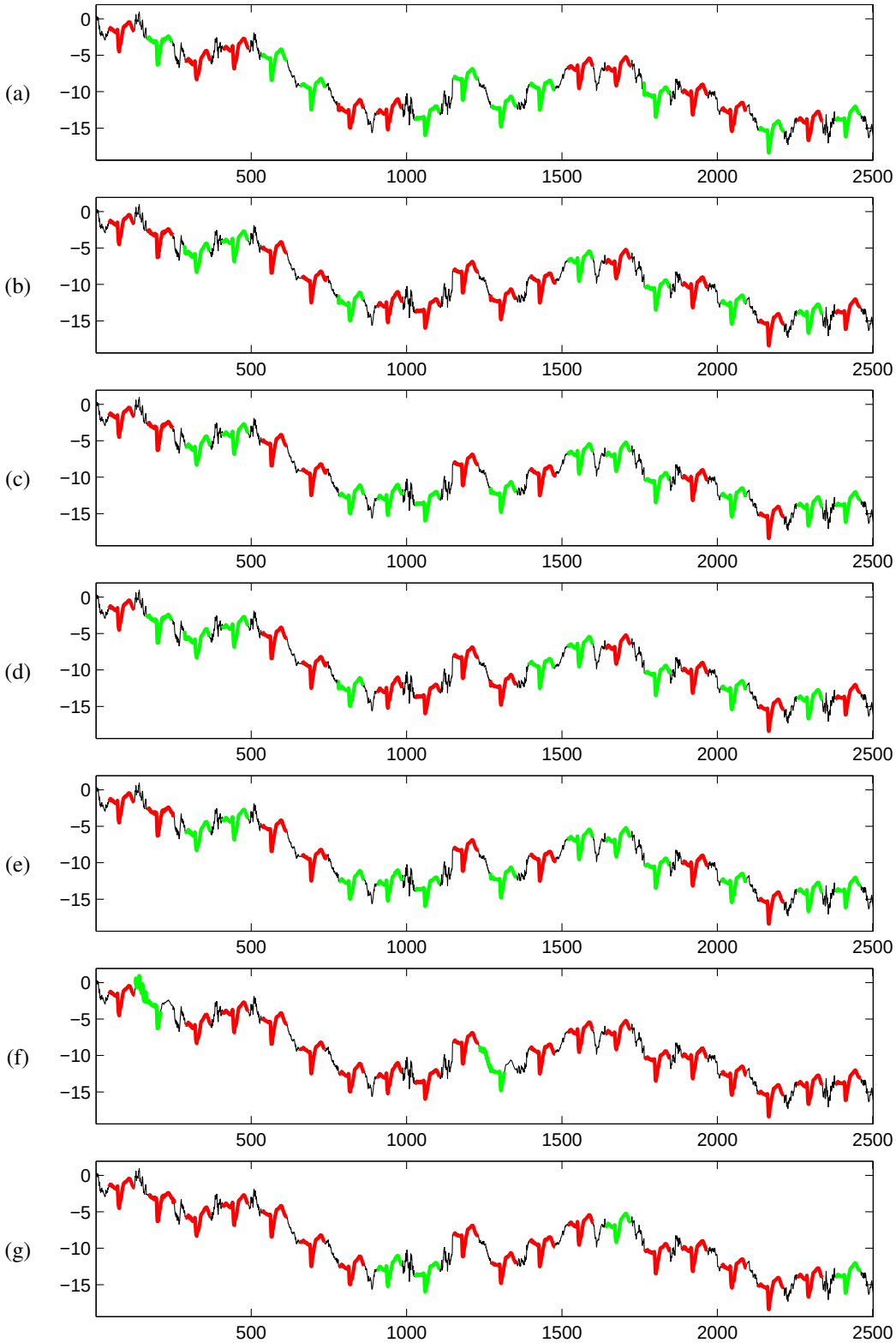

Figure B.11: TwoLeadECG dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

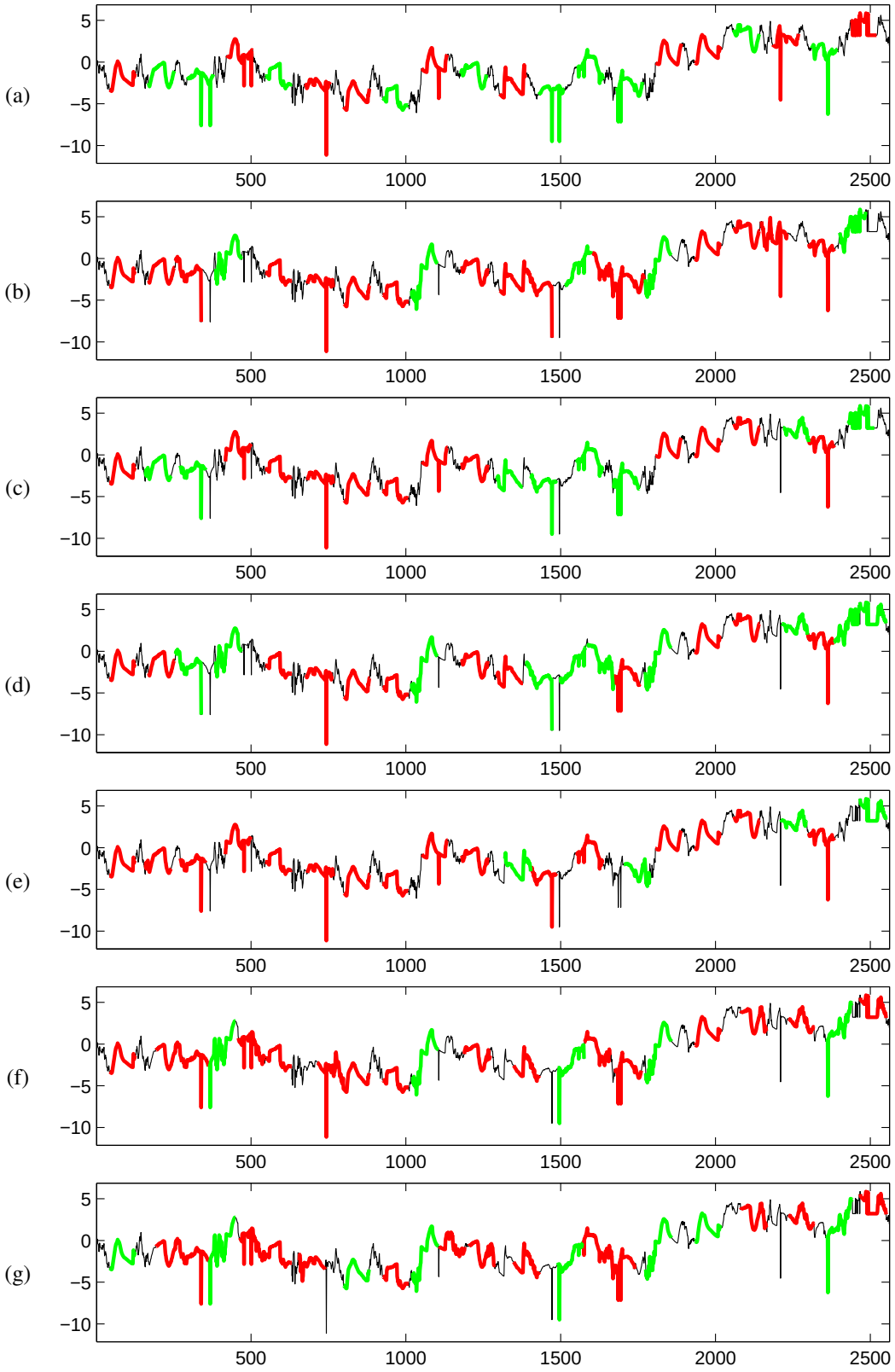

Figure B.12: MoteStrain dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

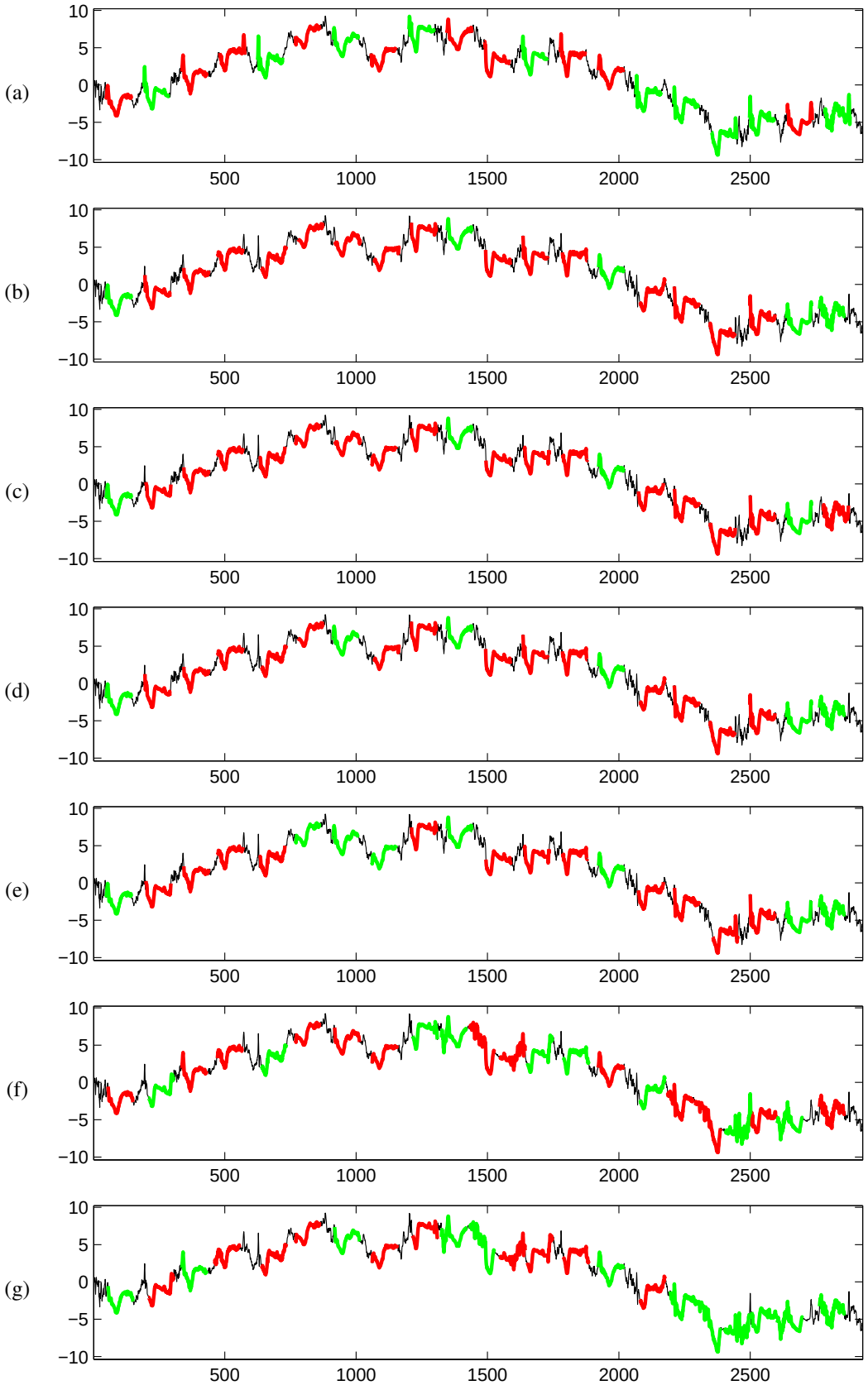

Figure B.13: ECG200 dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

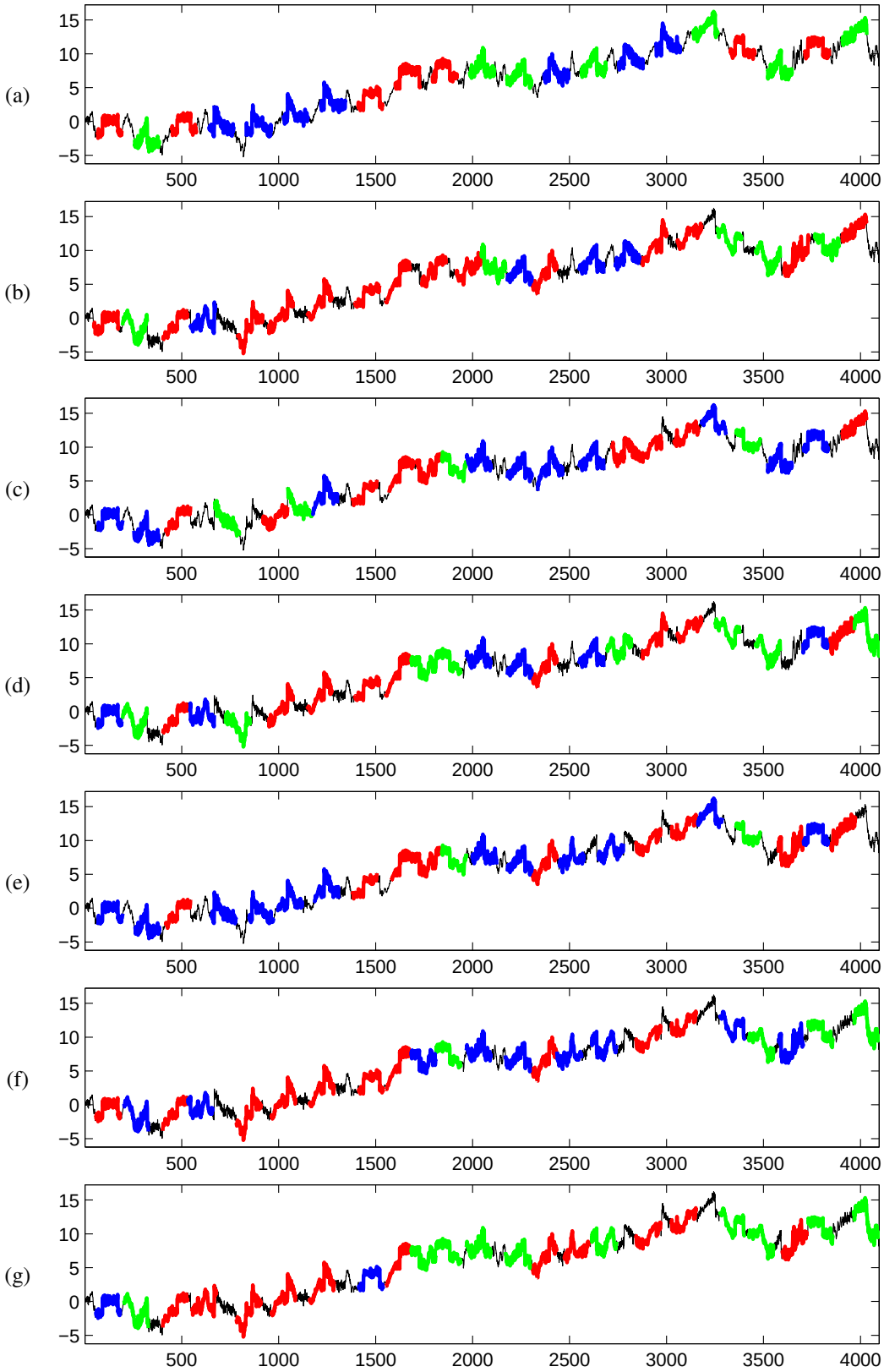

Figure B.14: CBF dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

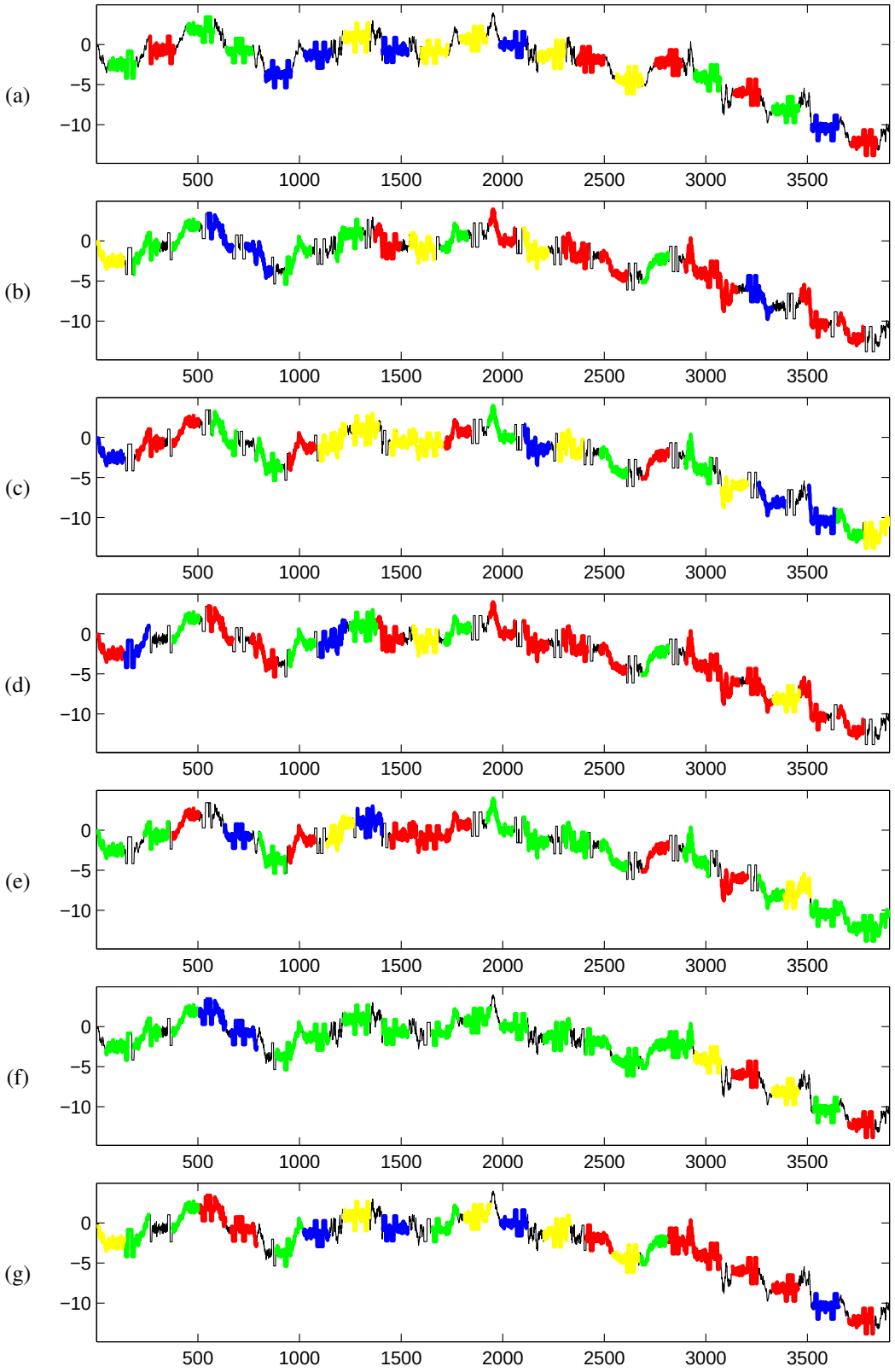

Figure B.15: Two_Patterns dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

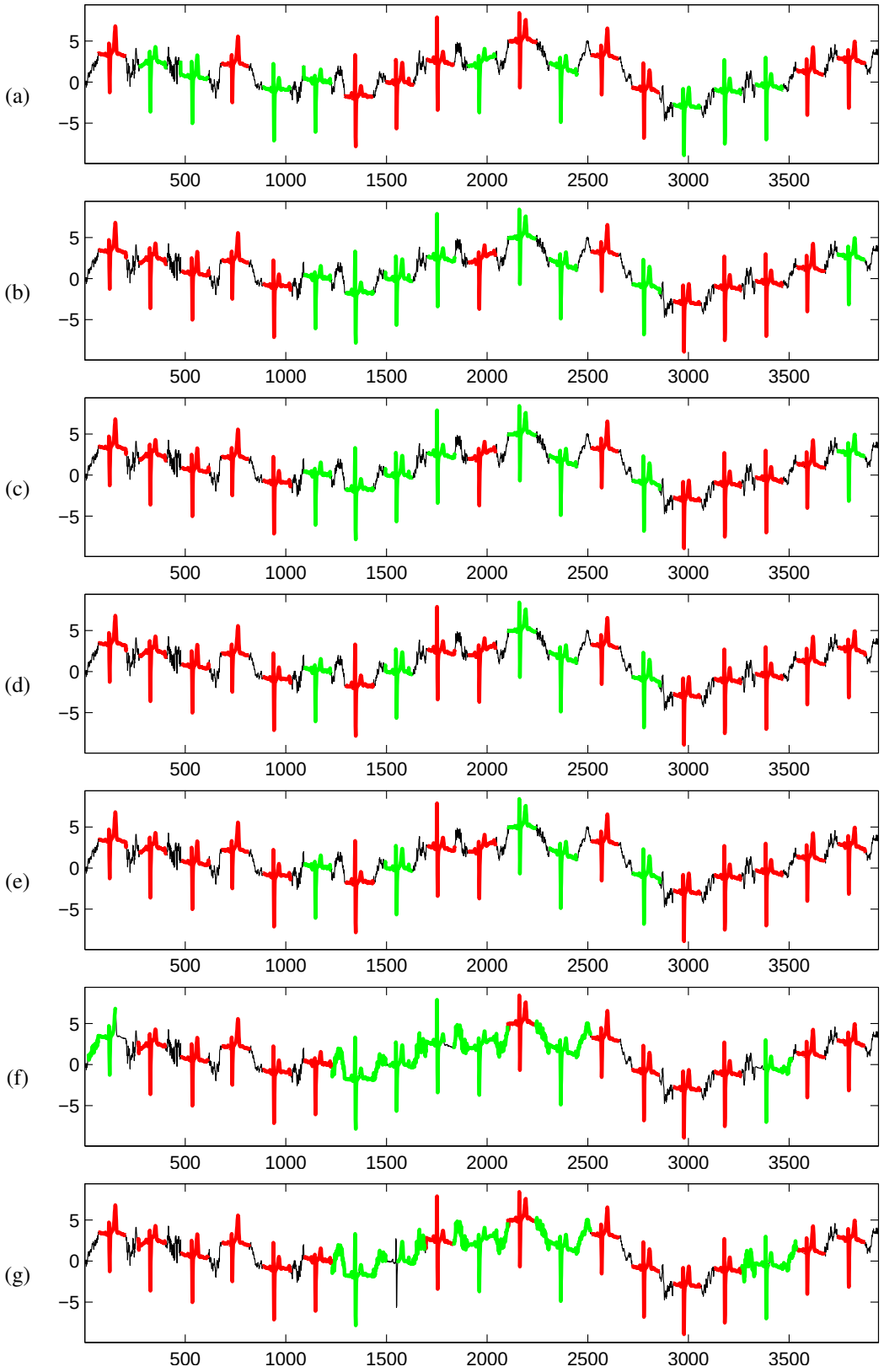

Figure B.16: ECGFiveDays dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

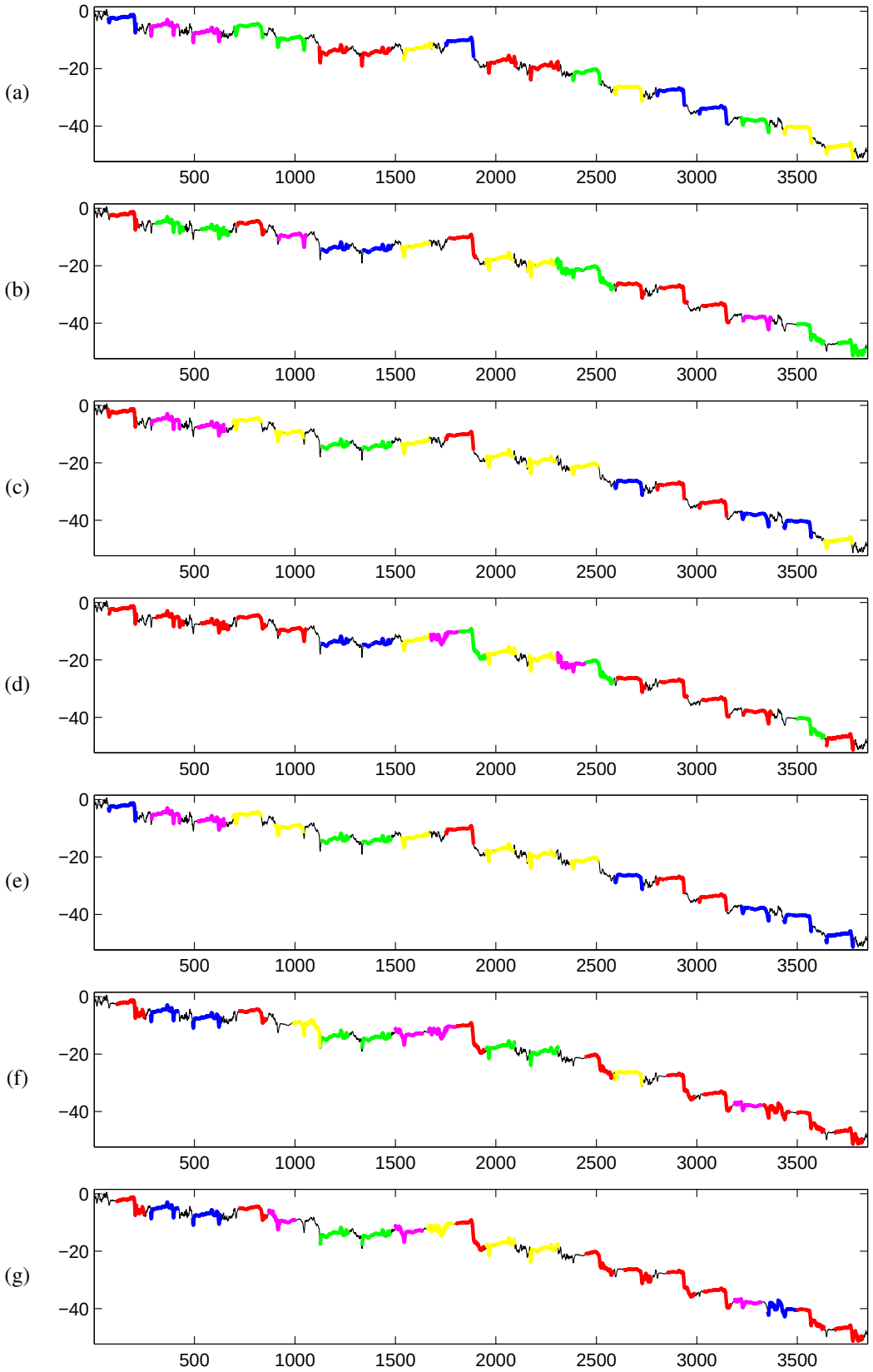

Figure B.17: ECG5000 dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

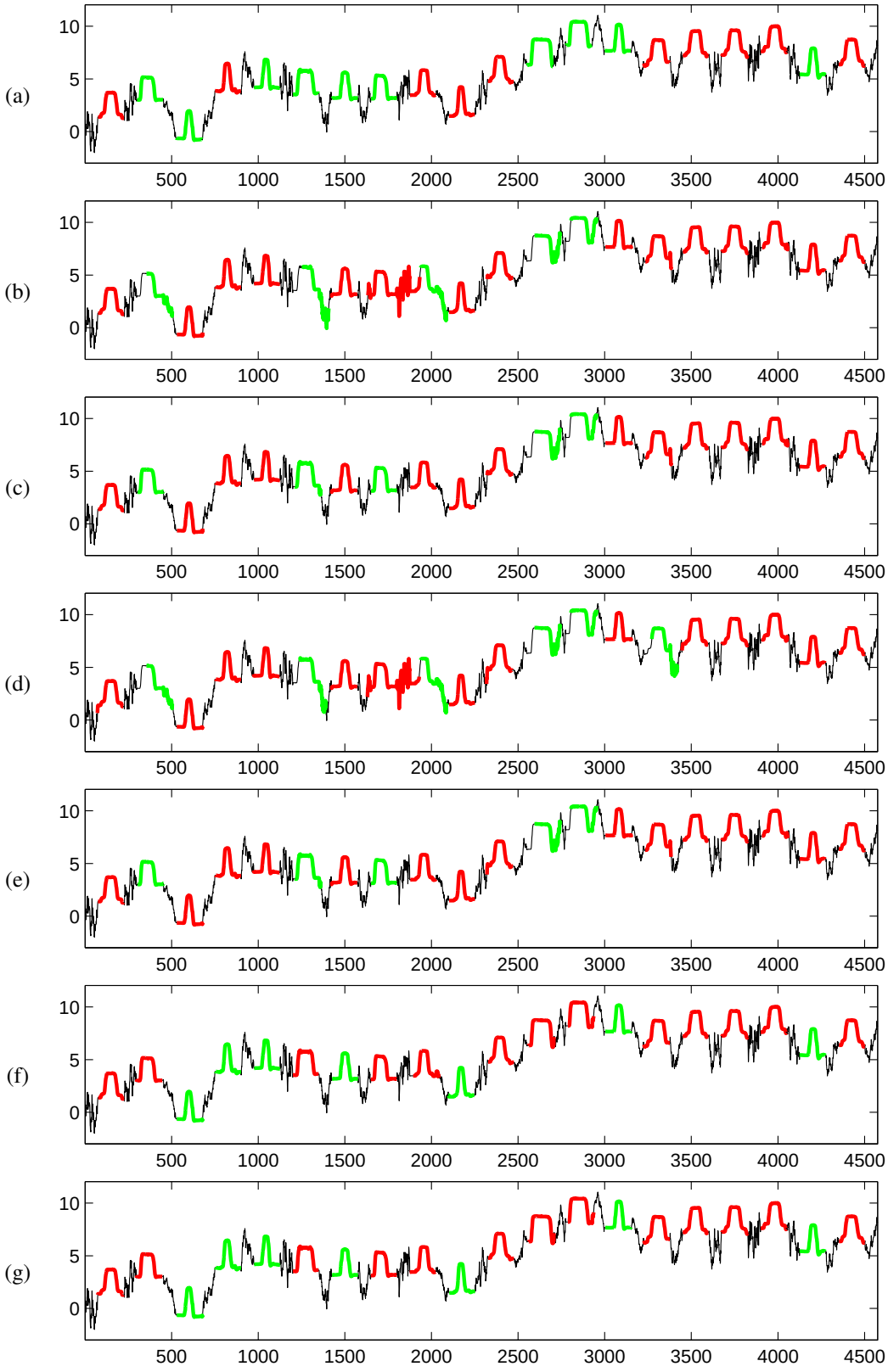

Figure B.18: Gun_Point dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

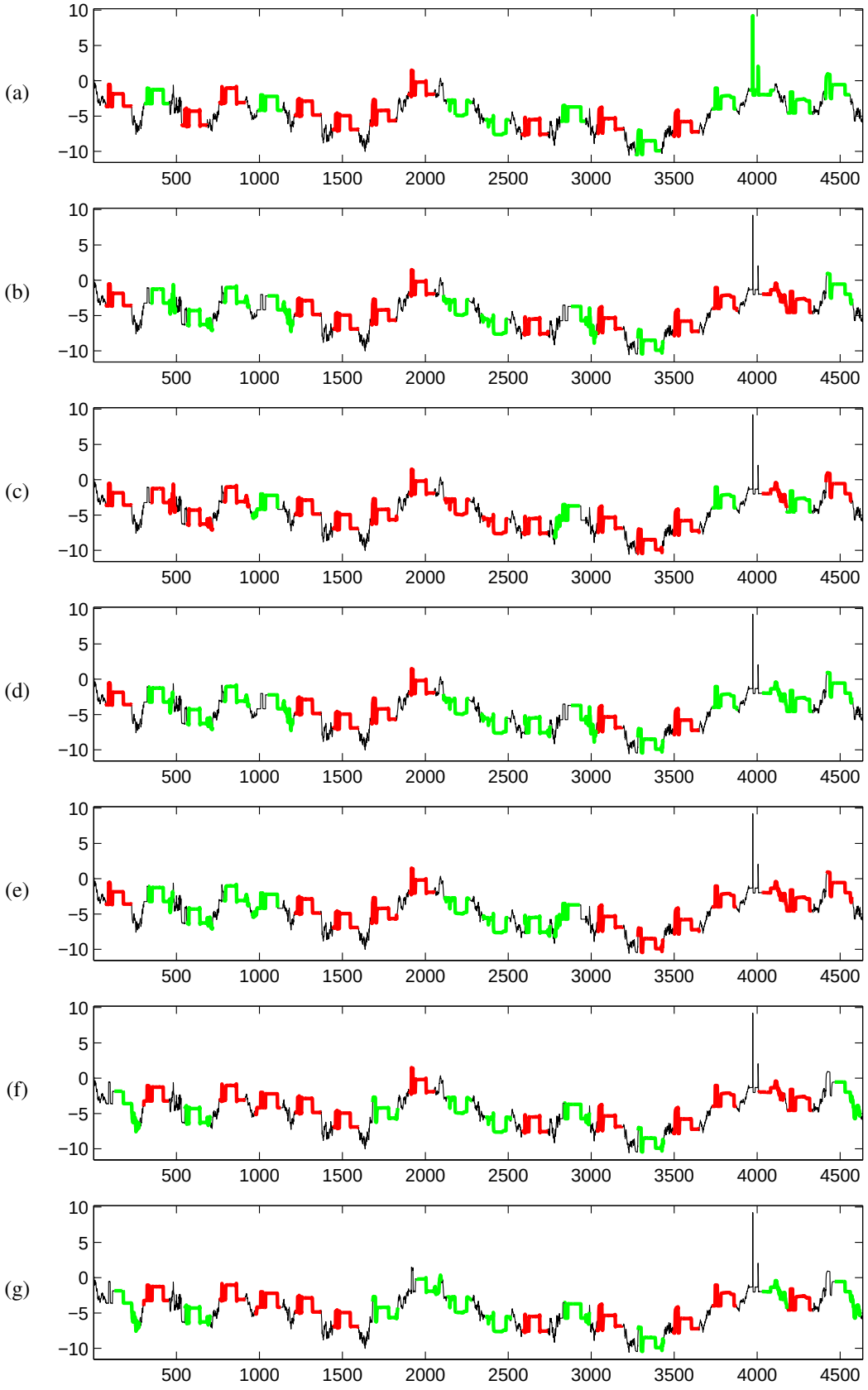

Figure B.19: wafer dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

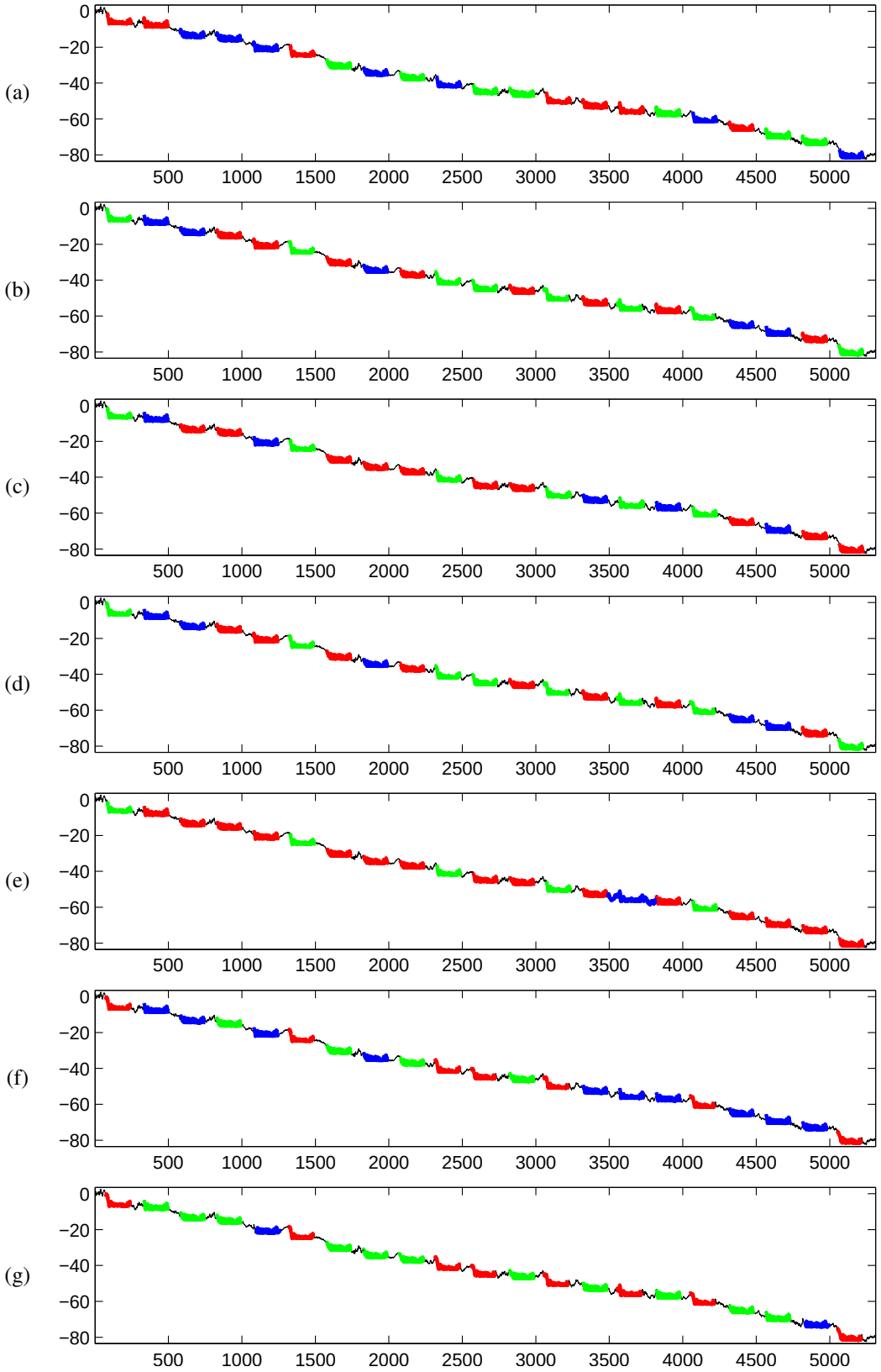

Figure B.20: ChlorineConcentration dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

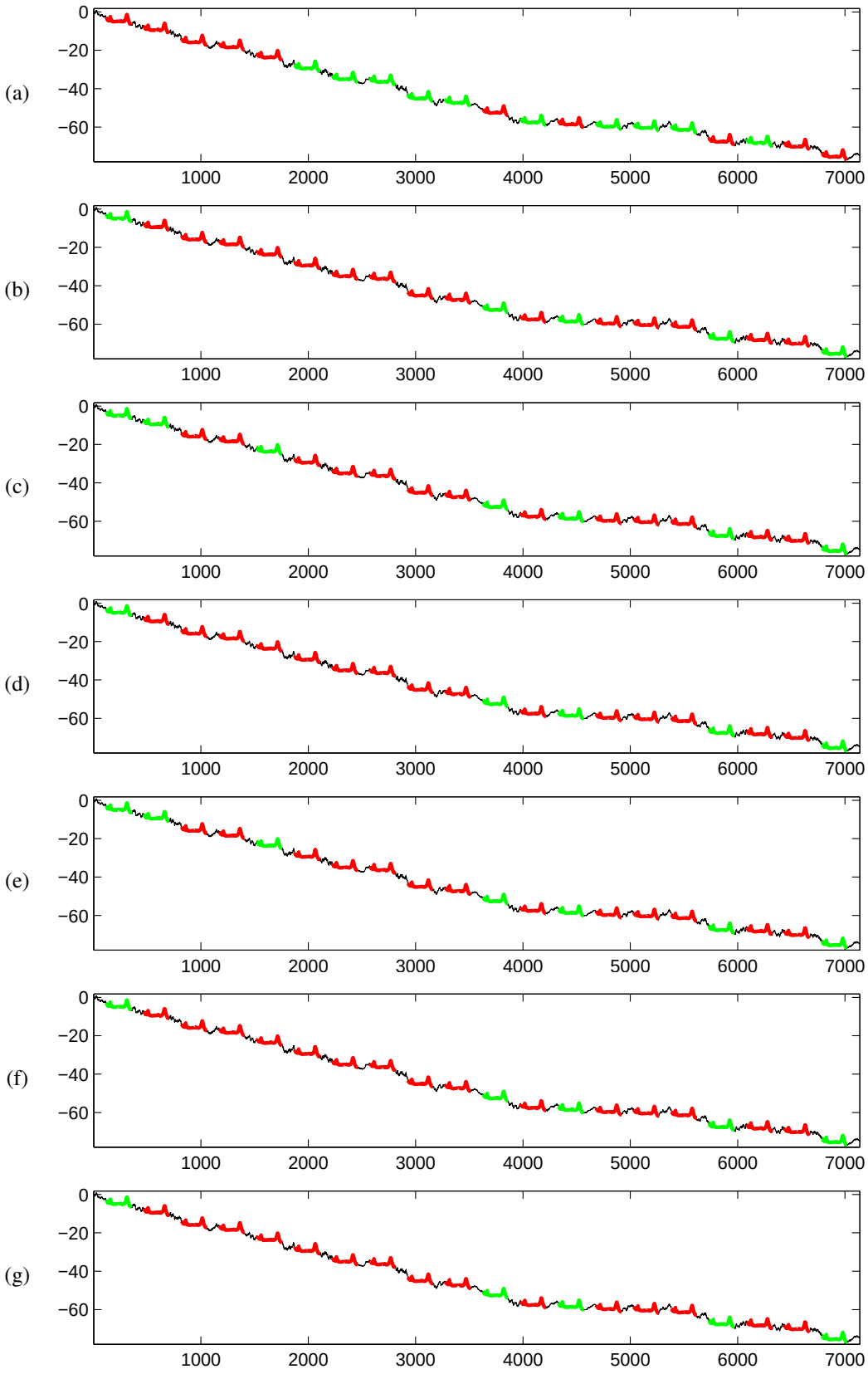

Figure B.21: Wine dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

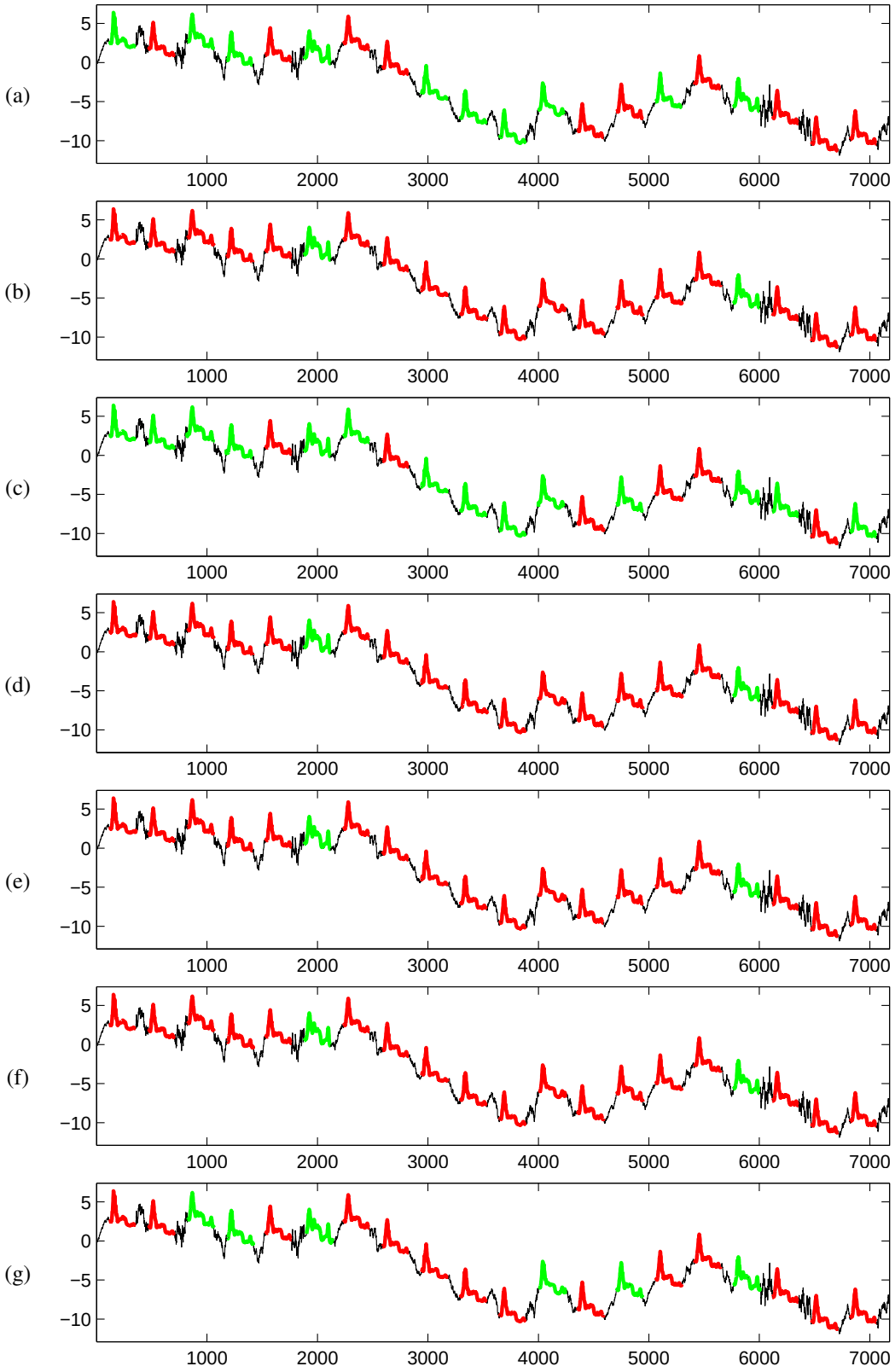

Figure B.22: Strawberry dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

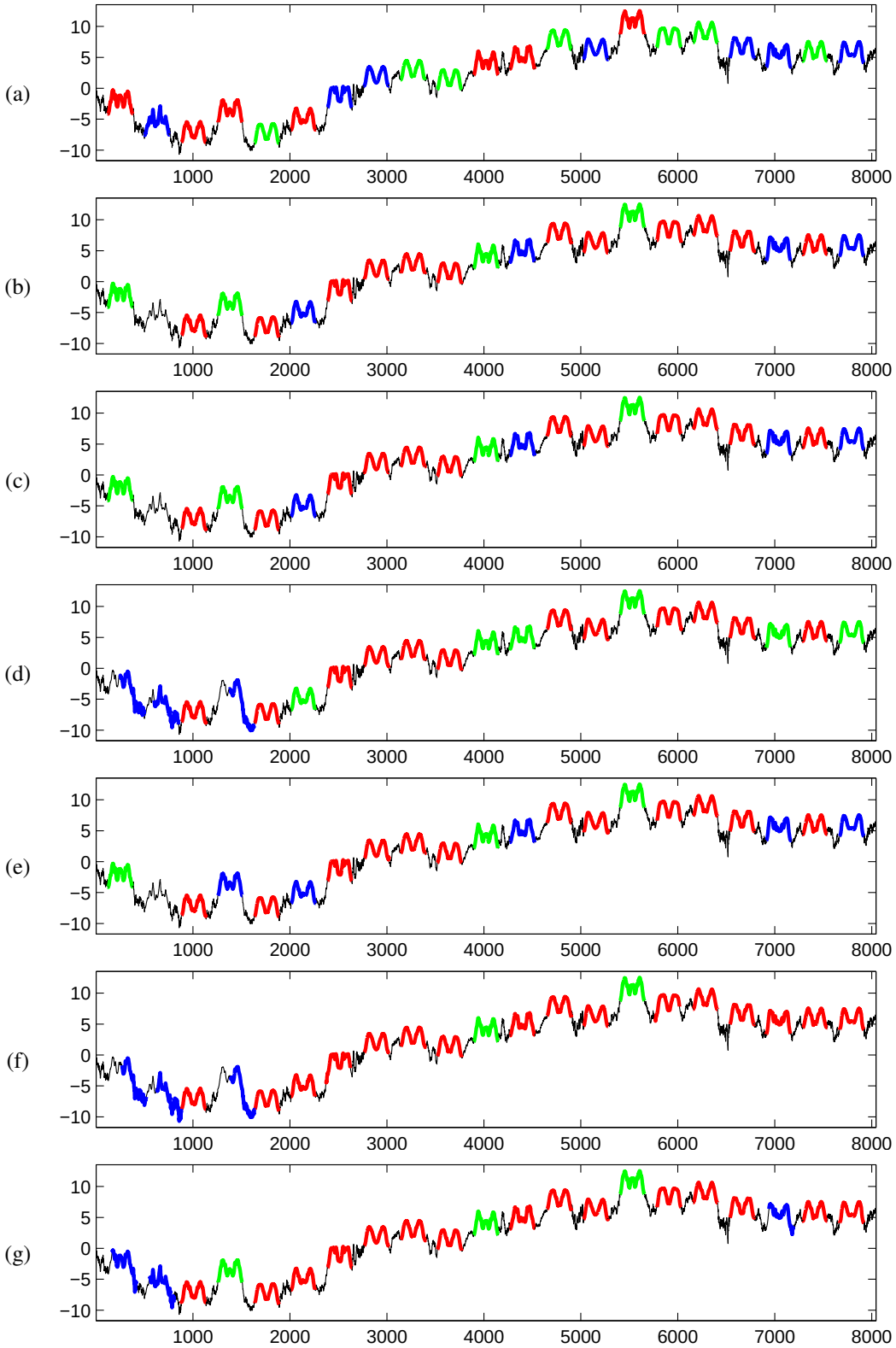

Figure B.23: ArrowHead dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

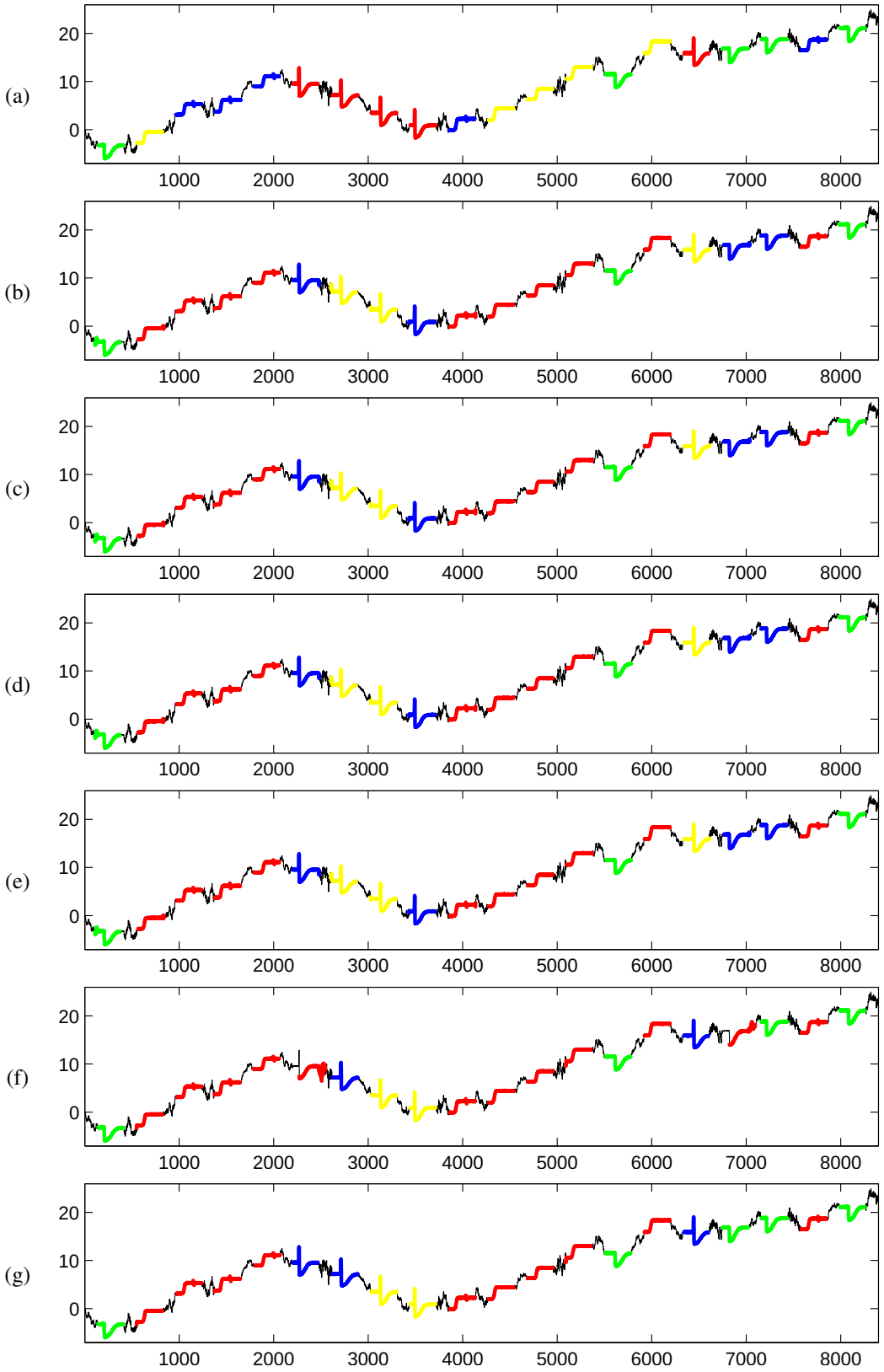

Figure B.24: Trace dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

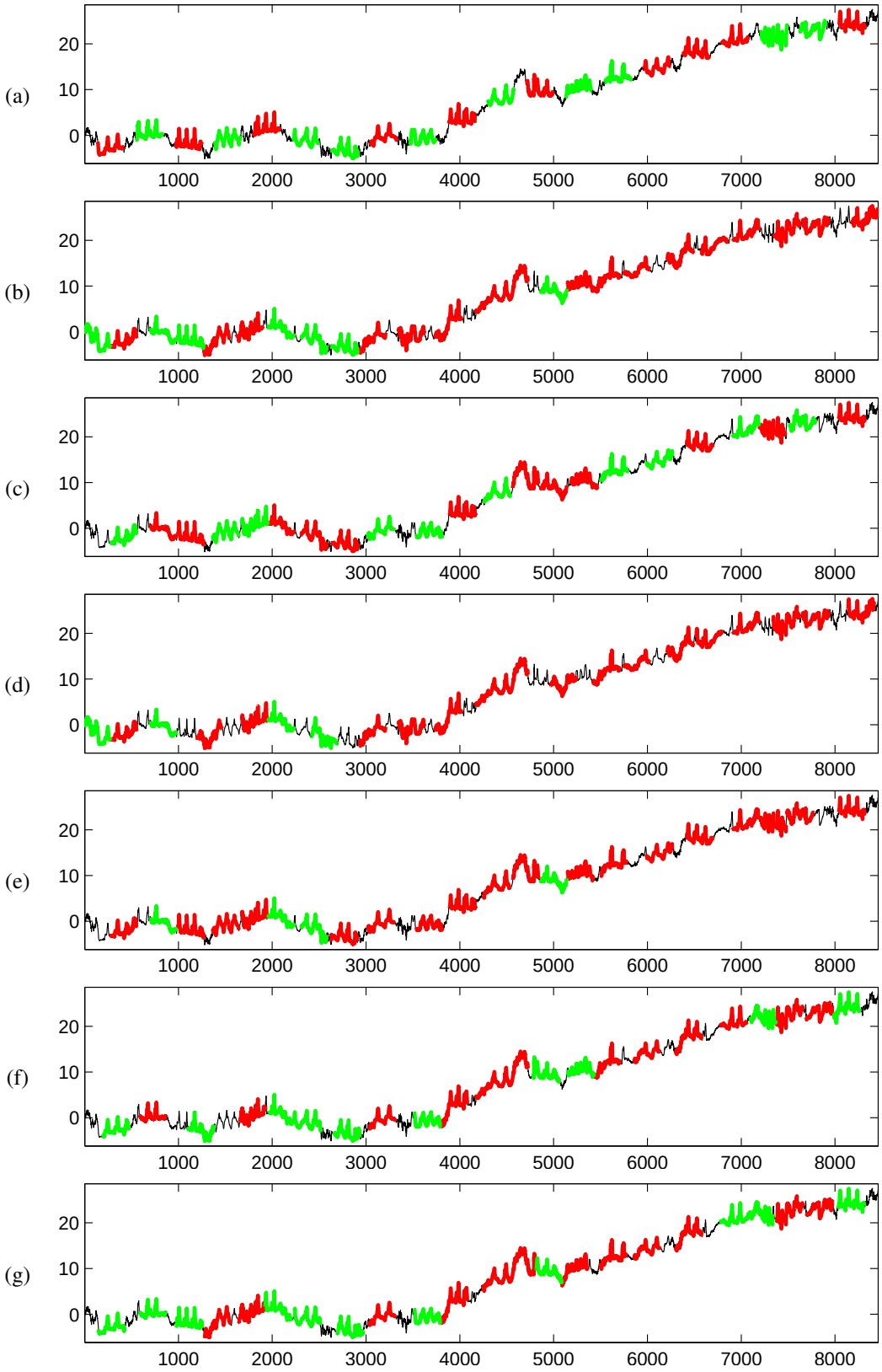

Figure B.25: ToeSegmentation1 dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

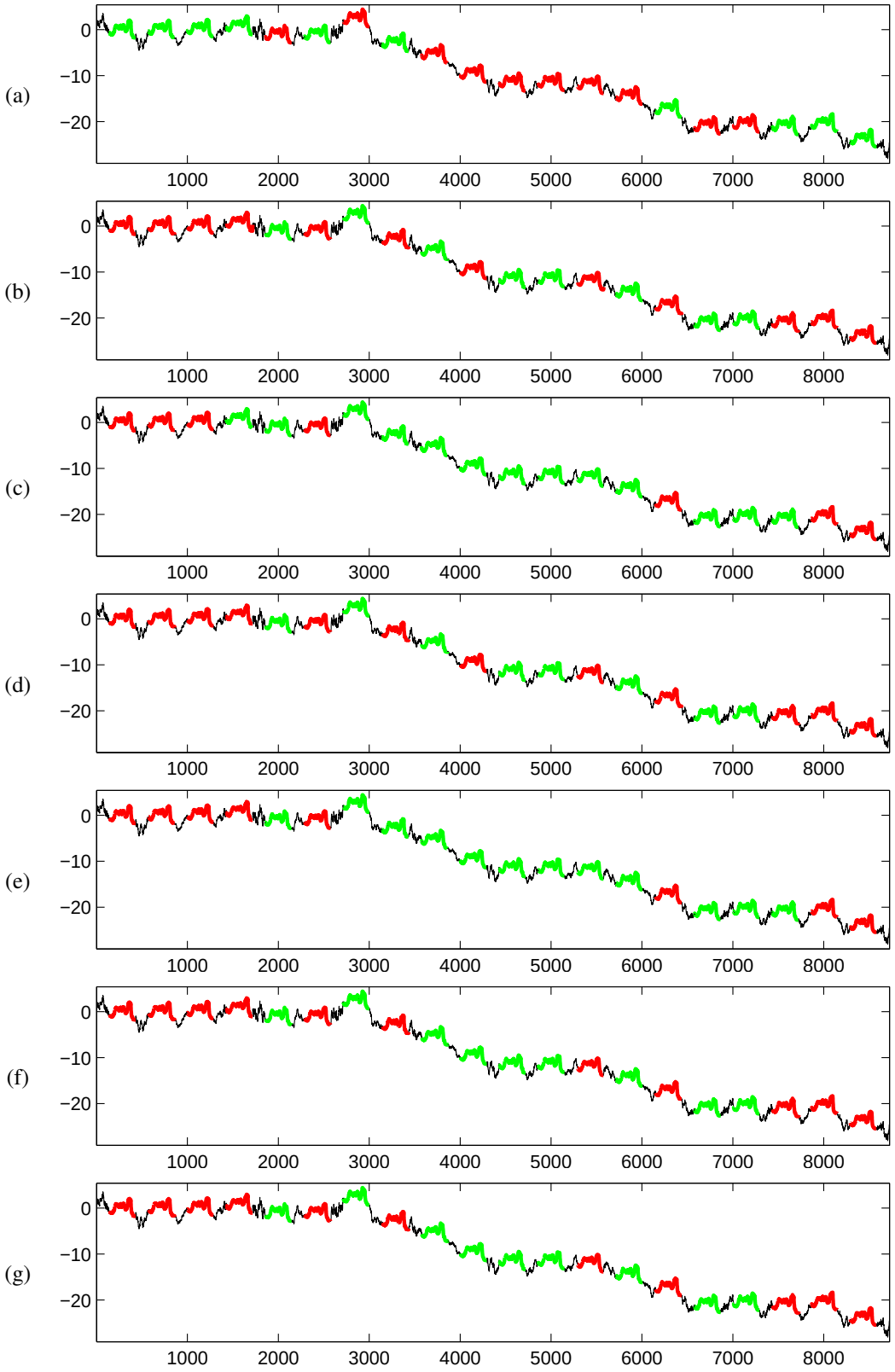

Figure B.26: Coffee dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

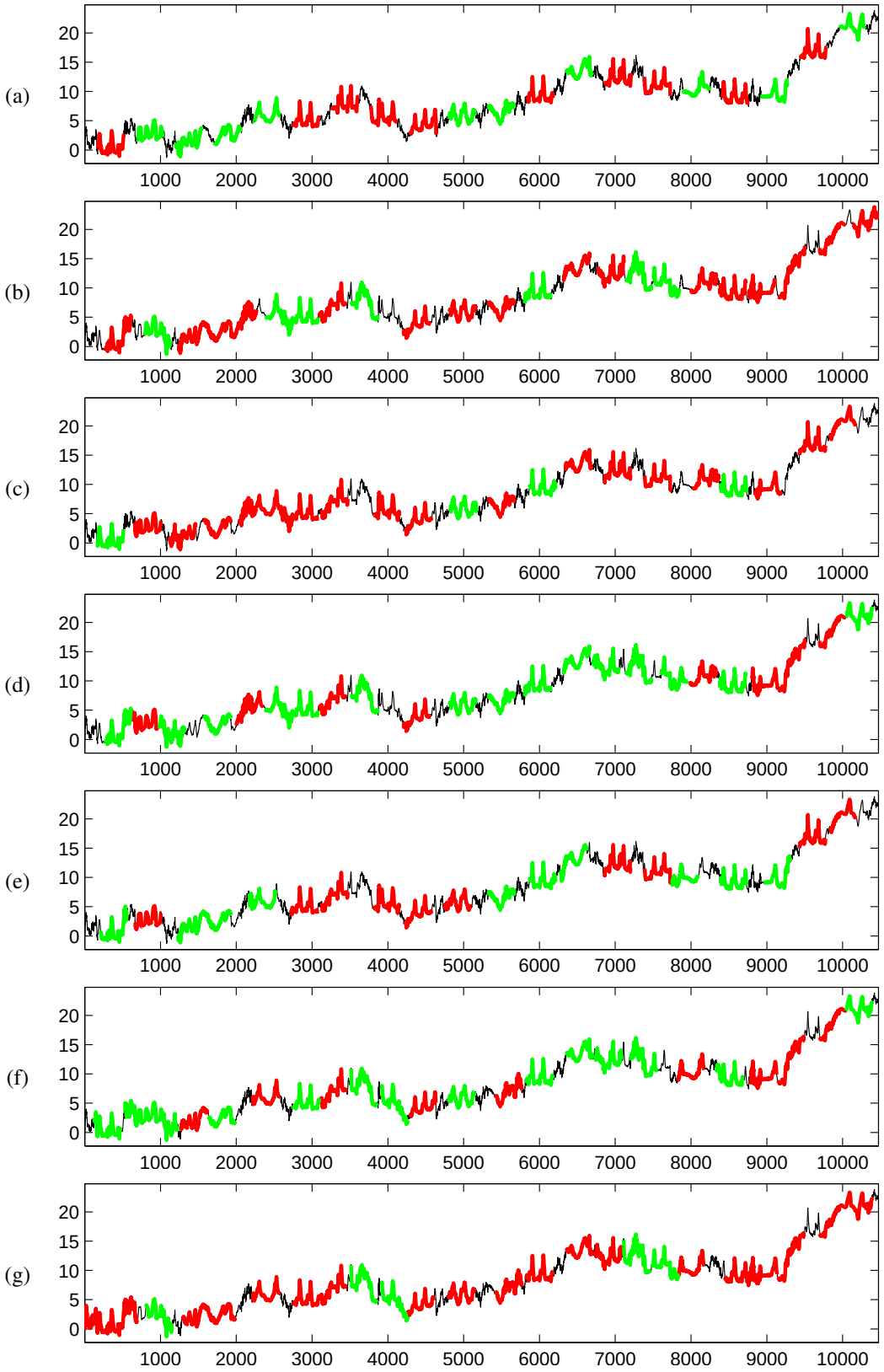

Figure B.27: ToeSegmentation2 dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

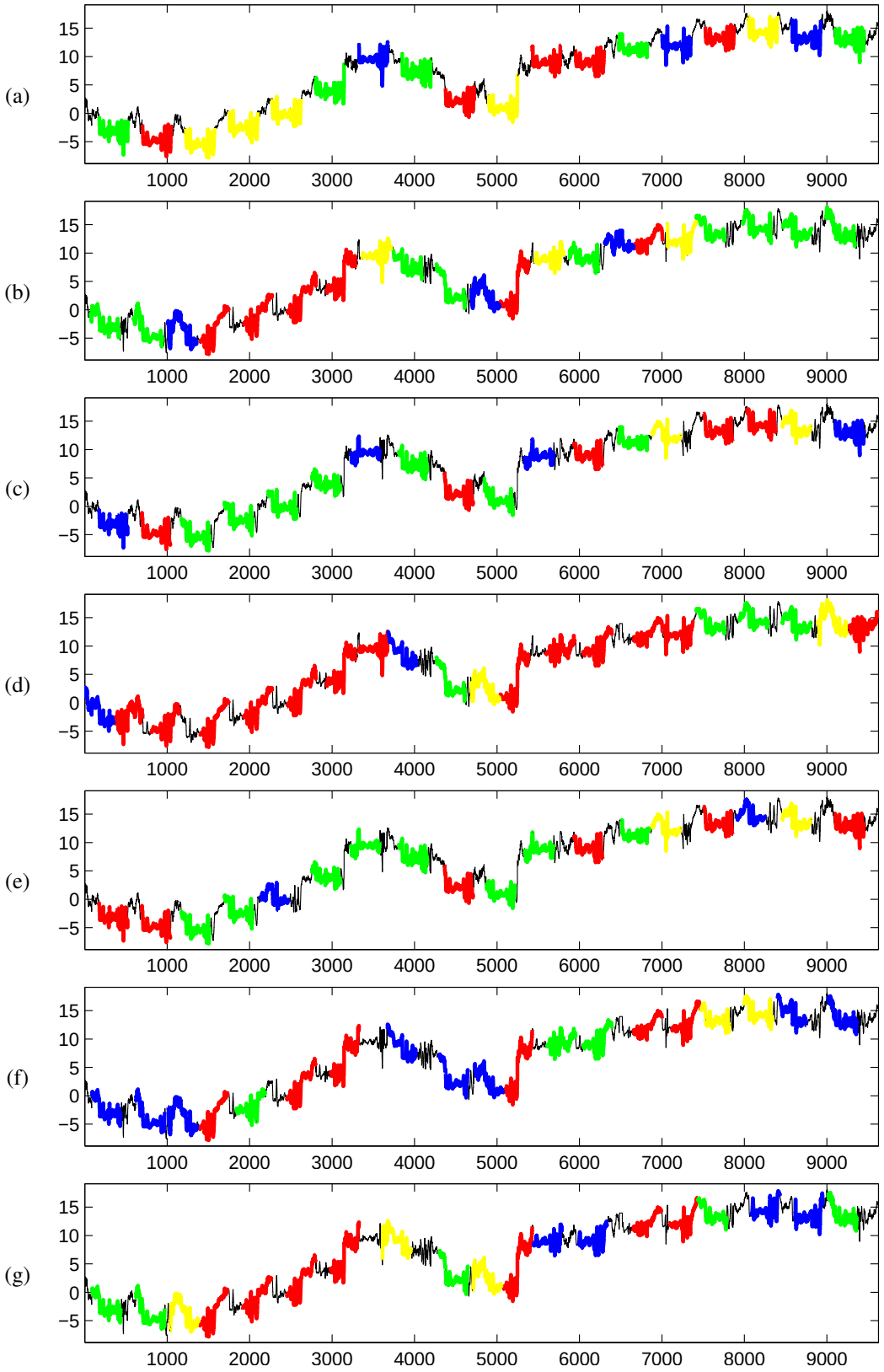

Figure B.28: FaceFour dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d), (e), (f) and (g) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z, E-SA-L, D-AA-Z and D-SA-Z, respectively.

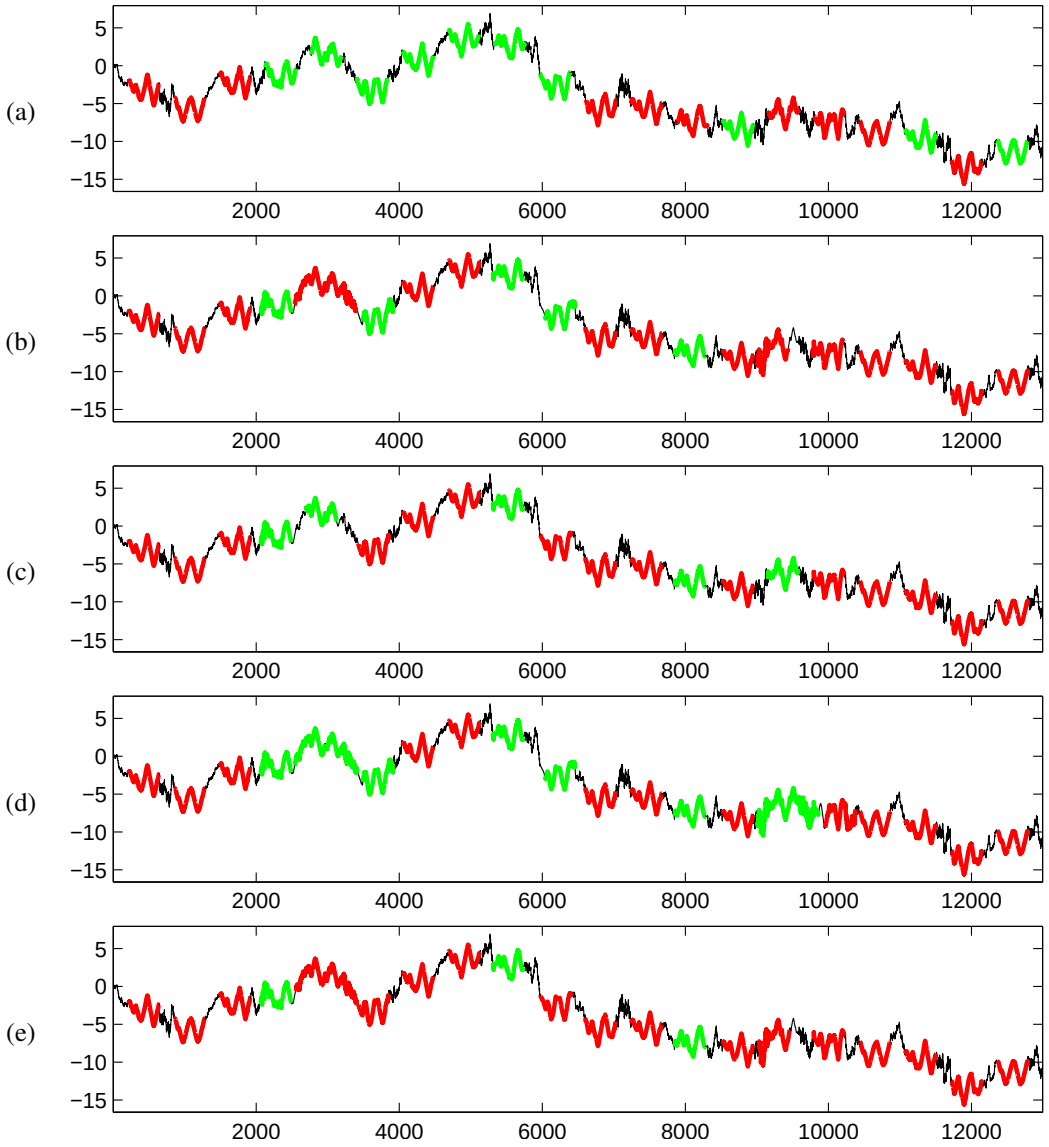

Figure B.29: yoga dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

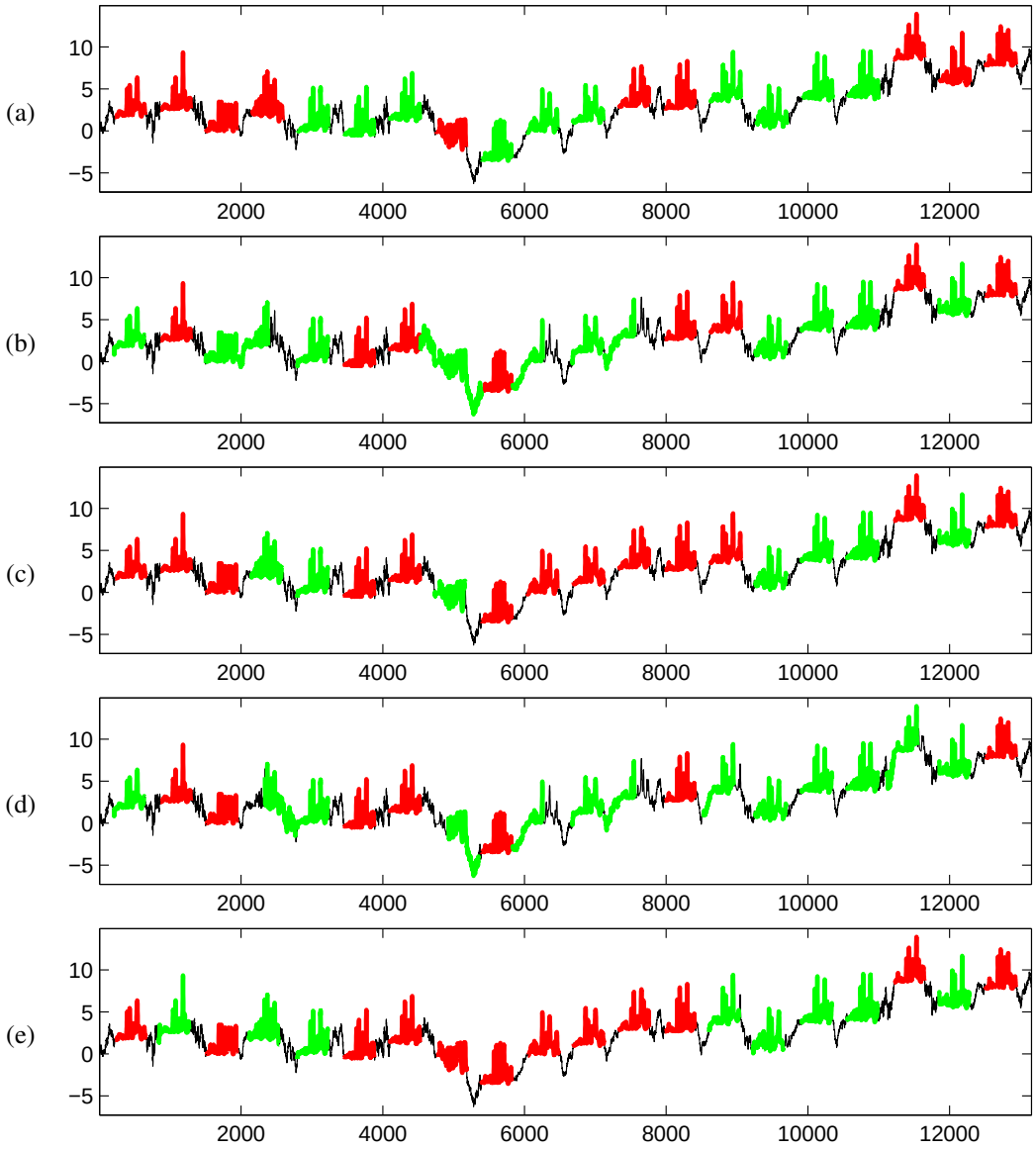

Figure B.30: Ham dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

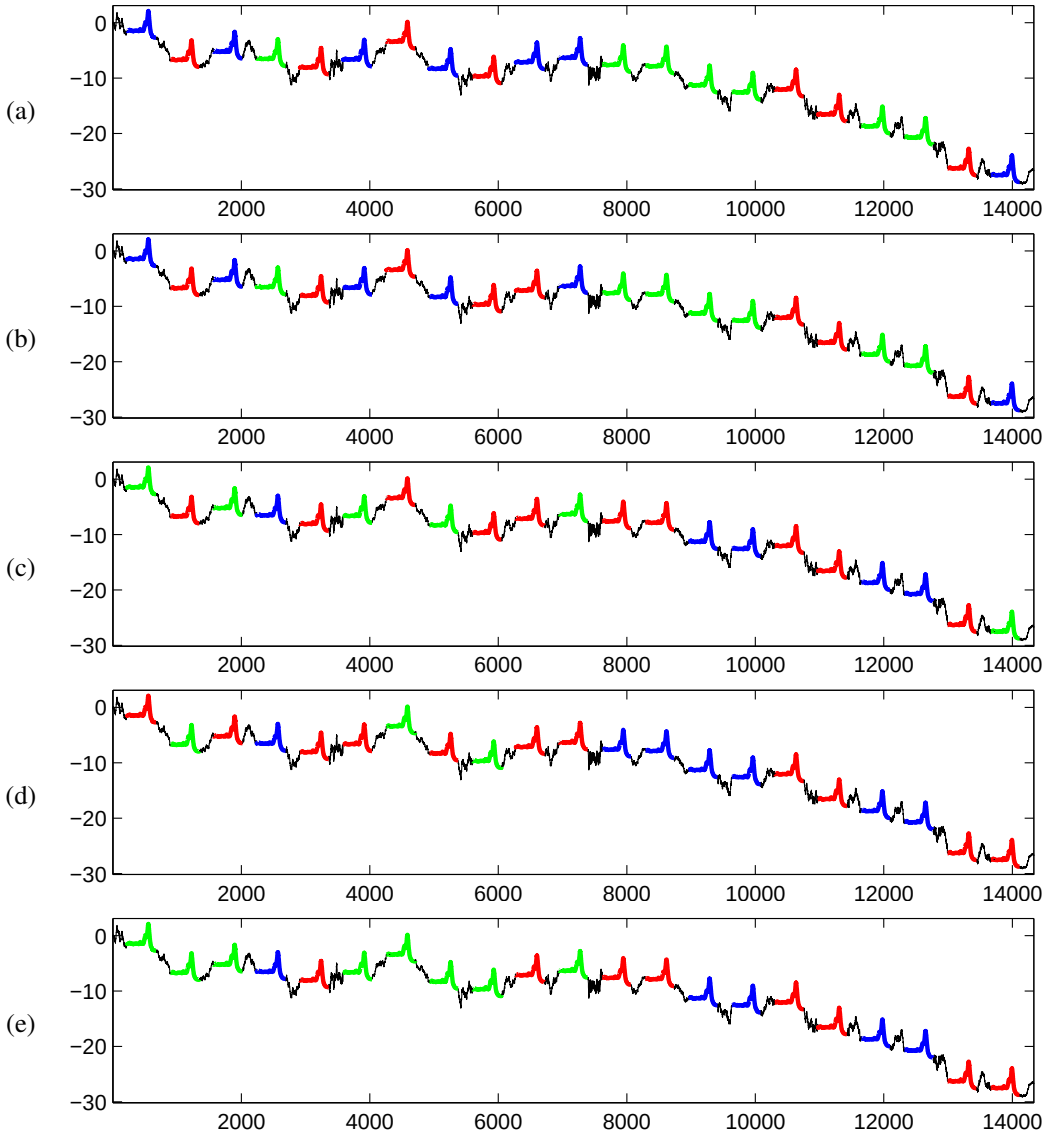

Figure B.31: Meat dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

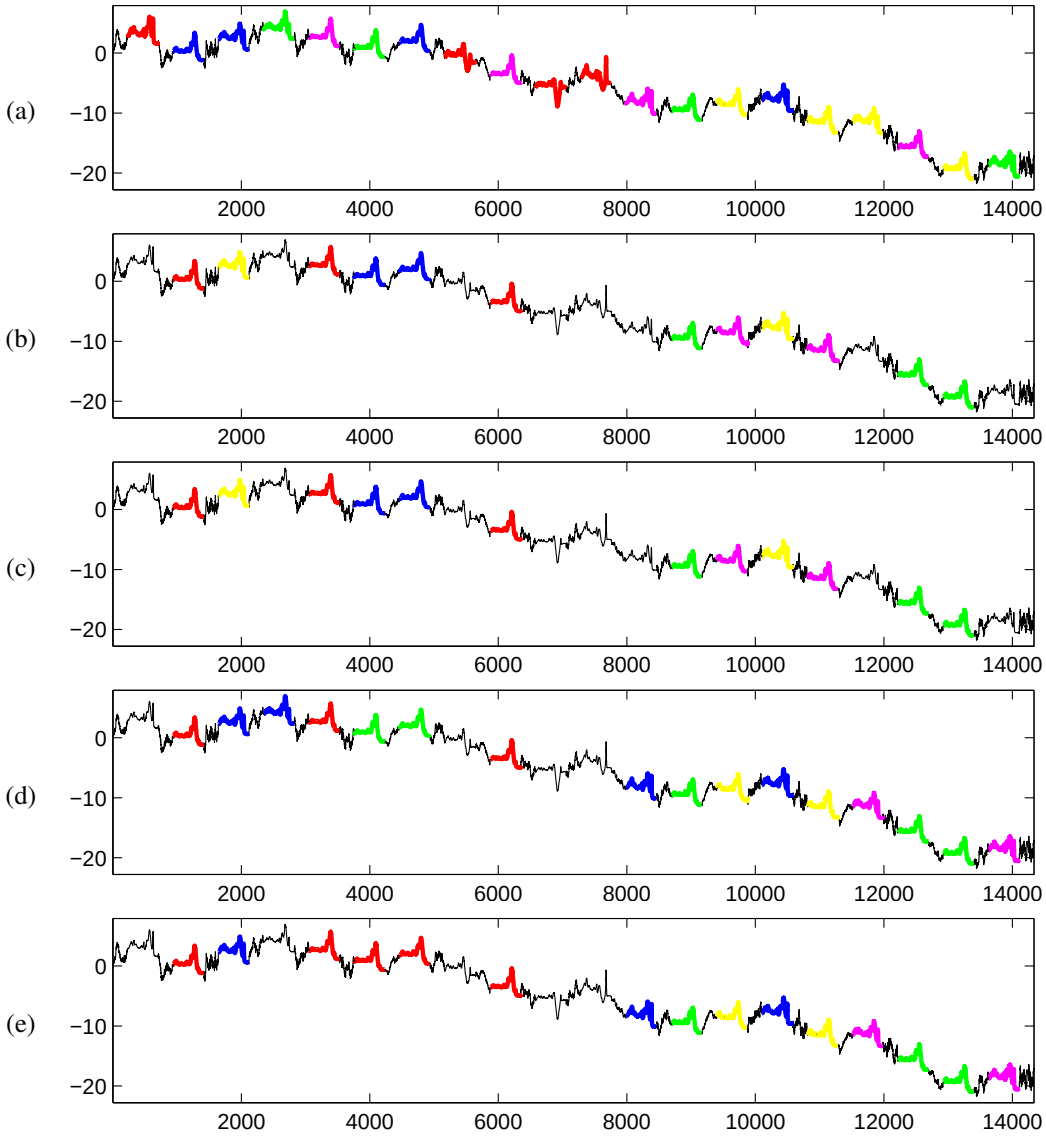

Figure B.32: Beef dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

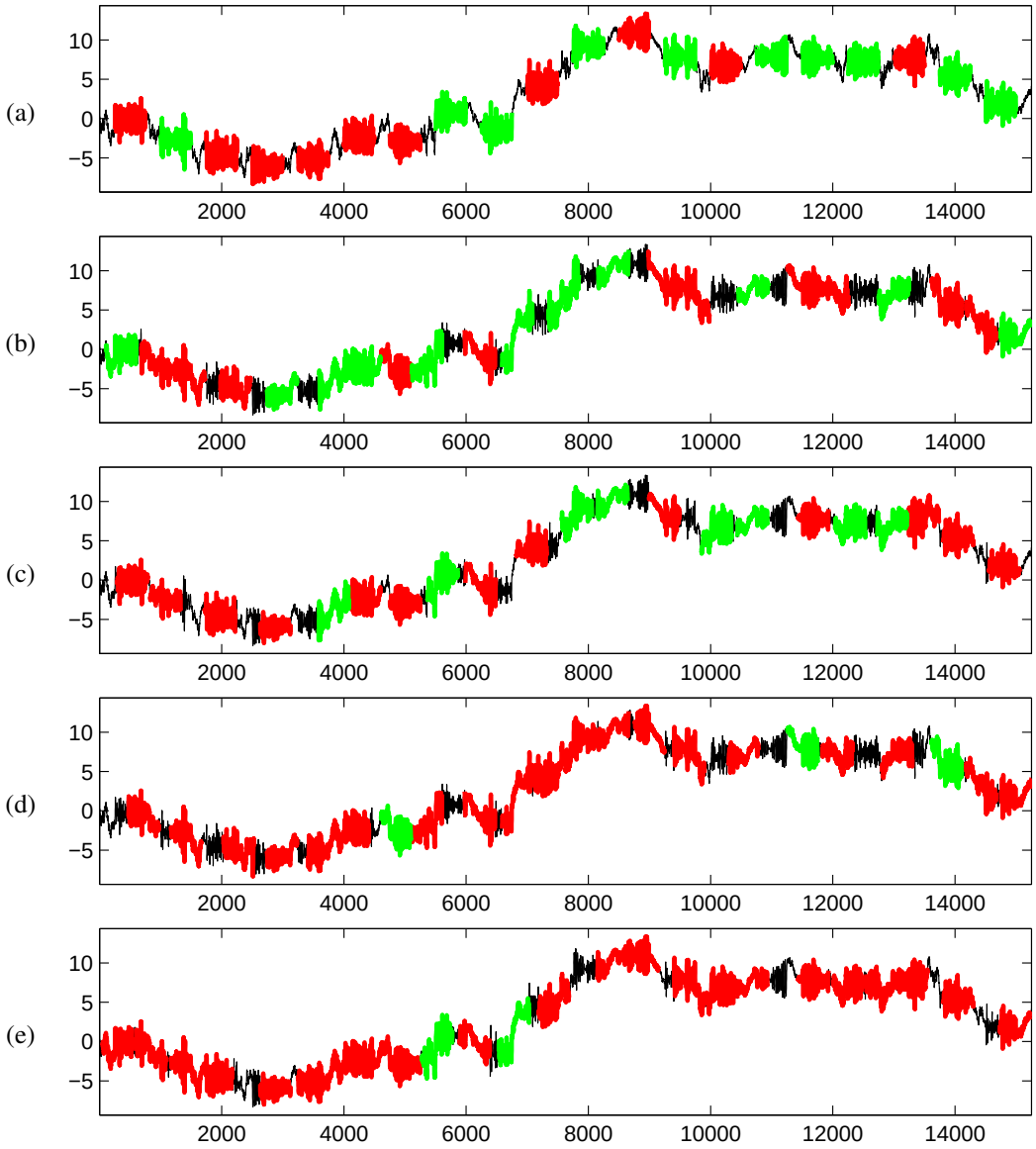

Figure B.33: FordA dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

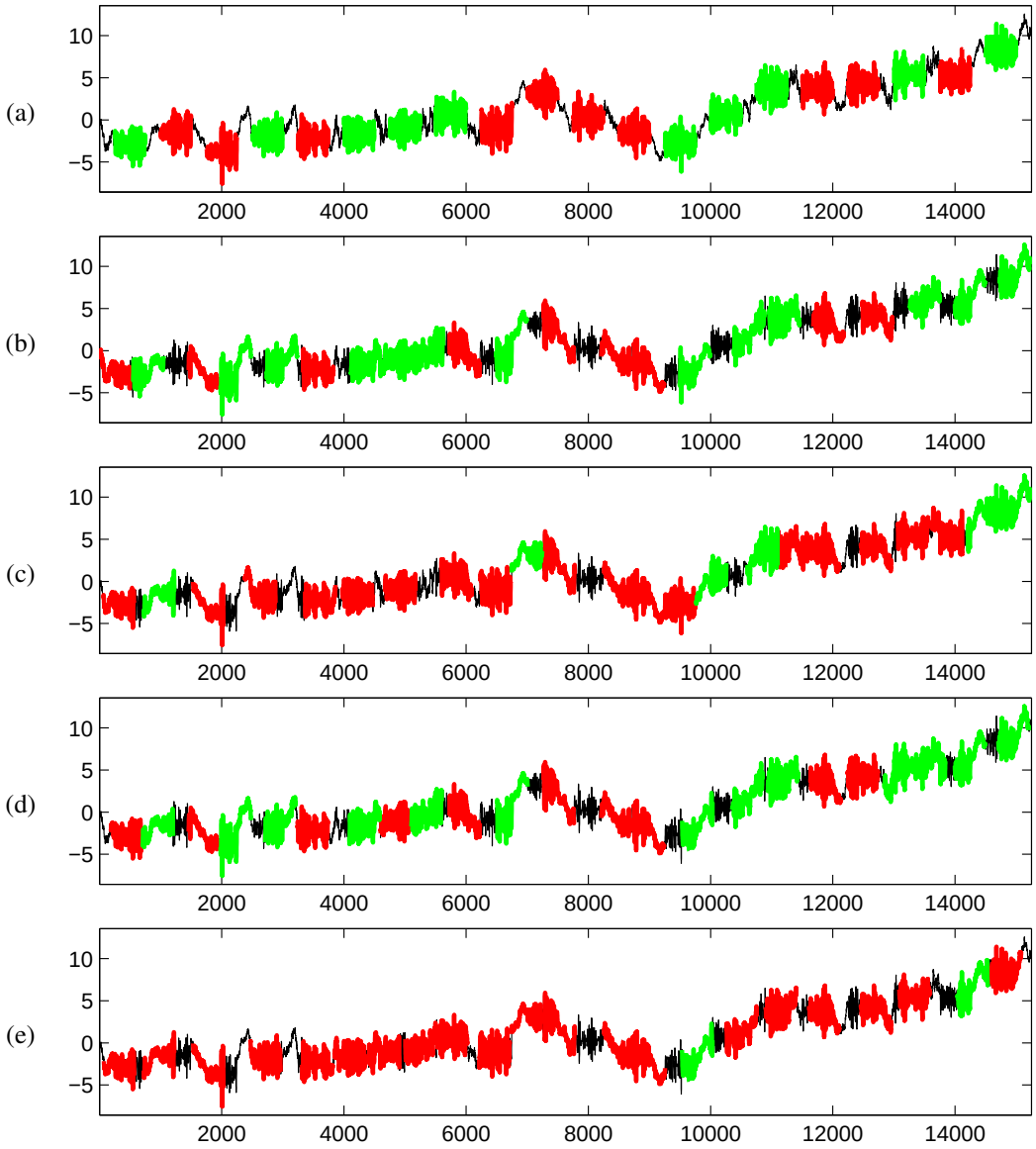

Figure B.34: FordB dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

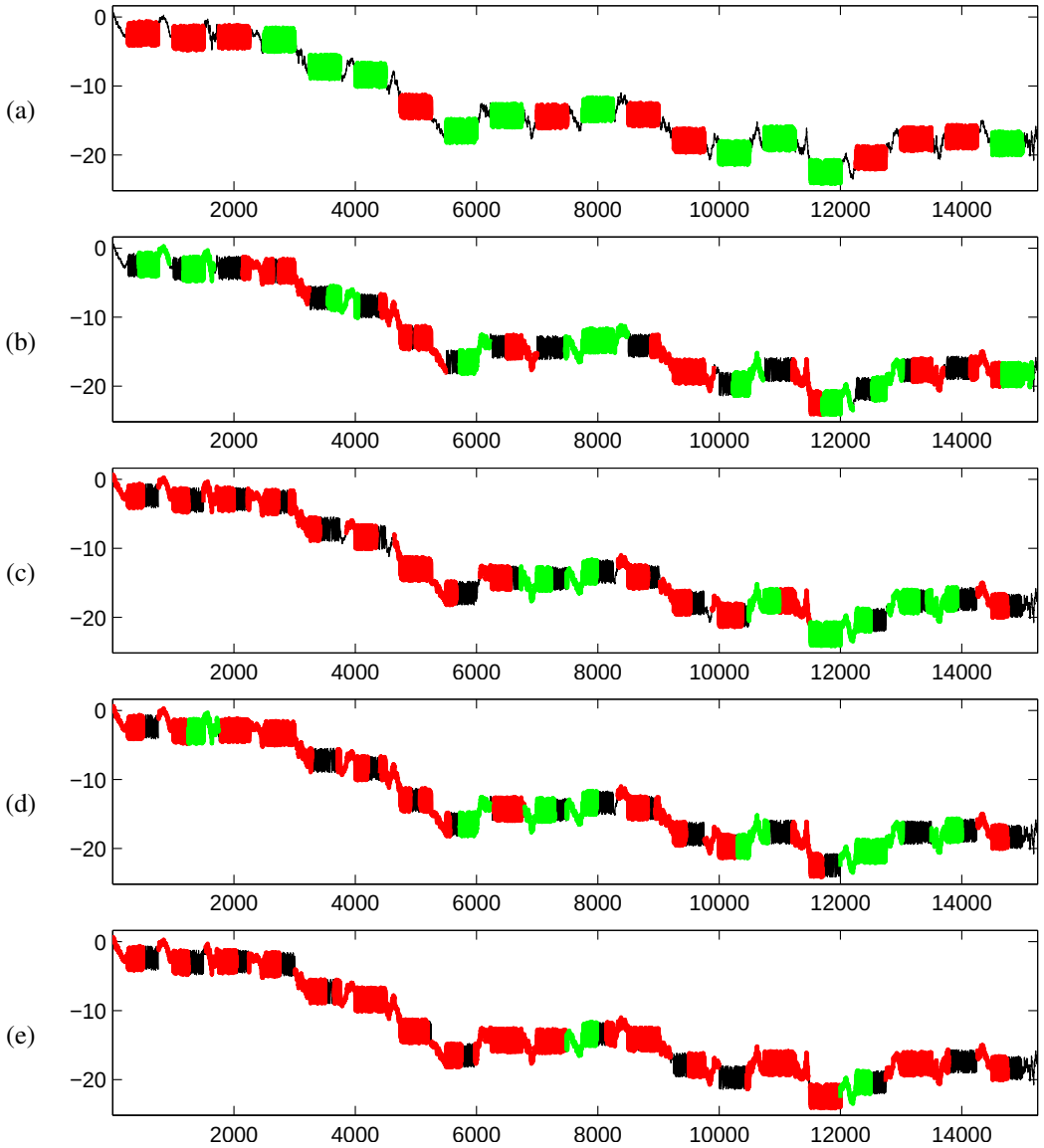

Figure B.35: ShapeletSim dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

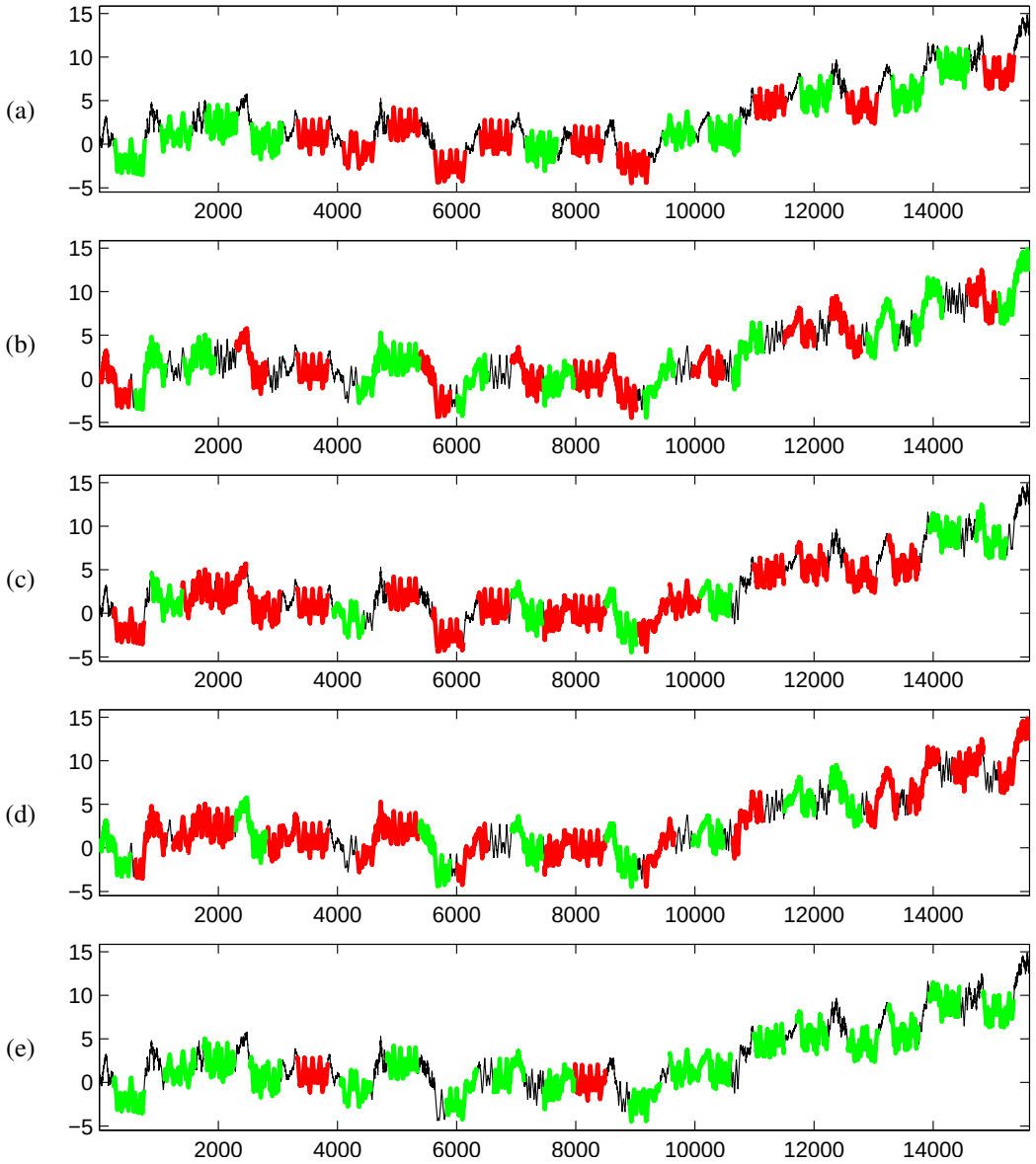

Figure B.36: BeetleFly dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

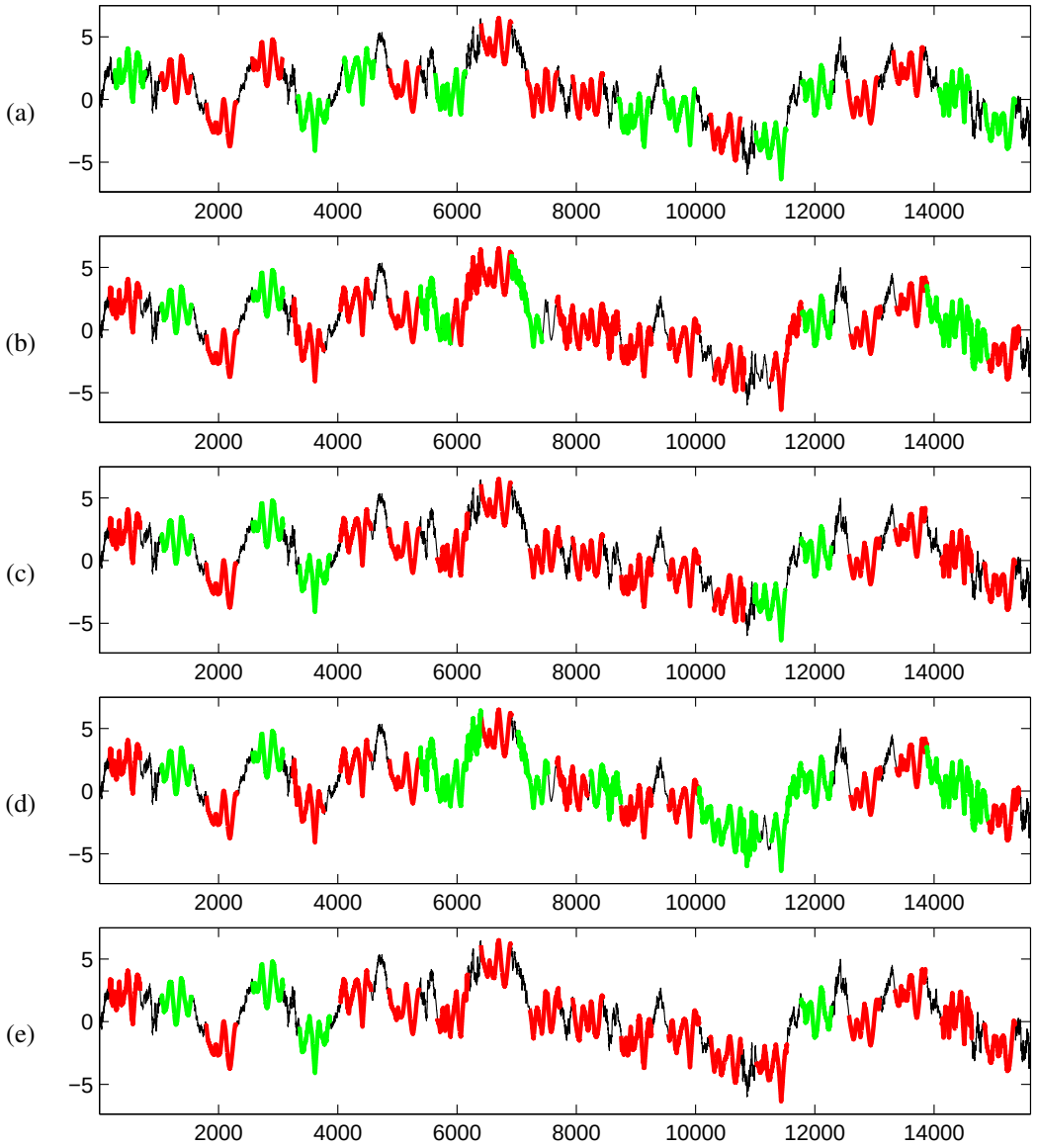

Figure B.37: BirdChicken dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

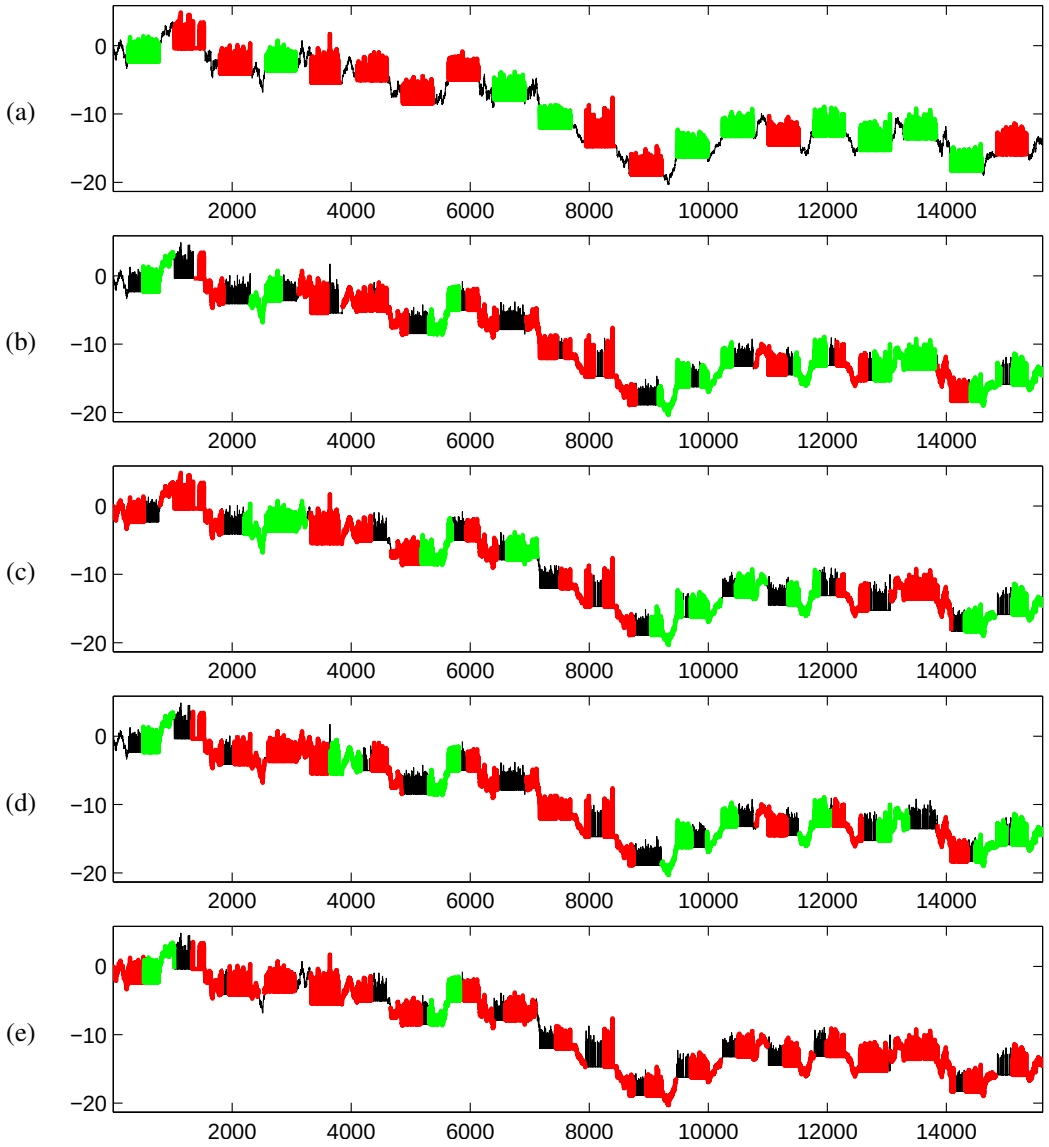

Figure B.38: Earthquakes dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

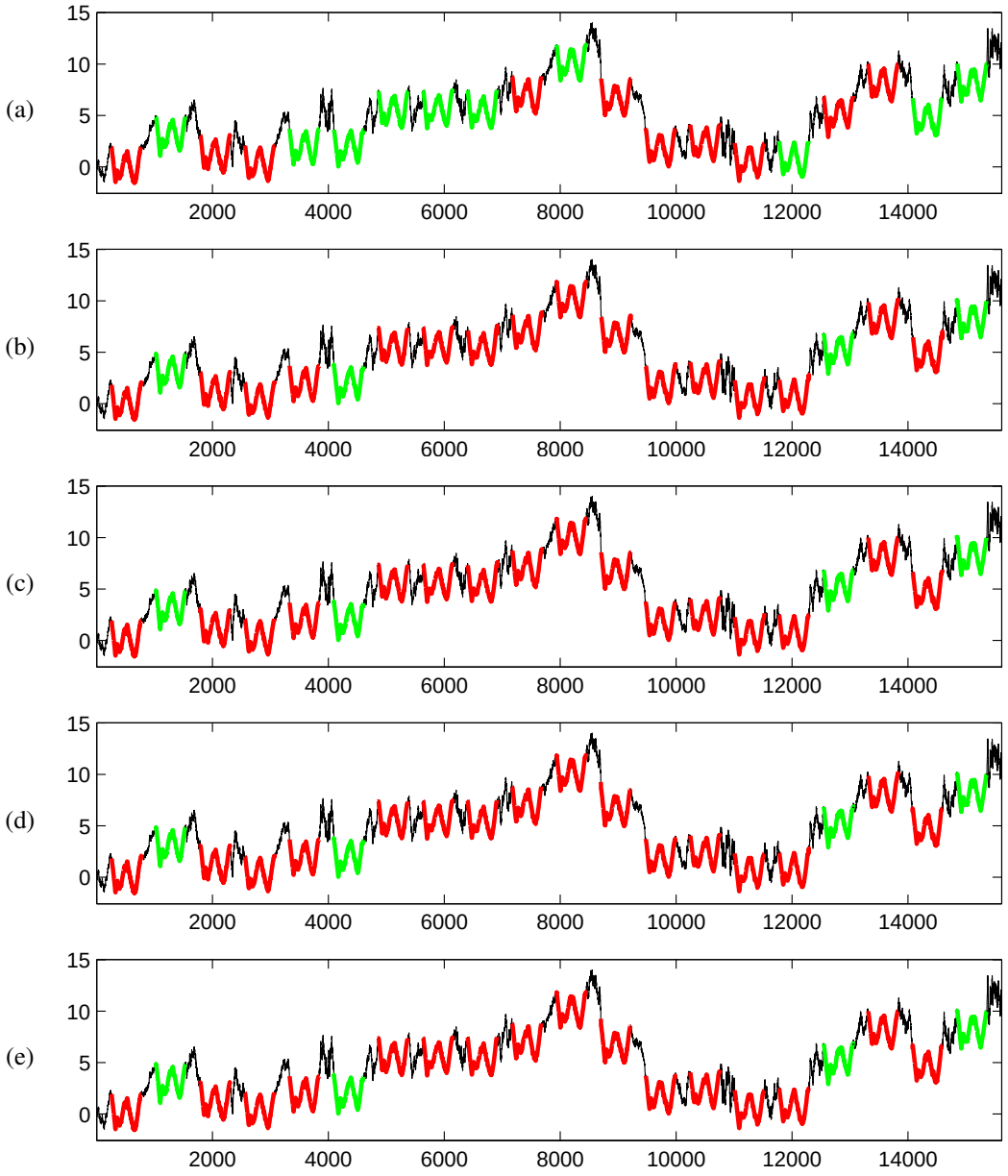

Figure B.39: Herring dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

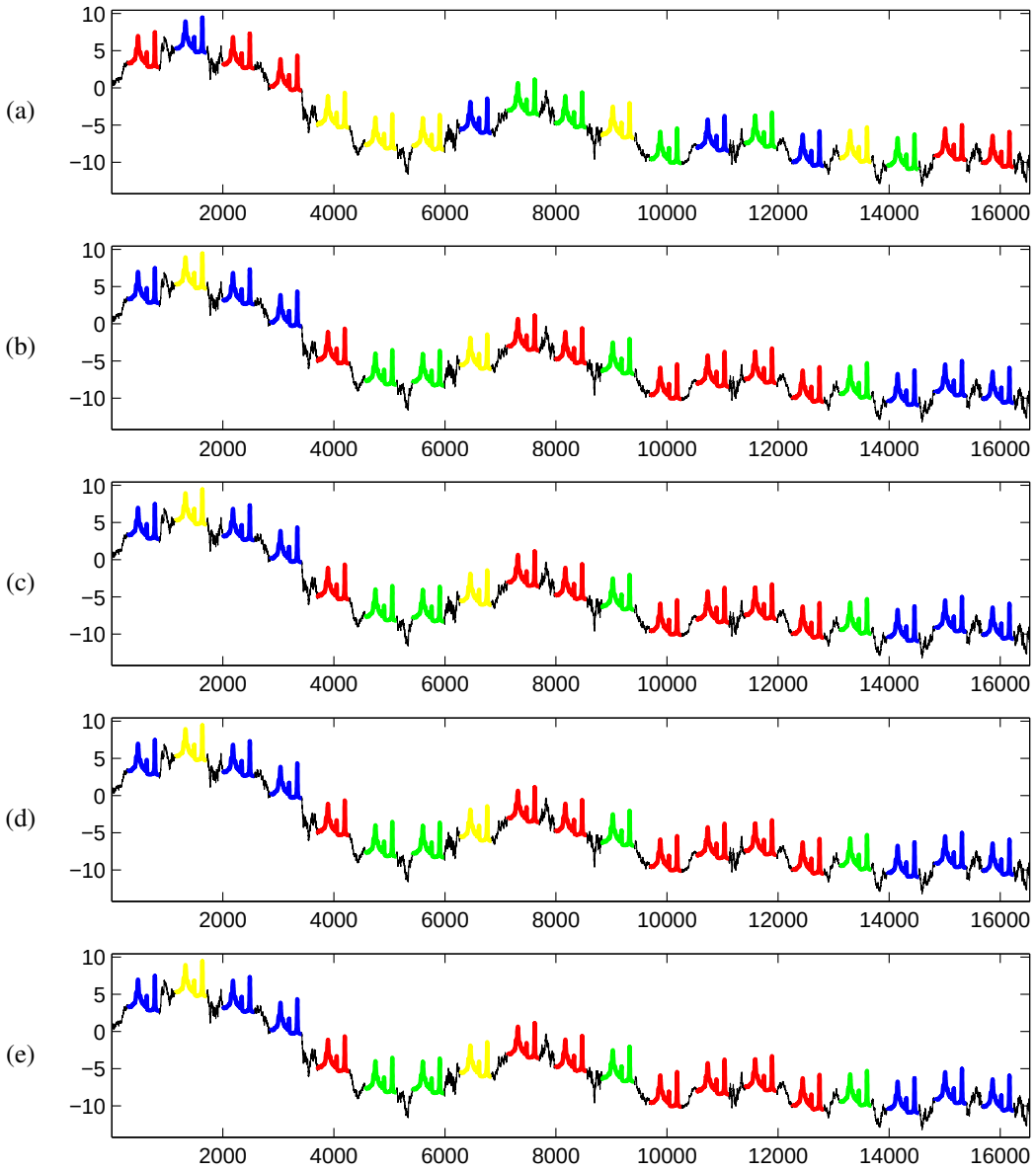

Figure B.40: OliveOil dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

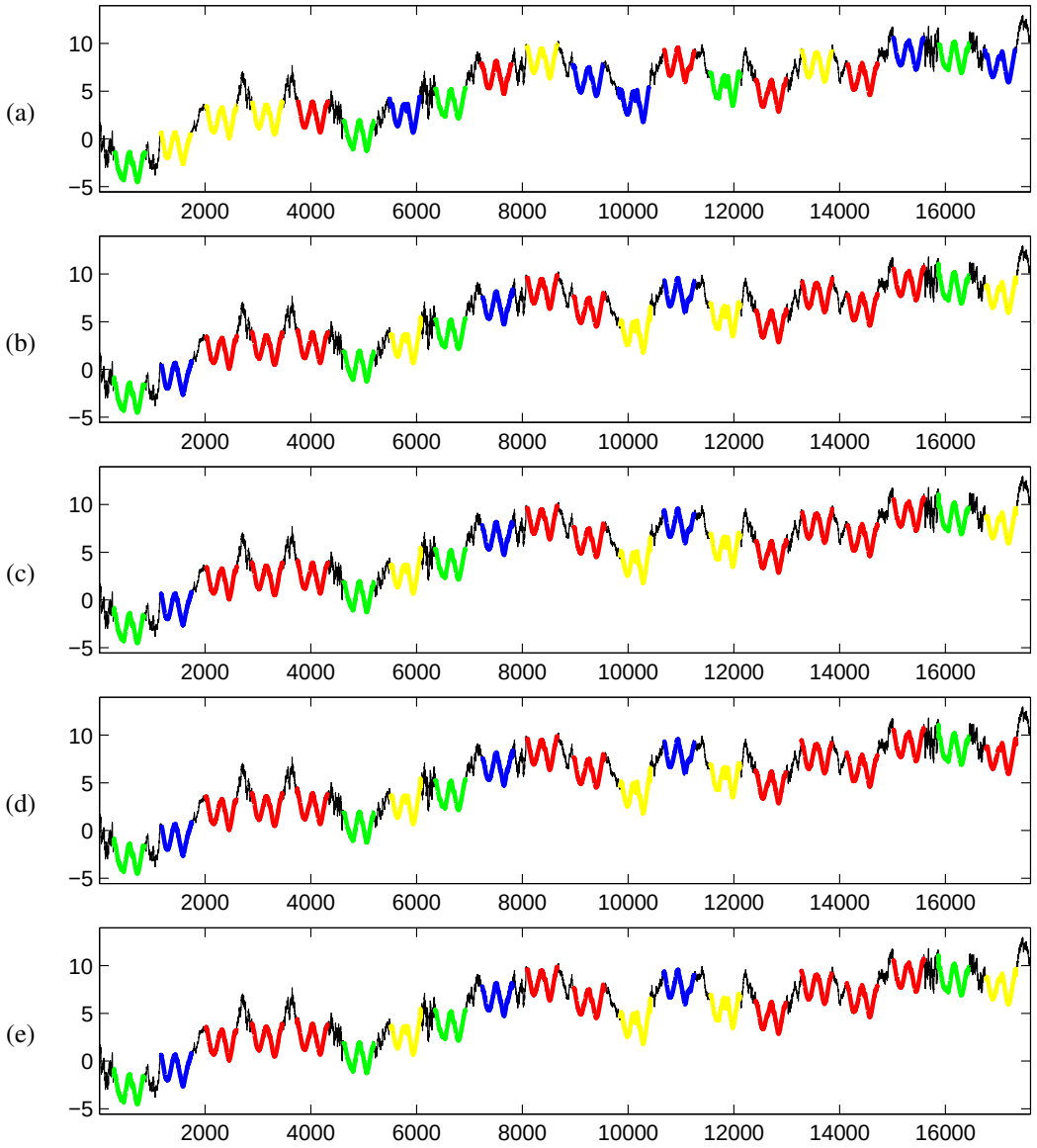

Figure B.41: Car dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

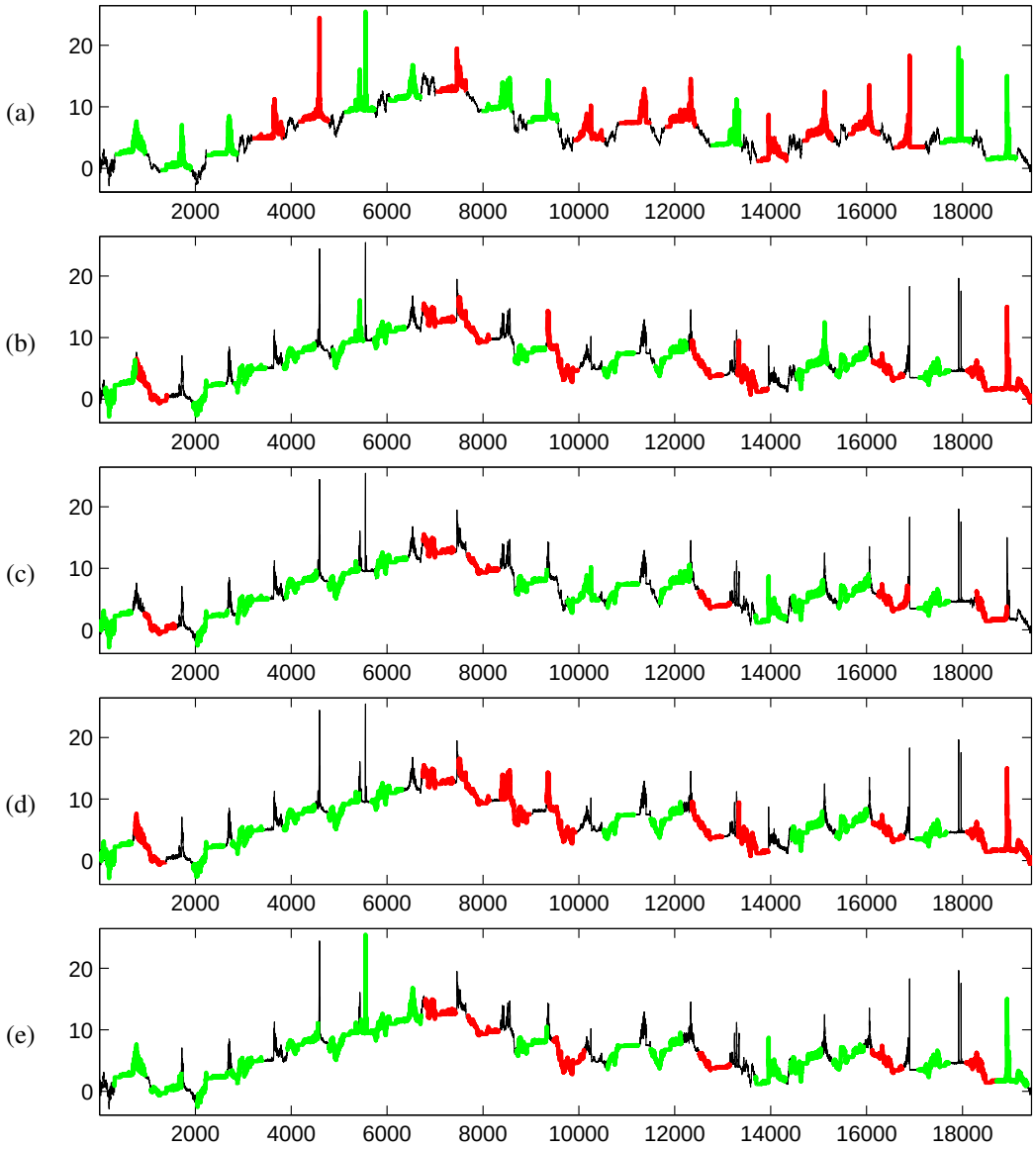

Figure B.42: Lighting2 dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

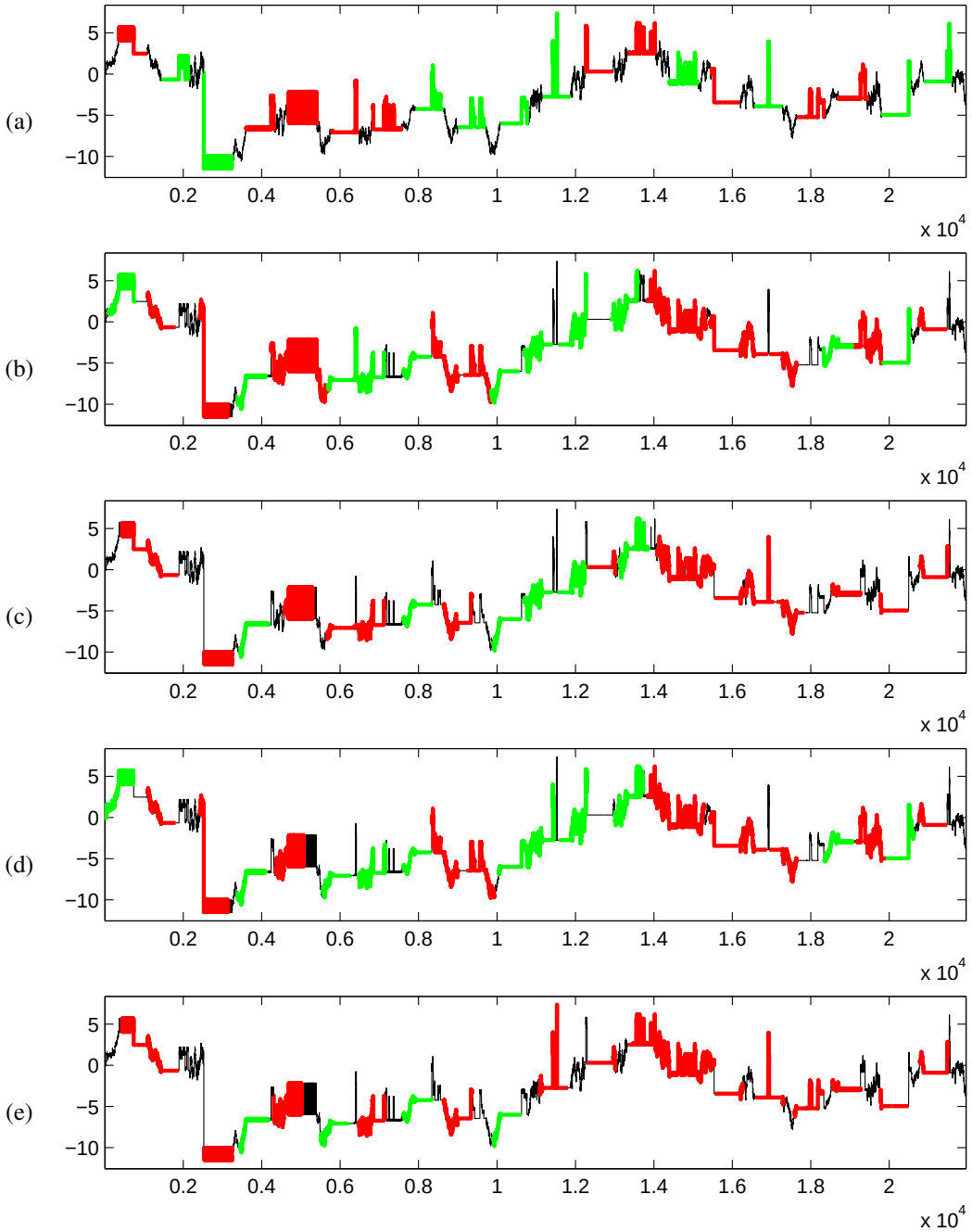

Figure B.43: Computers dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

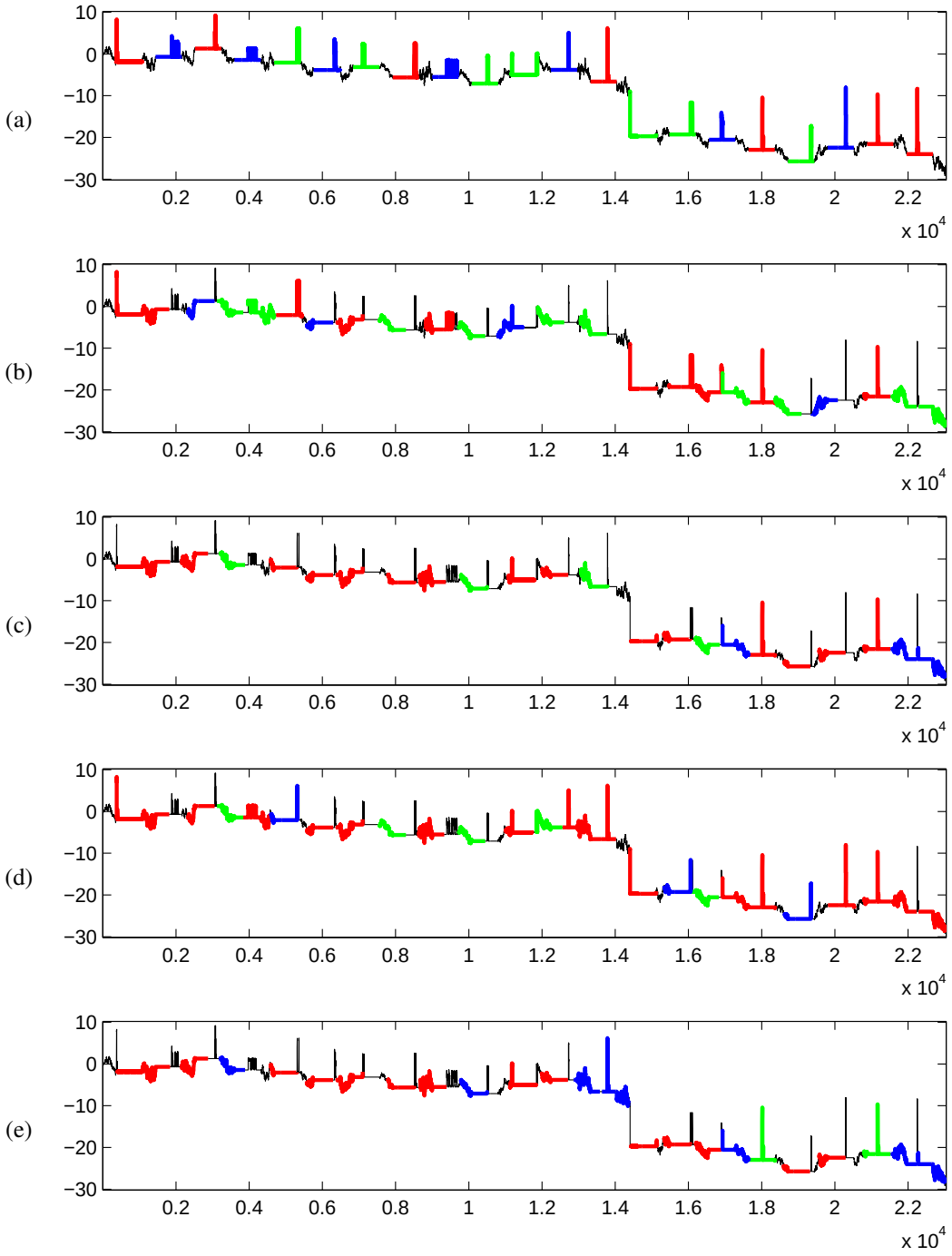

Figure B.44: LargeKitchenAppliances dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

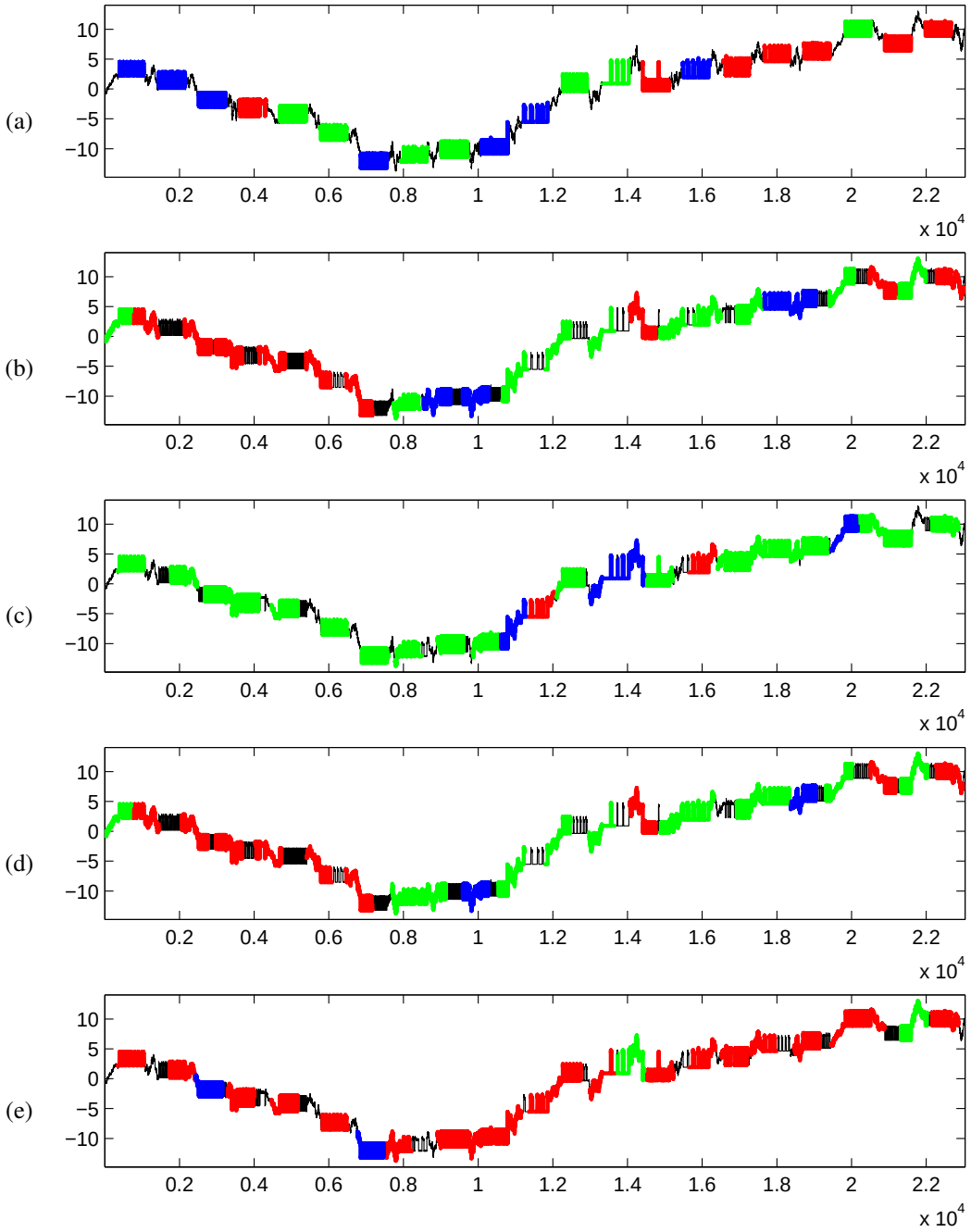

Figure B.45: RefrigerationDevices dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

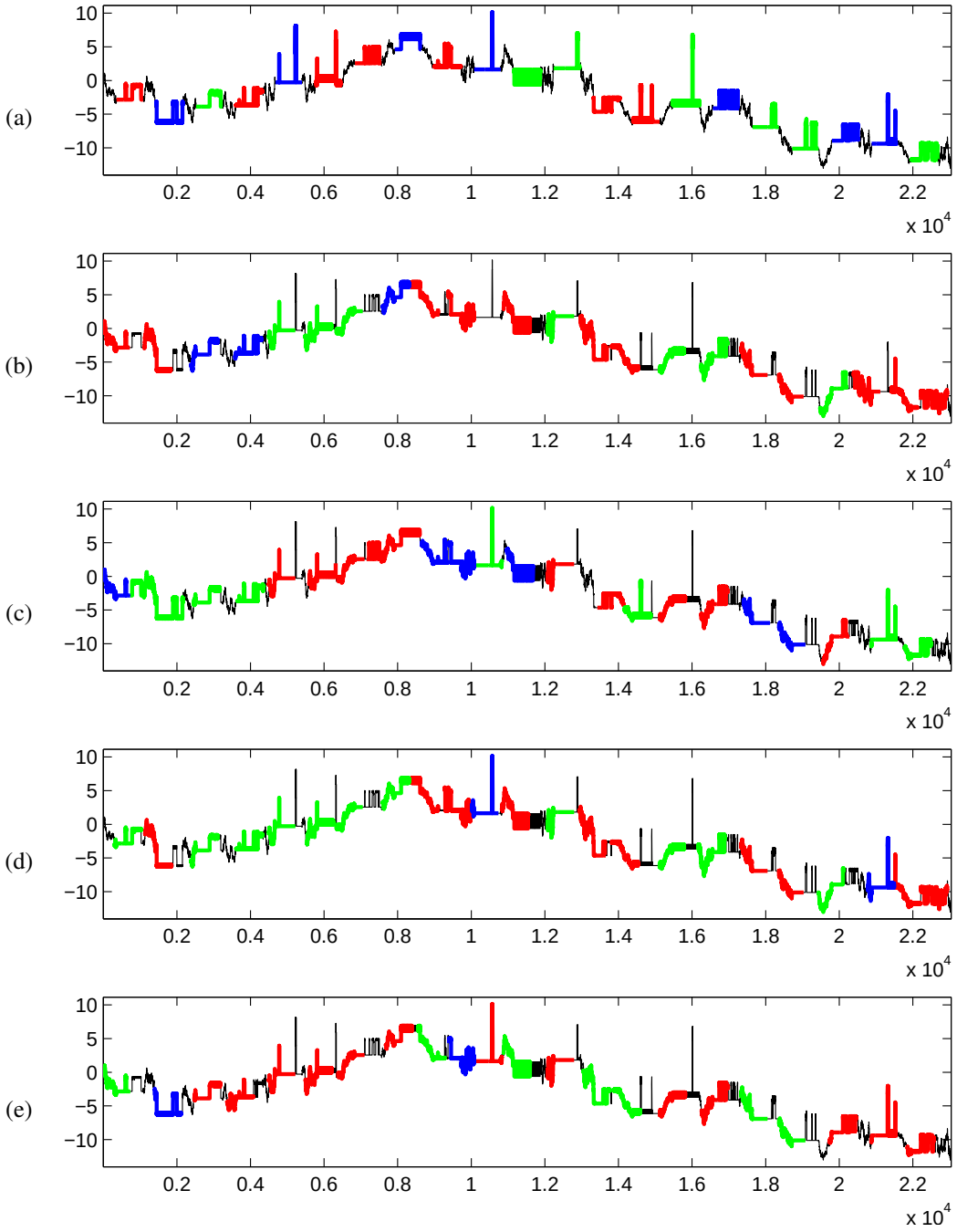

Figure B.46: ScreenType dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

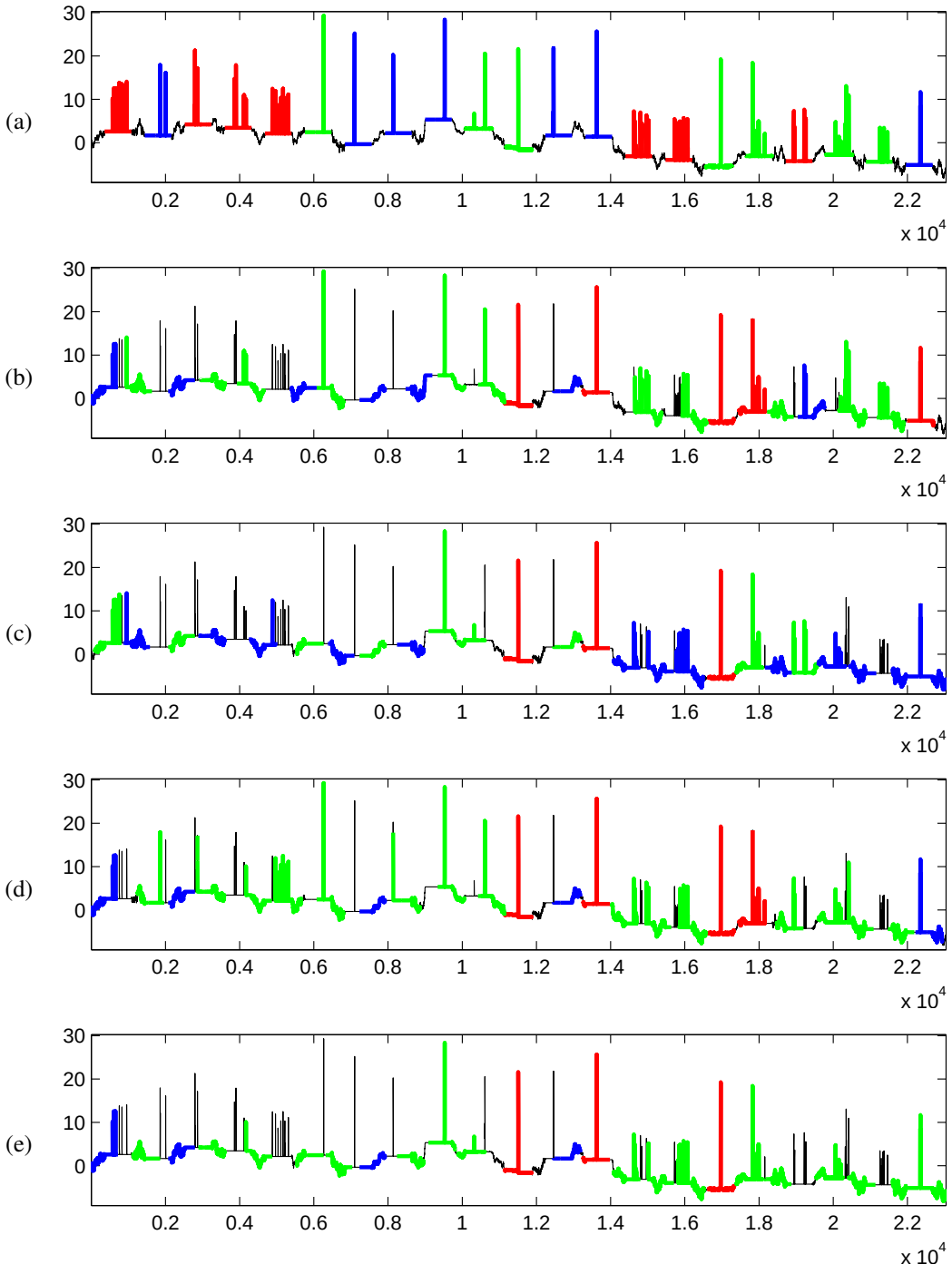

Figure B.47: SmallKitchenAppliances dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

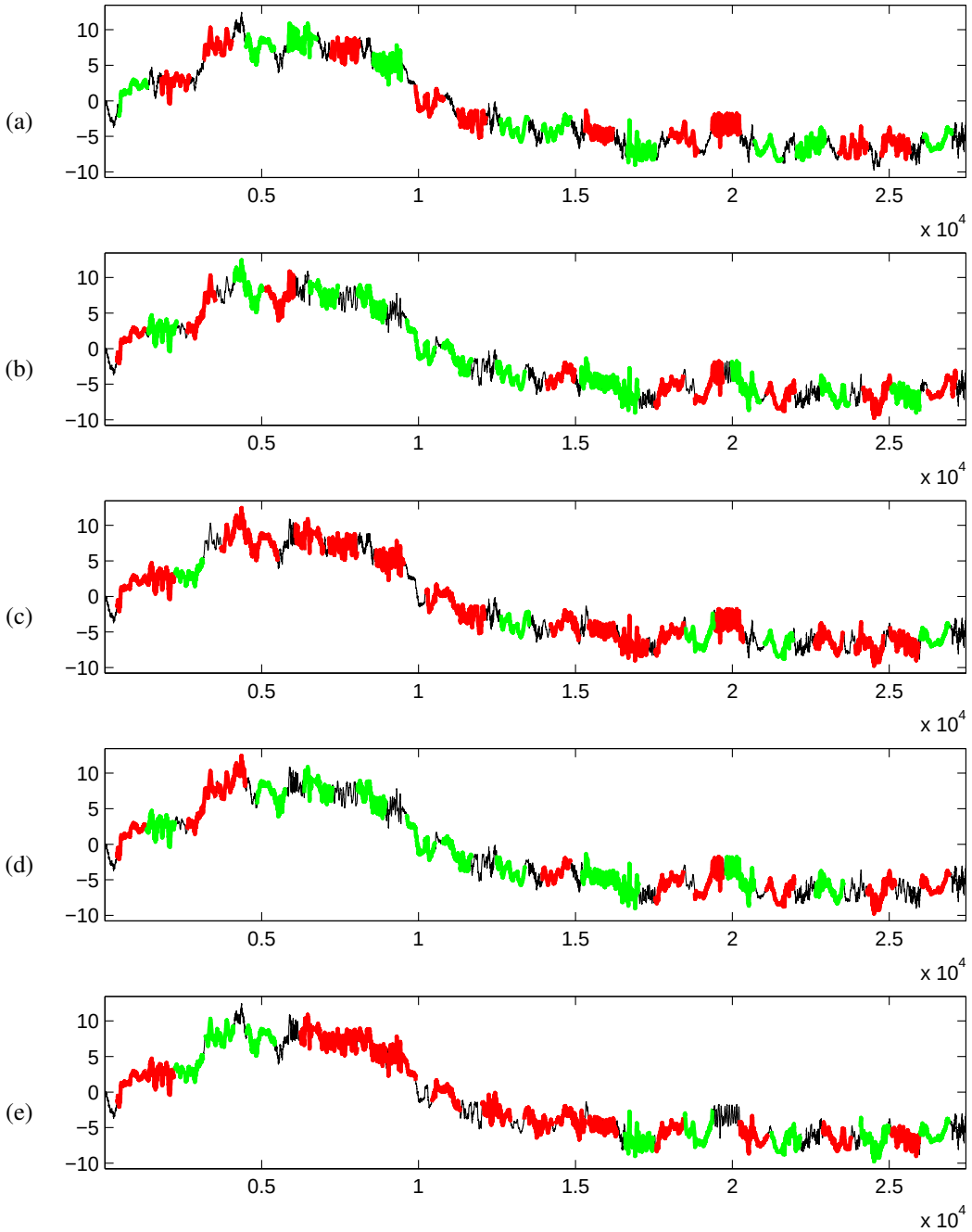

Figure B.48: WormsTwoClass dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

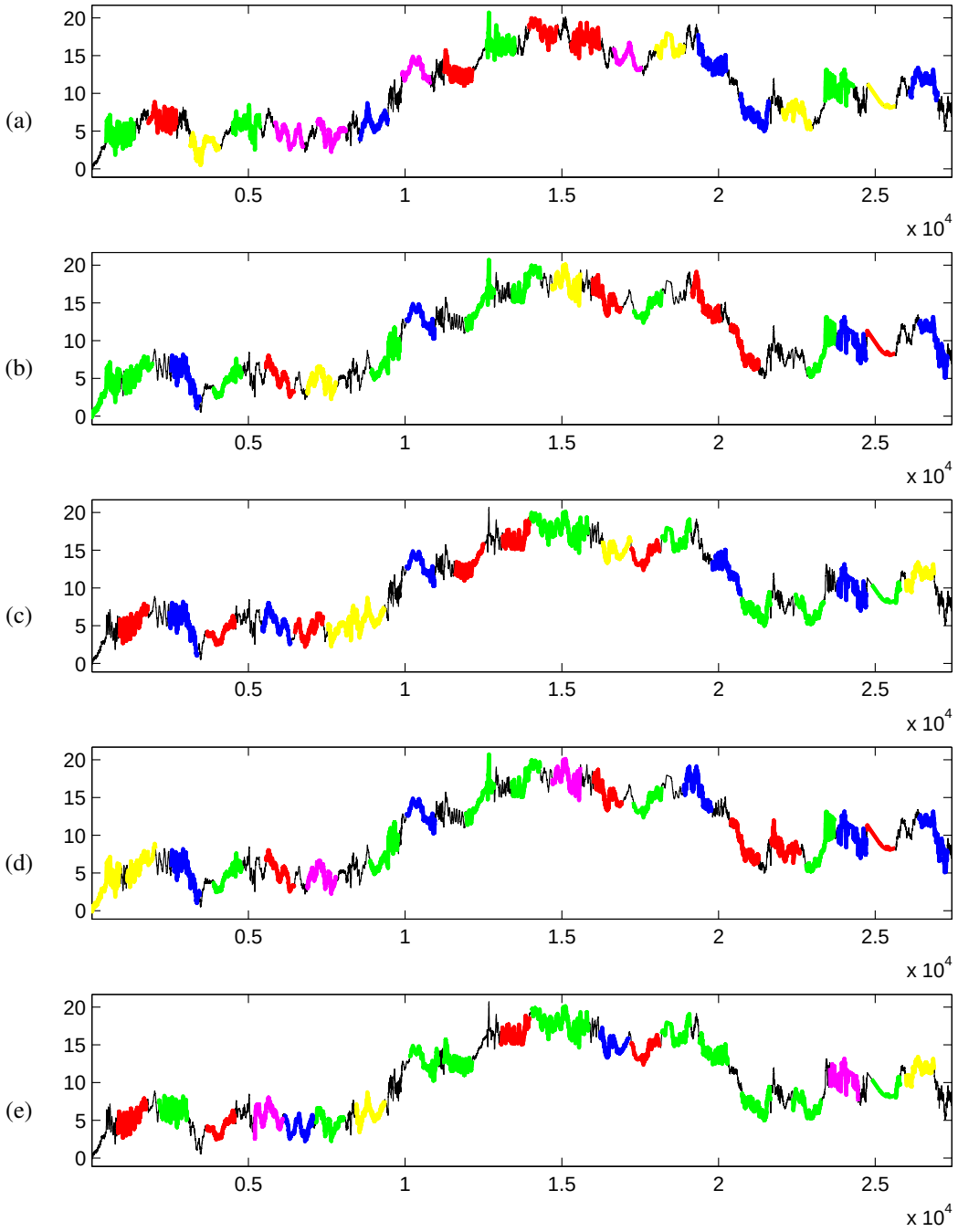

Figure B.49: Worms dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

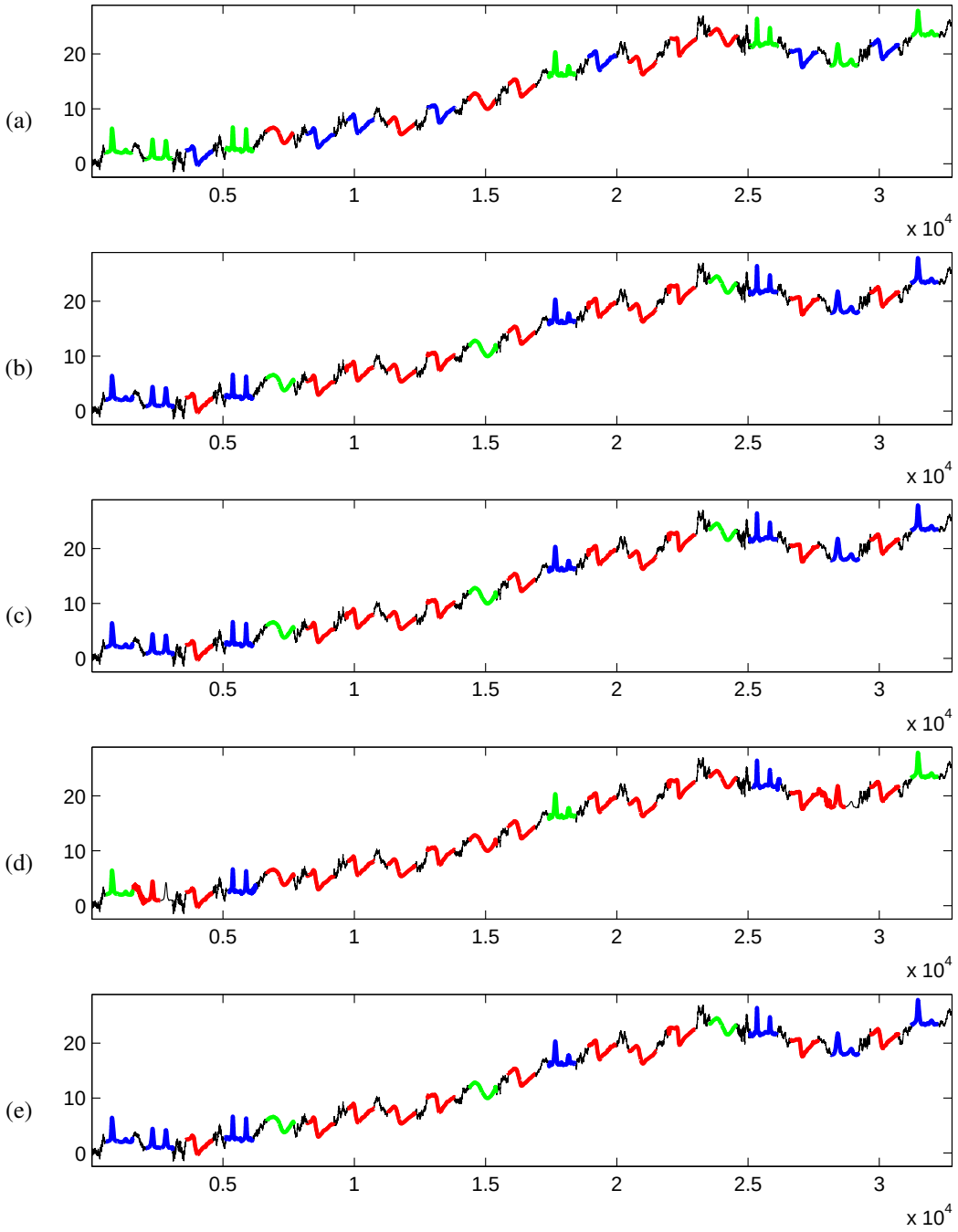

Figure B.50: StarLightCurves dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

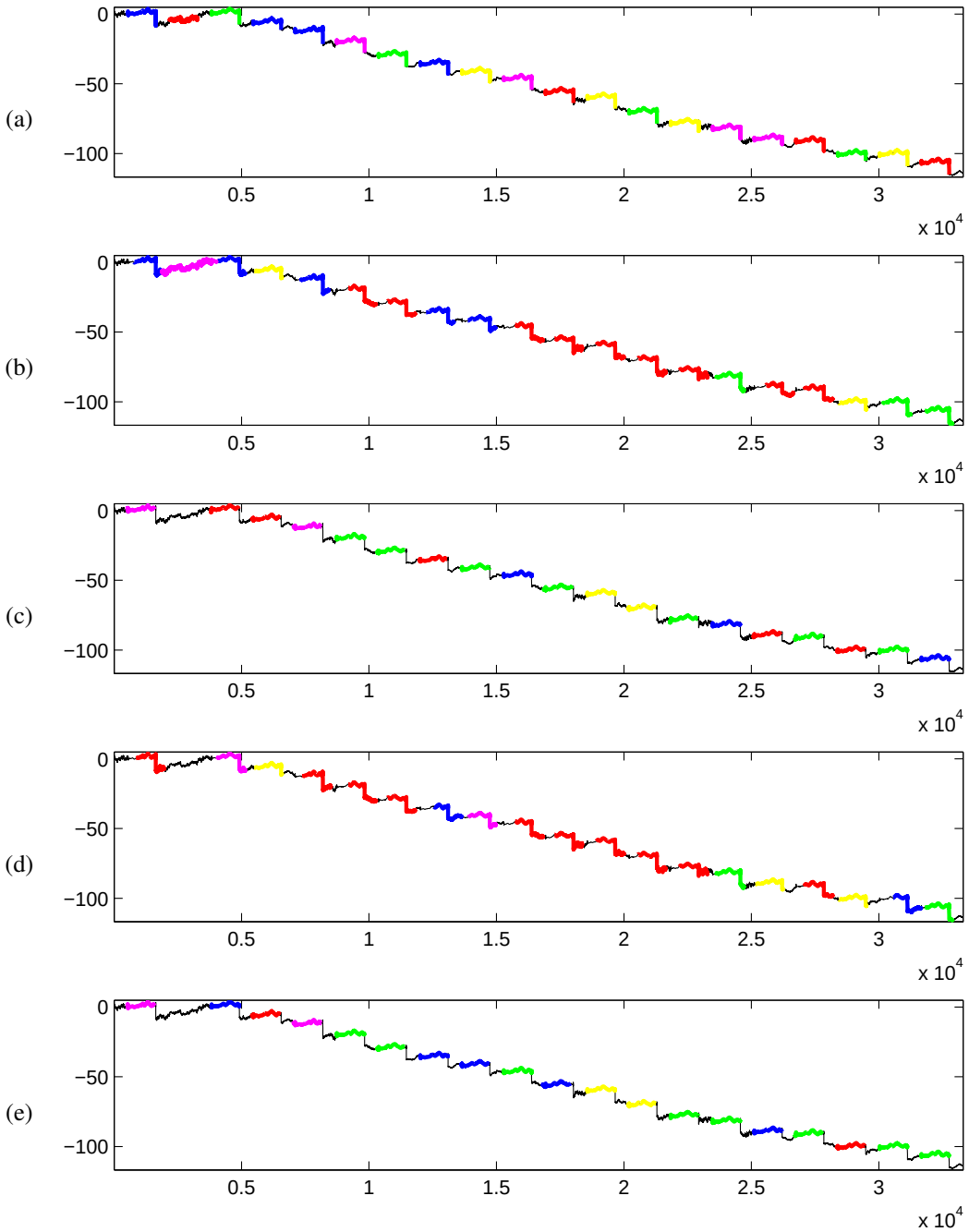

Figure B.51: Haptics dataset: (a) Input time series labeled with classes of planted data. (b), (c), (d) and (e) are output from SSTSC with E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

APPENDICES C

COMPLETE EXPERIMENTAL RESULTS OF THE EXPERIMENT IN SECTION 3.5.4 WHEN SCALING FACTOR IS SET TO 1.2

Table C.1: Rand Index (RI) from all algorithms on all datasets when the scaling factor (f) is 1.2 and number of clusters (k) is set to the number of classes in each dataset

Dataset	Rand Index							
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.47	0.45	**0.56**	0.51	0.48	0.49	0.47	0.47
SonyAIBORobotSurfaceII	0.61	**0.63**	0.48	0.59	0.54	0.53	0.47	0.49
SonyAIBORobotSurface	0.47	0.45	0.48	0.46	**0.56**	0.47	0.48	0.46
DistalPhalanxOutlineCorrect	**0.49**	**0.49**	**0.49**	**0.49**	**0.49**	**0.49**	0.47	0.47
MiddlePhalanxOutlineCorrect	0.56	0.53	**0.66**	**0.66**	0.61	0.56	0.49	0.49
PhalangesOutlinesCorrect	0.47	0.47	**0.49**	0.49	0.47	0.47	**0.49**	**0.49**
ProximalPhalanxOutlineCorrect	**0.49**	**0.49**	0.47	0.47	**0.49**	**0.49**	**0.49**	**0.49**
DistalPhalanxOutlineAgeGroup	0.54	0.45	0.54	0.54	0.59	0.44	**0.60**	**0.60**
MiddlePhalanxOutlineAgeGroup	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**	**0.59**	0.53	0.53
ProximalPhalanxOutlineAgeGroup	**0.73**	0.71	0.60	0.60	0.55	0.55	0.57	0.57
TwoLeadECG	0.52	0.53	0.48	0.49	0.52	0.53	**0.56**	**0.56**
MoteStrain	0.49	0.46	0.48	0.47	**0.54**	0.47	0.52	0.51
ECG200	0.48	0.47	**0.61**	0.56	0.49	0.50	0.49	0.48
CBF	**0.71**	0.69	0.60	0.56	0.51	0.50	0.60	0.56
Two_Patterns	0.73	0.69	0.66	0.69	0.63	**0.73**	0.53	0.64
ECGFiveDays	**0.51**	**0.51**	0.49	0.49	**0.51**	**0.51**	0.47	0.47
ECG5000	**0.72**	0.62	0.63	0.66	0.60	0.36	0.71	0.69
Gun_Point	0.52	**0.54**	0.52	**0.54**	0.52	0.53	0.52	0.52
wafer	0.48	0.47	0.66	**0.74**	0.47	0.47	0.48	0.47
ChlorineConcentration	0.47	0.50	0.49	0.47	0.47	**0.59**	0.45	0.45
Wine	0.47	0.47	0.49	0.49	0.47	0.47	**0.52**	**0.52**
Strawberry	0.48	0.48	0.48	0.48	0.48	0.48	**0.52**	**0.52**
ArrowHead	0.54	0.53	**0.65**	**0.65**	0.54	0.49	0.48	0.48
Trace	0.69	0.83	0.84	**0.84**	0.69	0.82	0.72	0.77
ToeSegmentation1	0.58	0.56	0.48	0.47	0.49	**0.62**	0.47	0.47
Coffee	**0.52**	**0.52**	0.49	0.49	0.47	0.47	**0.52**	**0.52**
ToeSegmentation2	0.51	0.54	0.49	**0.58**	0.53	0.49	0.48	0.47
FaceFour	0.50	0.33	**0.76**	0.67	0.52	0.40	0.71	0.71
yoga	0.48	0.47	0.47	0.47	0.48	0.47	0.48	**0.49**
Ham	**0.50**	0.40	0.47	0.47	0.47	0.44	0.47	0.49
Meat	0.54	0.54	0.52	0.52	0.52	0.52	**0.56**	**0.56**
Beef	0.74	0.65	**0.75**	0.67	0.74	0.65	0.67	0.57
FordA	**0.53**	0.40	0.48	0.47	0.48	0.43	0.47	0.52
FordB	0.45	0.00	0.48	0.46	0.47	**0.52**	0.49	0.47
ShapeletSim	0.44	0.33	**0.57**	0.33	0.49	0.46	0.44	0.00
BeetleFly	0.51	0.60	0.53	0.49	0.68	**1.00**	0.49	0.47
BirdChicken	**0.61**	0.43	0.48	0.45	0.52	0.44	0.56	0.58
Earthquakes	0.47	**0.57**	0.49	0.52	0.43	0.40	0.50	0.33
Herring	0.49	0.49	**0.52**	**0.52**	0.49	0.49	0.49	0.49
OliveOil	**0.69**	**0.69**	0.62	0.62	0.61	0.61	0.63	0.63
Car	**0.74**	0.74	**0.74**	**0.74**	**0.74**	0.74	0.66	0.66
Lighting2	0.47	0.40	0.48	0.50	0.47	0.50	0.52	**0.57**
Computers	0.49	0.53	0.51	0.44	0.49	**0.56**	0.48	0.47
LargeKitchenAppliances	0.63	0.67	0.64	**0.70**	0.58	0.59	0.43	0.47
RefrigerationDevices	**0.57**	0.40	0.48	0.51	0.52	0.43	0.40	0.36
ScreenType	0.50	0.17	0.33	0.28	0.48	0.47	0.47	**0.60**
SmallKitchenAppliances	**0.54**	0.53	0.47	0.46	**0.54**	0.39	0.37	0.29
WormsTwoClass	0.46	0.40	0.47	0.43	0.54	**1.00**	0.49	0.46
Worms	0.81	**0.83**	0.73	**0.83**	0.68	0.81	0.69	0.60
StarLightCurves	0.70	**0.71**	0.70	**0.71**	0.70	**0.71**	0.70	**0.71**
Haptics	0.68	0.64	0.71	0.71	0.66	0.72	**0.73**	**0.73**

Table C.2: Precision from all algorithms on all datasets when the scaling factor (f) is 1.2 and number of clusters (k) is set to the number of classes in each dataset

Dataset	Precision							
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.47	0.45	**0.53**	0.48	0.47	0.46	0.45	0.45
SonyAIBORobotSurfaceII	0.57	**0.59**	0.46	0.54	0.51	0.51	0.47	0.49
SonyAIBORobotSurface	0.47	0.54	0.48	0.56	0.63	**1.00**	0.48	0.50
DistalPhalanxOutlineCorrect	**0.48**	**0.48**	**0.48**	**0.48**	**0.48**	**0.48**	0.46	0.46
MiddlePhalanxOutlineCorrect	0.53	0.51	**0.64**	**0.64**	0.57	0.53	0.48	0.48
PhalangesOutlinesCorrect	0.46	0.46	**0.48**	0.47	0.47	0.47	0.47	0.47
ProximalPhalanxOutlineCorrect	**0.47**	**0.47**	0.44	0.44	**0.47**	**0.47**	**0.47**	**0.47**
DistalPhalanxOutlineAgeGroup	0.32	0.33	0.33	0.33	0.35	0.32	**0.38**	**0.38**
MiddlePhalanxOutlineAgeGroup	**0.34**	**0.34**	**0.34**	**0.34**	**0.34**	**0.34**	0.26	0.26
ProximalPhalanxOutlineAgeGroup	**0.54**	0.51	0.39	0.39	0.33	0.32	0.30	0.30
TwoLeadECG	0.50	0.50	0.45	0.46	0.50	0.50	**0.52**	**0.52**
MoteStrain	0.47	0.44	0.46	0.45	**0.52**	0.44	0.50	0.50
ECG200	0.45	0.45	**0.58**	0.53	0.47	0.47	0.47	0.45
CBF	**0.52**	0.46	0.39	0.34	0.31	0.25	0.39	0.37
Two_Patterns	0.35	0.30	0.23	0.17	0.30	**0.50**	0.24	0.33
ECGFiveDays	**0.48**	**0.48**	0.46	0.46	**0.48**	**0.48**	0.45	0.45
ECG5000	**0.28**	0.26	0.18	0.26	0.19	0.22	0.26	0.27
Gun_Point	0.50	**0.51**	0.50	**0.51**	0.50	0.50	0.50	0.50
wafer	0.47	0.47	0.62	**0.70**	0.47	0.47	0.47	0.46
ChlorineConcentration	0.29	0.18	0.28	0.28	0.31	**0.32**	0.31	0.31
Wine	0.44	0.44	0.47	0.47	0.47	0.47	**0.50**	**0.50**
Strawberry	0.45	0.45	0.45	0.45	0.45	0.45	**0.49**	**0.49**
ArrowHead	0.28	0.28	**0.42**	**0.42**	0.29	0.29	0.31	0.31
Trace	0.38	0.56	0.60	**0.61**	0.38	0.55	0.40	0.47
ToeSegmentation1	0.54	0.56	0.45	0.43	0.48	**0.59**	0.45	0.43
Coffee	0.49	0.49	0.47	0.47	0.47	0.47	**0.50**	**0.50**
ToeSegmentation2	0.49	0.50	0.47	**0.56**	0.51	0.47	0.47	0.47
FaceFour	0.21	**1.00**	0.47	0.47	0.21	0.29	0.38	0.44
yoga	0.45	0.45	0.45	0.44	0.45	0.45	0.46	**0.47**
Ham	0.47	0.40	0.44	0.42	0.46	0.46	0.47	**0.49**
Meat	0.33	0.33	0.28	0.28	**0.33**	**0.33**	0.31	0.31
Beef	**0.24**	0.17	0.24	0.16	**0.24**	0.17	0.16	0.15
FordA	0.50	0.40	0.47	0.44	0.48	0.43	0.47	**0.52**
FordB	0.44	0.00	0.47	0.46	0.48	**0.67**	0.48	0.47
ShapeletSim	0.39	0.33	**0.52**	0.33	0.45	0.43	0.44	0.00
BeetleFly	0.51	0.60	0.50	0.49	0.68	**1.00**	0.48	0.47
BirdChicken	**0.58**	0.50	0.45	0.47	0.49	0.44	0.52	0.54
Earthquakes	0.46	0.50	0.47	**0.52**	0.33	0.33	0.46	0.33
Herring	0.48	0.48	**0.49**	**0.49**	0.48	0.48	0.48	0.48
OliveOil	**0.27**	**0.27**	0.20	0.20	0.16	0.16	0.22	0.22
Car	0.42	**0.42**	0.40	0.40	0.42	**0.42**	0.27	0.27
Lighting2	0.45	0.50	0.45	0.50	0.45	0.44	0.50	**0.52**
Computers	0.47	0.50	0.49	0.44	0.47	**0.53**	0.47	0.45
LargeKitchenAppliances	0.39	0.45	0.44	**0.50**	0.33	0.36	0.32	0.34
RefrigerationDevices	0.31	0.33	0.28	0.25	0.26	0.18	0.29	**0.36**
ScreenType	0.28	**1.00**	0.29	0.28	0.30	0.40	0.30	0.50
SmallKitchenAppliances	0.28	0.23	0.31	0.29	**0.32**	0.29	0.30	0.29
WormsTwoClass	0.50	0.50	0.43	0.47	0.57	**1.00**	0.46	0.45
Worms	**0.43**	0.00	0.14	0.00	0.23	0.33	0.13	0.00
StarLightCurves	0.50	**0.52**	0.50	0.51	0.50	0.51	0.50	0.51
Haptics	**0.22**	0.00	0.15	0.15	0.11	0.13	0.14	0.14

Table C.3: Recall from all algorithms on all datasets when the scaling factor (f) is 1.2 and number of clusters (k) is set to the number of classes in each dataset

Dataset	Recall							
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.47	0.45	0.53	0.48	**0.72**	0.61	0.53	0.59
SonyAIBORobotSurfaceII	0.72	0.72	0.59	**0.81**	0.70	0.67	0.47	0.49
SonyAIBORobotSurface	0.57	0.54	0.65	0.56	0.53	0.47	0.65	**0.67**
DistalPhalanxOutlineCorrect	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	**0.82**	0.64	0.64
MiddlePhalanxOutlineCorrect	0.62	0.67	0.67	0.67	0.72	**0.85**	0.67	0.67
PhalangesOutlinesCorrect	0.64	0.64	0.67	0.72	**0.80**	**0.80**	0.47	0.47
ProximalPhalanxOutlineCorrect	**0.49**	**0.49**	0.44	0.44	**0.49**	**0.49**	**0.49**	**0.49**
DistalPhalanxOutlineAgeGroup	0.46	**0.69**	0.52	0.52	0.46	0.65	0.51	0.51
MiddlePhalanxOutlineAgeGroup	**0.41**	**0.41**	**0.41**	**0.41**	**0.41**	**0.41**	0.32	0.32
ProximalPhalanxOutlineAgeGroup	**0.71**	0.70	0.62	0.62	0.48	0.44	0.33	0.33
TwoLeadECG	0.77	**0.84**	0.50	0.49	0.77	**0.84**	0.73	0.73
MoteStrain	0.49	0.58	0.59	0.63	0.56	0.57	0.63	**0.71**
ECG200	0.50	**0.63**	0.59	0.53	0.47	0.50	0.49	0.47
CBF	0.57	0.60	0.62	0.59	0.45	0.40	0.59	**0.65**
Two_Patterns	0.38	0.30	0.28	0.14	0.56	0.50	0.57	**0.71**
ECGFiveDays	**0.56**	**0.56**	0.53	0.53	**0.56**	**0.56**	0.48	0.48
ECG5000	0.45	0.60	0.36	0.38	0.44	**1.00**	0.44	0.50
Gun_Point	0.77	0.78	0.77	0.78	0.77	**0.84**	0.77	0.77
wafer	0.78	0.47	0.73	**0.92**	0.88	0.47	0.72	0.67
ChlorineConcentration	0.51	0.21	0.44	0.50	0.60	0.44	**0.68**	**0.68**
Wine	0.44	0.44	0.56	0.56	0.47	0.47	**0.63**	**0.63**
Strawberry	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**
ArrowHead	0.33	0.37	0.44	0.44	0.37	0.48	**0.62**	**0.62**
Trace	0.70	0.83	0.65	0.69	0.70	**0.86**	0.65	0.72
ToeSegmentation1	0.79	0.56	0.50	0.48	0.82	**0.93**	0.47	0.45
Coffee	0.50	0.50	0.49	0.49	0.47	0.47	**0.77**	**0.77**
ToeSegmentation2	0.72	0.78	0.56	0.67	0.67	0.84	**0.90**	0.47
FaceFour	0.45	0.33	0.64	0.70	0.45	0.33	0.52	**0.75**
yoga	0.50	0.47	0.47	0.50	0.50	0.47	0.59	**0.64**
Ham	0.64	0.40	0.44	0.43	0.63	**0.72**	0.47	0.49
Meat	0.52	0.52	0.40	0.40	**0.59**	**0.59**	0.37	0.37
Beef	**0.30**	0.25	0.27	0.21	**0.30**	0.25	0.27	0.29
FordA	0.57	**1.00**	0.89	0.80	0.48	0.43	0.88	0.52
FordB	0.55	0.00	**0.78**	0.46	0.48	0.67	0.77	0.47
ShapeletSim	0.44	**1.00**	0.85	**1.00**	0.47	0.75	0.44	0.00
BeetleFly	0.48	**1.00**	0.84	0.49	0.65	**1.00**	0.82	0.47
BirdChicken	0.59	0.38	0.50	0.47	0.50	0.39	0.73	**0.79**
Earthquakes	0.68	0.67	0.86	**1.00**	0.33	0.50	0.81	0.33
Herring	**0.82**	**0.82**	0.50	0.50	**0.82**	**0.82**	0.67	0.67
OliveOil	0.28	0.28	0.28	0.28	0.19	0.19	**0.31**	**0.31**
Car	0.55	**0.69**	0.48	0.48	0.55	**0.69**	0.38	0.38
Lighting2	0.46	0.33	0.46	0.33	0.46	0.44	0.63	**0.85**
Computers	0.72	0.86	0.83	0.44	0.72	**0.89**	0.78	0.83
LargeKitchenAppliances	0.46	0.42	0.72	0.78	0.37	0.31	0.78	**0.85**
RefrigerationDevices	0.38	0.50	0.46	0.42	0.32	0.40	**0.69**	0.36
ScreenType	0.32	0.17	**0.84**	0.28	0.41	0.67	0.65	0.50
SmallKitchenAppliances	0.33	0.30	0.67	0.64	0.46	0.80	**0.81**	0.29
WormsTwoClass	0.50	0.33	0.45	0.64	0.57	**1.00**	0.53	0.50
Worms	0.45	0.00	0.17	0.00	0.40	**1.00**	0.17	0.00
StarLightCurves	0.56	0.57	0.65	**0.69**	0.65	**0.69**	0.65	**0.69**
Haptics	**0.40**	0.00	0.19	0.19	0.17	0.25	0.15	0.15

Table C.4: F1-score from all algorithms on all datasets when the scaling factor (f) is 1.2 and number of clusters (k) is set to the number of classes in each dataset

Dataset	F1-Measure							
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.47	0.45	0.53	0.48	**0.57**	0.52	0.49	0.51
SonyAIBORobotSurfaceII	0.63	**0.65**	0.52	0.65	0.59	0.58	0.47	0.49
SonyAIBORobotSurface	0.52	0.54	0.55	0.56	0.57	**0.64**	0.55	0.57
DistalPhalanxOutlineCorrect	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**	**0.61**	0.54	0.54
MiddlePhalanxOutlineCorrect	0.57	0.58	**0.65**	**0.65**	0.63	0.65	0.56	0.56
PhalangesOutlinesCorrect	0.54	0.54	0.56	0.57	**0.59**	**0.59**	0.47	0.47
ProximalPhalanxOutlineCorrect	**0.48**	**0.48**	0.44	0.44	**0.48**	**0.48**	**0.48**	**0.48**
DistalPhalanxOutlineAgeGroup	0.38	**0.45**	0.40	0.40	0.40	0.43	0.44	0.44
MiddlePhalanxOutlineAgeGroup	**0.37**	**0.37**	**0.37**	**0.37**	**0.37**	**0.37**	0.29	0.29
ProximalPhalanxOutlineAgeGroup	**0.61**	0.59	0.48	0.48	0.39	0.37	0.32	0.32
TwoLeadECG	0.60	**0.63**	0.48	0.48	0.60	**0.63**	0.61	0.61
MoteStrain	0.48	0.50	0.52	0.53	0.54	0.49	0.56	**0.59**
ECG200	0.48	0.53	**0.59**	0.53	0.47	0.48	0.48	0.46
CBF	**0.54**	0.52	0.48	0.43	0.37	0.31	0.47	0.47
Two_Patterns	0.36	0.30	0.25	0.15	0.39	**0.50**	0.33	0.45
ECGFiveDays	**0.52**	**0.52**	0.49	0.49	**0.52**	**0.52**	0.46	0.46
ECG5000	0.34	**0.36**	0.24	0.31	0.27	**0.36**	0.33	0.35
Gun_Point	0.60	0.62	0.60	0.62	0.60	**0.63**	0.60	0.60
wafer	0.59	0.47	0.67	**0.80**	0.61	0.47	0.57	0.54
ChlorineConcentration	0.37	0.20	0.34	0.36	0.41	0.37	**0.43**	**0.43**
Wine	0.44	0.44	0.51	0.51	0.47	0.47	**0.56**	**0.56**
Strawberry	0.48	0.48	0.48	0.48	0.48	0.48	**0.50**	**0.50**
ArrowHead	0.30	0.32	**0.43**	**0.43**	0.32	0.36	0.42	0.42
Trace	0.49	**0.67**	0.63	0.65	0.49	**0.67**	0.50	0.57
ToeSegmentation1	0.64	0.56	0.48	0.45	0.61	**0.72**	0.46	0.44
Coffee	0.50	0.50	0.48	0.48	0.47	0.47	**0.60**	**0.60**
ToeSegmentation2	0.58	0.61	0.51	0.61	0.58	0.60	**0.62**	0.47
FaceFour	0.28	0.50	0.54	**0.56**	0.29	0.31	0.44	0.56
yoga	0.48	0.46	0.46	0.47	0.48	0.46	0.52	**0.54**
Ham	0.54	0.40	0.44	0.43	0.53	**0.57**	0.47	0.49
Meat	0.40	0.40	0.33	0.33	**0.42**	**0.42**	0.33	0.33
Beef	**0.26**	0.20	0.25	0.18	**0.26**	0.20	0.20	0.19
FordA	0.53	0.57	**0.62**	0.57	0.48	0.43	0.61	0.52
FordB	0.49	0.00	0.59	0.46	0.48	**0.67**	0.59	0.47
ShapeletSim	0.41	0.50	**0.65**	0.50	0.46	0.55	0.44	0.00
BeetleFly	0.49	0.75	0.63	0.49	0.67	**1.00**	0.61	0.47
BirdChicken	0.59	0.43	0.48	0.47	0.50	0.41	0.61	**0.64**
Earthquakes	0.55	0.57	0.61	**0.69**	0.33	0.40	0.59	0.33
Herring	**0.61**	**0.61**	0.50	0.50	**0.61**	**0.61**	0.56	0.56
OliveOil	**0.27**	**0.27**	0.24	0.24	0.17	0.17	0.26	0.26
Car	0.47	**0.52**	0.44	0.44	0.47	**0.52**	0.32	0.32
Lighting2	0.45	0.40	0.45	0.40	0.45	0.44	0.56	**0.65**
Computers	0.57	0.63	0.61	0.44	0.57	**0.67**	0.59	0.59
LargeKitchenAppliances	0.42	0.43	0.55	**0.61**	0.35	0.33	0.45	0.49
RefrigerationDevices	0.34	0.40	0.35	0.31	0.29	0.25	**0.41**	0.36
ScreenType	0.30	0.29	0.43	0.28	0.35	**0.50**	0.41	**0.50**
SmallKitchenAppliances	0.30	0.26	0.43	0.40	0.38	0.42	**0.43**	0.29
WormsTwoClass	0.50	0.40	0.44	0.54	0.57	**1.00**	0.49	0.48
Worms	0.44	0.00	0.15	0.00	0.29	**0.50**	0.14	0.00
StarLightCurves	0.53	0.54	0.57	**0.59**	0.57	**0.59**	0.57	**0.59**
Haptics	**0.28**	0.00	0.17	0.17	0.13	0.17	0.15	0.15

Table C.5: AoR from all algorithms on all datasets when the scaling factor (f) is 1.2 and number of clusters (k) is set to the number of classes in each dataset

Dataset	Accuracy on Retrieval (AoR)							
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	**1.00**	0.55	**1.00**	0.85	**1.00**	0.65	**1.00**	0.85
SonyAIBORobotSurfaceII	**1.00**	0.90	**1.00**	0.60	0.95	0.90	**1.00**	0.85
SonyAIBORobotSurface	**1.00**	0.75	**1.00**	0.81	0.88	0.63	**1.00**	0.81
DistalPhalanxOutlineCorrect	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
MiddlePhalanxOutlineCorrect	**1.00**	0.90	**1.00**	**1.00**	**1.00**	0.85	**1.00**	**1.00**
PhalangesOutlinesCorrect	**1.00**	**1.00**	**1.00**	0.95	**1.00**	**1.00**	**1.00**	**1.00**
ProximalPhalanxOutlineCorrect	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
DistalPhalanxOutlineAgeGroup	**1.00**	0.76	**1.00**	**1.00**	**1.00**	0.71	**1.00**	**1.00**
MiddlePhalanxOutlineAgeGroup	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
ProximalPhalanxOutlineAgeGroup	**1.00**	0.95	**1.00**	**1.00**	**1.00**	0.95	**1.00**	**1.00**
TwoLeadECG	**1.00**	0.90	**1.00**	0.95	**1.00**	0.90	**1.00**	**1.00**
MoteStrain	**1.00**	0.65	**1.00**	0.75	0.95	0.60	**1.00**	0.85
ECG200	**1.00**	0.75	**1.00**	0.85	**1.00**	0.80	**1.00**	0.90
CBF	0.90	0.43	**1.00**	0.62	0.81	0.43	**1.00**	0.86
Two_Patterns	0.90	0.50	**1.00**	0.45	0.90	0.30	0.85	0.60
ECGFiveDays	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
ECG5000	0.94	0.61	**1.00**	0.83	**1.00**	0.61	**1.00**	0.83
Gun_Point	**1.00**	0.95	**1.00**	0.95	**1.00**	0.90	**1.00**	**1.00**
wafer	0.90	0.60	**1.00**	0.70	0.90	0.60	**1.00**	0.85
ChlorineConcentration	**1.00**	0.57	**1.00**	0.81	**1.00**	0.57	**1.00**	**1.00**
Wine	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Strawberry	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
ArrowHead	**1.00**	0.95	**1.00**	**1.00**	**1.00**	0.86	**1.00**	**1.00**
Trace	**1.00**	0.80	**1.00**	0.90	**1.00**	0.85	**1.00**	0.95
ToeSegmentation1	0.90	0.55	**1.00**	0.70	**1.00**	0.65	**1.00**	0.75
Coffee	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
ToeSegmentation2	0.85	0.65	**1.00**	0.75	0.90	0.55	**1.00**	0.60
FaceFour	**1.00**	0.17	**1.00**	0.67	0.94	0.33	**1.00**	0.67
yoga	**1.00**	0.90	**1.00**	0.90	**1.00**	0.90	**1.00**	0.95
Ham	0.90	0.30	**0.95**	0.60	**0.95**	0.45	0.80	0.55
Meat	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Beef	**1.00**	0.85	**1.00**	0.85	**1.00**	0.85	**1.00**	0.85
FordA	0.50	0.25	**0.90**	0.50	**0.90**	0.35	0.85	0.35
FordB	0.60	0.10	**0.90**	0.40	0.70	0.35	0.85	0.50
ShapeletSim	0.45	0.20	0.40	0.15	**0.65**	0.40	0.45	0.00
BeetleFly	0.70	0.25	0.90	0.55	0.80	0.30	**1.00**	0.70
BirdChicken	**1.00**	0.40	**1.00**	0.60	**1.00**	0.45	**1.00**	0.90
Earthquakes	0.60	0.40	**0.65**	0.35	0.35	0.25	0.45	0.15
Herring	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
OliveOil	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Car	**1.00**	0.90	**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**
Lighting2	0.95	0.25	**1.00**	0.20	0.95	0.45	**1.00**	0.40
Computers	**0.95**	0.50	**0.95**	0.45	**0.95**	0.55	0.90	0.60
LargeKitchenAppliances	0.95	0.62	0.95	0.67	0.90	0.62	**1.00**	0.67
RefrigerationDevices	0.81	0.24	**0.95**	0.48	0.71	0.33	0.90	0.43
ScreenType	0.57	0.19	**0.71**	0.43	0.57	0.29	0.67	0.29
SmallKitchenAppliances	0.90	0.43	0.90	0.62	0.90	0.43	**1.00**	0.71
WormsTwoClass	0.65	0.25	0.75	0.35	0.65	0.15	**0.95**	0.65
Worms	0.80	0.20	0.80	0.20	**0.90**	0.35	**0.90**	0.25
StarLightCurves	**1.00**	0.90	**1.00**	0.90	**1.00**	0.90	**1.00**	0.90
Haptics	**1.00**	0.40	0.95	0.95	**1.00**	0.45	0.95	0.95

Table C.6: AoD from all algorithms on all datasets when the scaling factor (f) is 1.2 and number of clusters (k) is set to the number of classes in each dataset

Dataset	Accuracy on Detection (AoD)							
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.84	0.93	0.92	**0.94**	0.88	0.93	0.91	0.94
SonyAIBORobotSurfaceII	0.97	0.99	0.89	**0.99**	0.93	0.96	0.91	0.94
SonyAIBORobotSurface	0.83	0.90	0.89	**0.93**	0.81	0.92	0.88	0.92
DistalPhalanxOutlineCorrect	0.90	0.90	0.90	0.90	0.91	0.91	**0.95**	**0.95**
MiddlePhalanxOutlineCorrect	0.88	0.89	0.87	0.87	0.87	0.88	**0.94**	**0.94**
PhalangesOutlinesCorrect	0.93	0.93	0.96	**0.96**	0.95	0.95	0.93	0.93
ProximalPhalanxOutlineCorrect	0.87	0.87	0.88	0.88	0.88	0.88	**0.94**	**0.94**
DistalPhalanxOutlineAgeGroup	0.84	0.89	0.87	0.87	0.84	0.89	**0.94**	**0.94**
MiddlePhalanxOutlineAgeGroup	0.94	0.94	**0.94**	**0.94**	0.94	0.94	0.94	0.94
ProximalPhalanxOutlineAgeGroup	0.93	0.94	0.92	0.92	0.92	0.92	**0.94**	**0.94**
TwoLeadECG	0.90	0.91	0.92	0.92	0.90	0.91	**0.95**	**0.95**
MoteStrain	0.80	0.89	0.86	0.90	0.77	0.89	0.91	**0.93**
ECG200	0.86	0.92	0.85	0.87	0.84	0.90	0.91	**0.93**
CBF	0.74	0.90	0.87	**0.94**	0.73	0.89	0.89	0.92
Two_Patterns	0.77	0.87	0.74	0.85	0.71	0.91	0.79	**0.95**
ECGFiveDays	0.92	0.92	0.95	0.95	0.92	0.92	**0.96**	**0.96**
ECG5000	0.76	0.89	0.83	0.86	0.79	0.88	0.88	**0.91**
Gun_Point	0.91	**0.92**	0.92	**0.92**	0.89	0.90	0.91	0.91
wafer	0.80	0.89	0.83	0.87	0.80	0.88	0.90	**0.92**
ChlorineConcentration	0.86	0.94	0.88	0.91	0.85	0.93	**0.94**	**0.94**
Wine	0.86	0.86	0.92	0.92	0.87	0.87	**0.92**	**0.92**
Strawberry	0.90	0.90	0.91	0.91	0.90	0.90	**0.95**	**0.95**
ArrowHead	0.94	0.96	0.91	0.91	0.90	0.94	**0.98**	**0.98**
Trace	0.87	0.89	0.90	0.91	0.87	0.89	0.93	**0.93**
ToeSegmentation1	0.77	**0.91**	0.82	0.87	0.82	0.90	0.83	0.88
Coffee	0.91	0.91	0.90	0.90	0.92	0.92	**0.96**	**0.96**
ToeSegmentation2	0.79	0.89	0.84	0.89	0.78	0.89	0.81	**0.92**
FaceFour	0.70	**0.96**	0.83	0.89	0.72	0.88	0.85	0.90
yoga	0.87	0.91	0.91	**0.93**	0.90	0.92	0.92	0.93
Ham	0.74	0.92	0.86	**0.96**	0.77	0.92	0.79	0.94
Meat	0.91	0.91	0.92	0.92	0.92	0.92	**0.98**	**0.98**
Beef	0.90	0.92	0.88	0.93	0.90	0.92	0.91	**0.97**
FordA	0.64	0.86	0.77	0.89	0.72	0.90	0.74	**1.02**
FordB	0.64	0.82	0.78	0.99	0.71	**1.00**	0.75	0.90
ShapeletSim	0.69	**0.92**	0.67	0.89	0.69	0.88	0.60	0.00
BeetleFly	0.65	0.87	0.75	0.87	0.65	0.87	0.81	**0.89**
BirdChicken	0.79	0.88	0.82	0.87	0.79	0.89	0.91	**0.92**
Earthquakes	0.74	0.92	0.72	0.91	0.70	**0.96**	0.67	0.96
Herring	0.86	0.86	0.87	0.87	0.87	0.87	**0.92**	**0.92**
OliveOil	0.86	0.86	0.87	0.87	0.88	0.88	**0.94**	**0.94**
Car	0.84	0.85	0.87	0.87	0.84	0.85	**0.92**	**0.92**
Lighting2	0.73	0.92	0.71	0.92	0.79	**0.95**	0.78	0.92
Computers	0.75	0.88	0.74	0.86	0.78	0.88	0.81	**0.91**
LargeKitchenAppliances	0.82	0.91	0.82	0.89	0.79	0.90	0.84	**0.92**
RefrigerationDevices	0.67	0.88	0.79	**0.94**	0.73	0.93	0.77	0.92
ScreenType	0.65	0.83	0.71	0.85	0.69	**0.91**	0.68	0.87
SmallKitchenAppliances	0.74	**0.90**	0.81	0.88	0.73	0.87	0.84	0.89
WormsTwoClass	0.68	0.93	0.74	0.86	0.66	**0.97**	0.80	0.92
Worms	0.67	0.88	0.66	0.88	0.69	0.86	0.71	**0.89**
StarLightCurves	0.89	0.91	0.90	0.92	0.90	0.92	0.91	**0.93**
Haptics	0.77	**0.92**	0.87	0.87	0.78	0.92	0.87	0.87

Table C.7: Excess Rate from all algorithms on all datasets when the scaling factor (f) is 1.2 and number of clusters (k) is set to the number of classes in each dataset

Dataset	Excess Rate							
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.09	0.50	**0.05**	0.19	**0.05**	0.38	0.09	0.23
SonyAIBORobotSurfaceII	0.23	0.31	0.23	0.54	0.27	0.31	**0.13**	0.26
SonyAIBORobotSurface	0.27	0.45	0.20	0.35	0.36	0.55	**0.11**	0.28
DistalPhalanxOutlineCorrect	0.09	0.09	0.09	0.09	**0.05**	**0.05**	**0.05**	**0.05**
MiddlePhalanxOutlineCorrect	0.05	0.14	0.09	0.09	0.09	0.23	**0.00**	**0.00**
PhalangesOutlinesCorrect	0.05	0.05	0.05	0.10	0.05	0.05	**0.00**	**0.00**
ProximalPhalanxOutlineCorrect	0.05	0.05	0.05	0.05	0.05	0.05	**0.00**	**0.00**
DistalPhalanxOutlineAgeGroup	0.05	0.27	**0.00**	**0.00**	0.05	0.32	**0.00**	**0.00**
MiddlePhalanxOutlineAgeGroup	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
ProximalPhalanxOutlineAgeGroup	**0.00**	0.05	**0.00**	**0.00**	**0.00**	0.05	**0.00**	**0.00**
TwoLeadECG	0.05	0.14	0.05	0.10	0.05	0.14	**0.00**	**0.00**
MoteStrain	0.23	0.50	0.09	0.32	0.30	0.56	**0.05**	0.19
ECG200	0.09	0.32	**0.00**	0.15	0.09	0.27	**0.00**	0.10
CBF	0.32	0.68	0.16	0.48	0.35	0.65	**0.00**	0.14
Two_Patterns	0.36	0.64	**0.17**	0.63	0.31	0.77	0.23	0.45
ECGFiveDays	0.05	0.05	0.14	0.14	0.05	0.05	**0.00**	**0.00**
ECG5000	0.23	0.50	**0.00**	0.17	0.18	0.50	**0.00**	0.17
Gun_Point	0.09	0.14	0.09	0.14	0.09	0.18	**0.00**	**0.00**
wafer	0.25	0.50	0.05	0.33	0.22	0.48	**0.00**	0.15
ChlorineConcentration	**0.00**	0.43	**0.00**	0.19	**0.00**	0.43	**0.00**	**0.00**
Wine	0.13	0.13	0.09	0.09	0.13	0.13	**0.00**	**0.00**
Strawberry	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
ArrowHead	0.13	0.17	**0.00**	**0.00**	0.13	0.25	**0.00**	**0.00**
Trace	**0.00**	0.20	**0.00**	0.10	**0.00**	0.15	**0.00**	0.05
ToeSegmentation1	0.31	0.58	**0.13**	0.39	0.23	0.50	**0.13**	0.35
Coffee	0.09	0.09	0.05	0.05	0.13	0.13	**0.00**	**0.00**
ToeSegmentation2	0.35	0.50	**0.13**	0.35	0.31	0.58	**0.13**	0.48
FaceFour	0.25	0.88	**0.00**	0.33	0.26	0.74	**0.00**	0.33
yoga	0.17	0.25	**0.00**	0.10	0.17	0.25	0.13	0.17
Ham	0.25	0.75	**0.24**	0.52	**0.24**	0.64	0.30	0.52
Meat	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Beef	**0.00**	0.15	0.09	0.23	**0.00**	0.15	0.05	0.19
FordA	0.60	0.80	0.33	0.63	**0.31**	0.73	0.32	0.72
FordB	0.52	0.92	0.33	0.70	0.50	0.75	**0.32**	0.60
ShapeletSim	0.64	0.84	0.69	0.88	**0.54**	0.71	0.57	1.00
BeetleFly	0.46	0.81	0.28	0.56	0.38	0.77	**0.17**	0.42
BirdChicken	0.09	0.64	**0.05**	0.43	0.13	0.61	0.09	0.18
Earthquakes	0.56	0.70	**0.54**	0.75	0.71	0.79	0.61	0.87
Herring	0.05	0.05	0.05	0.05	0.05	0.05	**0.00**	**0.00**
OliveOil	0.05	0.05	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Car	0.05	0.14	**0.00**	**0.00**	0.05	0.14	**0.00**	**0.00**
Lighting2	0.30	0.81	0.23	0.85	0.27	0.65	**0.13**	0.65
Computers	0.24	0.60	0.27	0.65	**0.17**	0.52	0.18	0.45
LargeKitchenAppliances	0.35	0.58	0.33	0.53	0.37	0.57	**0.13**	0.42
RefrigerationDevices	0.41	0.83	0.29	0.64	0.44	0.74	**0.27**	0.65
ScreenType	0.56	0.85	0.42	0.65	0.57	0.79	**0.39**	0.74
SmallKitchenAppliances	0.37	0.70	0.27	0.50	0.37	0.70	**0.19**	0.42
WormsTwoClass	0.48	0.80	0.40	0.72	0.46	0.88	**0.27**	0.50
Worms	0.43	0.86	0.38	0.85	0.36	0.75	**0.33**	0.81
StarLightCurves	**0.09**	0.17	**0.09**	0.17	**0.09**	0.17	**0.09**	0.17
Haptics	0.09	0.64	**0.00**	**0.00**	0.09	0.59	**0.00**	**0.00**

Table C.8: Rand Index (RI) from all algorithms on all datasets when the scaling factor (f) is 1.2 and number of clusters (k) is chosen by the SSTSC algorithms

Dataset	Rand Index							
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.64	0.60	**0.69**	0.67	0.64	0.59	0.58	0.58
SonyAIBORobotSurfaceII	0.65	**0.73**	0.55	0.64	0.61	0.69	0.59	0.58
SonyAIBORobotSurface	0.50	0.44	**0.60**	0.51	0.41	0.14	0.48	0.46
DistalPhalanxOutlineCorrect	0.49	0.49	0.49	0.49	**0.51**	**0.51**	0.49	0.49
MiddlePhalanxOutlineCorrect	0.55	0.55	**0.63**	**0.63**	0.54	0.51	0.52	0.52
PhalangesOutlinesCorrect	0.49	0.49	0.50	0.49	0.49	0.49	**0.53**	**0.53**
ProximalPhalanxOutlineCorrect	0.54	0.54	**0.54**	**0.54**	0.54	0.54	0.54	0.54
DistalPhalanxOutlineAgeGroup	0.67	0.62	**0.69**	**0.69**	0.57	0.44	0.64	0.64
MiddlePhalanxOutlineAgeGroup	0.59	0.59	**0.61**	**0.61**	0.59	0.59	0.53	0.53
ProximalPhalanxOutlineAgeGroup	**0.76**	0.74	0.65	0.65	0.75	0.73	0.60	0.60
TwoLeadECG	0.56	0.57	**0.61**	0.60	0.56	0.57	0.56	0.56
MoteStrain	0.57	0.67	0.59	0.63	0.63	**0.73**	0.67	0.69
ECG200	0.63	**0.64**	0.57	0.54	0.58	0.60	0.57	0.56
CBF	0.78	0.70	**0.88**	0.87	0.76	0.33	0.77	0.76
Two_Patterns	0.82	0.81	0.73	0.78	0.82	**0.87**	0.80	0.85
ECGFiveDays	0.51	0.51	0.53	0.54	0.51	0.51	**0.58**	**0.58**
ECG5000	**0.74**	0.53	0.63	0.66	**0.74**	0.62	0.71	0.69
Gun_Point	0.54	0.56	**0.61**	0.60	0.54	0.56	0.54	0.54
wafer	0.64	**0.71**	0.66	0.59	0.68	**0.71**	0.55	0.57
ChlorineConcentration	0.64	0.58	0.63	0.64	0.64	**0.68**	0.68	0.68
Wine	0.49	0.49	0.49	0.49	**0.53**	**0.53**	0.52	0.52
Strawberry	0.49	0.49	**0.51**	**0.51**	0.48	0.48	**0.51**	**0.51**
ArrowHead	0.62	0.58	**0.73**	**0.73**	0.68	0.67	0.63	0.63
Trace	0.69	0.83	0.79	**0.84**	0.69	0.82	0.75	0.80
ToeSegmentation1	0.63	**0.64**	0.53	0.53	0.57	0.52	0.58	0.58
Coffee	**0.57**	**0.57**	0.55	0.55	0.55	0.55	0.55	0.55
ToeSegmentation2	0.62	0.64	0.62	**0.65**	0.56	0.60	0.57	0.60
FaceFour	0.76	0.33	**0.82**	0.68	0.78	0.40	0.82	0.80
yoga	0.51	0.53	0.49	0.50	0.52	0.52	**0.54**	**0.54**
Ham	0.56	0.50	0.48	0.46	0.52	0.53	0.58	**0.67**
Meat	**0.74**	**0.74**	0.57	0.57	0.71	0.71	0.62	0.62
Beef	0.77	0.70	0.75	0.67	0.75	0.70	**0.78**	0.74
FordA	0.67	0.00	0.49	0.52	0.51	0.67	0.55	**1.00**
FordB	0.51	0.00	0.50	**0.60**	0.53	0.40	0.50	0.40
ShapeletSim	0.44	0.33	**0.57**	0.33	0.43	0.00	0.44	0.00
BeetleFly	0.58	**1.00**	0.55	0.19	0.64	0.70	0.55	0.61
BirdChicken	0.57	0.57	0.61	0.61	0.57	0.57	0.61	**0.62**
Earthquakes	0.33	0.00	0.68	**0.71**	0.43	0.40	0.54	0.00
Herring	**0.54**	**0.54**	0.53	0.53	0.52	0.52	0.50	0.50
OliveOil	**0.71**	**0.71**	**0.71**	**0.71**	0.69	0.69	0.68	0.68
Car	0.79	**0.80**	0.74	0.74	0.79	**0.80**	0.75	0.75
Lighting2	0.47	0.40	0.55	**1.00**	0.52	**1.00**	0.53	0.67
Computers	0.56	**0.60**	0.47	0.52	0.56	0.55	0.50	0.55
LargeKitchenAppliances	**0.69**	0.58	0.57	0.67	0.69	0.60	0.54	0.69
RefrigerationDevices	0.67	0.00	0.57	0.40	0.67	0.33	**0.68**	0.60
ScreenType	0.57	0.00	0.55	0.57	0.40	0.00	**0.66**	0.60
SmallKitchenAppliances	0.69	**0.76**	0.63	0.50	0.69	**0.76**	0.61	0.59
WormsTwoClass	0.43	**1.00**	0.45	0.00	0.54	**1.00**	0.50	0.27
Worms	0.84	**1.00**	0.73	0.83	0.75	**1.00**	0.82	0.83
StarLightCurves	**0.74**	0.74	**0.74**	0.74	**0.74**	0.74	**0.74**	0.74
Haptics	**0.79**	0.75	0.66	0.66	0.77	0.75	0.67	0.67

Table C.9: Precision from all algorithms on all datasets when the scaling factor (f) is 1.2 and number of clusters (k) is chosen by the SSTSC algorithms

Dataset	Precision							
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	**0.92**	0.71	0.86	0.79	**0.92**	0.75	0.63	0.59
SonyAIBORobotSurfaceII	**1.00**	**1.00**	0.61	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
SonyAIBORobotSurface	0.50	0.58	0.80	**1.00**	0.63	**1.00**	0.47	0.50
DistalPhalanxOutlineCorrect	0.42	0.42	0.42	0.42	**0.46**	**0.46**	0.41	0.41
MiddlePhalanxOutlineCorrect	0.57	0.56	**0.73**	**0.73**	0.52	0.50	0.49	0.49
PhalangesOutlinesCorrect	0.46	0.46	0.45	0.44	0.46	0.46	**0.50**	**0.50**
ProximalPhalanxOutlineCorrect	0.52	0.52	**0.53**	**0.53**	0.52	0.52	0.52	0.52
DistalPhalanxOutlineAgeGroup	0.43	0.42	**0.46**	**0.46**	0.32	0.32	0.39	0.39
MiddlePhalanxOutlineAgeGroup	**0.34**	**0.34**	0.30	0.30	**0.34**	**0.34**	0.26	0.26
ProximalPhalanxOutlineAgeGroup	**0.65**	0.61	0.40	0.40	0.61	0.57	0.30	0.30
TwoLeadECG	0.59	0.63	**0.63**	0.62	0.59	0.63	0.56	0.56
MoteStrain	0.73	0.89	0.70	0.79	**1.00**	**1.00**	0.93	0.96
ECG200	0.73	0.68	0.60	0.52	**0.78**	0.73	0.57	0.54
CBF	0.84	**1.00**	0.95	0.93	**1.00**	0.33	**1.00**	**1.00**
Two_Patterns	0.64	**1.00**	0.28	0.33	0.67	**1.00**	0.57	**1.00**
ECGFiveDays	0.46	0.46	0.52	0.55	0.46	0.46	**0.80**	**0.80**
ECG5000	**0.28**	0.23	0.18	0.26	0.15	0.18	0.26	0.27
Gun_Point	0.52	0.55	**0.64**	0.62	0.52	0.55	0.53	0.53
wafer	0.70	0.77	0.90	**1.00**	0.74	0.77	0.63	0.71
ChlorineConcentration	0.24	0.20	0.20	0.20	0.25	0.29	**0.33**	**0.33**
Wine	0.44	0.44	0.38	0.38	**0.51**	**0.51**	0.46	0.46
Strawberry	0.43	0.43	0.47	0.47	0.45	0.45	**0.48**	**0.48**
ArrowHead	0.34	0.33	**0.59**	**0.59**	0.42	0.43	0.39	0.39
Trace	0.38	0.56	0.51	**0.56**	0.38	0.55	0.42	0.51
ToeSegmentation1	0.79	**1.00**	0.50	0.43	0.75	0.60	0.78	0.75
Coffee	**0.60**	**0.60**	0.58	0.58	0.57	0.57	0.53	0.53
ToeSegmentation2	**1.00**	**1.00**	0.84	0.94	0.71	**1.00**	0.90	**1.00**
FaceFour	0.44	**1.00**	0.58	0.48	0.50	0.50	0.78	0.80
yoga	0.47	0.48	0.39	0.40	0.45	0.44	**0.54**	**0.54**
Ham	0.59	0.33	0.41	0.38	0.44	0.00	**0.67**	**0.67**
Meat	**0.66**	**0.66**	0.32	0.32	0.53	0.53	0.35	0.35
Beef	0.25	0.16	0.24	0.16	0.19	0.16	**0.26**	0.23
FordA	**1.00**	0.00	0.48	0.52	0.50	0.00	0.67	**1.00**
FordB	0.38	0.00	0.44	0.50	**0.60**	0.50	0.25	0.00
ShapeletSim	0.39	0.33	**0.52**	0.33	0.33	0.00	0.44	0.00
BeetleFly	0.73	**1.00**	0.59	0.19	**1.00**	**1.00**	0.73	**1.00**
BirdChicken	0.83	**1.00**	**1.00**	**1.00**	0.90	**1.00**	0.94	**1.00**
Earthquakes	0.33	0.00	0.64	**0.67**	0.33	0.33	0.33	0.00
Herring	**0.54**	**0.54**	0.50	0.50	0.47	0.47	0.45	0.45
OliveOil	0.19	0.19	**0.29**	**0.29**	0.19	0.19	0.20	0.20
Car	0.50	**0.51**	0.40	0.40	0.50	**0.51**	0.41	0.41
Lighting2	0.45	0.50	0.54	**1.00**	0.48	**1.00**	0.52	**1.00**
Computers	0.67	**0.80**	0.55	0.57	0.67	0.67	0.45	0.50
LargeKitchenAppliances	0.47	**0.67**	0.40	0.51	0.54	**0.67**	0.35	0.50
RefrigerationDevices	0.45	0.00	0.27	0.20	0.25	0.00	0.29	**0.50**
ScreenType	**0.25**	0.00	0.18	0.00	**0.25**	0.00	0.14	0.00
SmallKitchenAppliances	0.45	**0.50**	0.42	0.25	0.46	**0.50**	0.30	0.22
WormsTwoClass	0.50	**1.00**	0.14	0.00	0.57	**1.00**	0.29	0.00
Worms	**0.67**	0.00	0.11	0.00	0.33	0.00	0.33	0.00
StarLightCurves	**0.62**	0.61	**0.62**	0.61	0.57	0.56	0.57	0.56
Haptics	**0.26**	0.00	0.12	0.12	0.19	0.14	0.13	0.13

Table C.10: Recall from all algorithms on all datasets when the scaling factor (f) is 1.2 and number of clusters (k) is chosen by the SSTSC algorithms

Dataset	Recall							
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.27	0.20	**0.42**	0.40	0.27	0.19	0.27	0.30
SonyAIBORobotSurfaceII	0.27	**0.45**	0.16	0.28	0.17	0.33	0.14	0.17
SonyAIBORobotSurface	0.13	0.18	0.27	0.27	0.10	0.14	0.47	**0.67**
DistalPhalanxOutlineCorrect	0.18	0.18	0.18	0.18	**0.21**	**0.21**	0.18	0.18
MiddlePhalanxOutlineCorrect	0.22	0.26	0.33	0.33	0.42	**0.53**	0.38	0.38
PhalangesOutlinesCorrect	**0.38**	**0.38**	0.27	0.27	**0.38**	**0.38**	0.20	0.20
ProximalPhalanxOutlineCorrect	0.27	0.27	**0.33**	**0.33**	0.27	0.27	0.27	0.27
DistalPhalanxOutlineAgeGroup	0.33	0.49	0.21	0.21	0.40	**0.65**	0.35	0.35
MiddlePhalanxOutlineAgeGroup	**0.41**	**0.41**	0.22	0.22	**0.41**	**0.41**	0.32	0.32
ProximalPhalanxOutlineAgeGroup	0.44	0.40	0.35	0.35	**0.48**	0.44	0.27	0.27
TwoLeadECG	0.21	0.23	0.40	**0.42**	0.21	0.23	0.31	0.31
MoteStrain	0.15	0.32	0.26	0.26	0.20	**0.40**	0.32	**0.40**
ECG200	0.33	**0.43**	0.29	0.25	0.16	0.22	0.42	0.39
CBF	0.35	0.50	**0.62**	0.59	0.27	0.33	0.22	0.22
Two_Patterns	0.25	0.43	0.18	0.14	0.21	**0.50**	0.16	0.29
ECGFiveDays	**0.21**	**0.21**	0.14	0.17	**0.21**	**0.21**	0.15	0.15
ECG5000	0.41	**0.56**	0.36	0.38	0.14	0.30	0.44	0.50
Gun_Point	0.39	**0.44**	0.39	**0.44**	0.39	**0.44**	0.20	0.20
wafer	0.42	**0.55**	0.31	0.27	0.51	**0.55**	0.11	0.16
ChlorineConcentration	0.09	**0.12**	0.08	0.08	0.10	0.11	0.08	0.08
Wine	**0.24**	**0.24**	0.12	0.12	0.20	0.20	0.14	0.14
Strawberry	0.26	0.26	0.22	0.22	**0.50**	**0.50**	0.39	0.39
ArrowHead	0.29	0.37	0.33	0.33	0.21	0.29	**0.42**	**0.42**
Trace	0.70	0.83	0.90	**1.00**	0.70	0.86	0.45	0.50
ToeSegmentation1	0.30	**0.33**	0.11	0.08	0.15	0.18	0.16	0.14
Coffee	0.29	0.29	0.20	0.20	0.22	0.22	**0.43**	**0.43**
ToeSegmentation2	0.19	**0.33**	0.23	0.33	0.10	0.14	0.11	0.19
FaceFour	0.33	0.33	0.64	**0.70**	0.15	0.17	0.21	0.25
yoga	0.17	**0.23**	0.12	0.14	0.10	0.14	0.16	0.16
Ham	0.18	0.25	0.26	0.42	0.06	0.00	0.20	**0.50**
Meat	0.30	0.30	**0.40**	**0.40**	0.27	0.27	0.32	0.32
Beef	0.23	0.17	**0.27**	0.21	0.19	0.17	0.20	0.21
FordA	0.36	0.00	0.79	**1.00**	0.08	0.00	0.08	**1.00**
FordB	0.12	0.00	0.11	**0.33**	0.08	0.17	0.03	0.00
ShapeletSim	0.44	**1.00**	0.85	**1.00**	0.33	0.00	0.44	0.00
BeetleFly	0.21	**1.00**	0.16	0.19	0.20	0.50	0.09	0.22
BirdChicken	0.12	**0.25**	0.18	0.24	0.11	**0.25**	0.20	0.23
Earthquakes	0.33	0.00	0.69	**0.91**	0.33	0.50	0.08	0.00
Herring	0.21	0.21	**0.29**	**0.29**	0.18	0.18	0.24	0.24
OliveOil	0.11	0.11	**0.25**	**0.25**	0.14	0.14	0.17	0.17
Car	0.48	**0.59**	0.48	0.48	0.48	**0.59**	0.38	0.38
Lighting2	0.46	0.33	0.31	**1.00**	0.14	**1.00**	0.17	0.29
Computers	0.12	0.19	0.33	**0.36**	0.12	0.15	0.14	0.16
LargeKitchenAppliances	0.14	0.33	0.76	**1.00**	0.15	0.24	0.63	0.79
RefrigerationDevices	0.24	0.00	0.23	**0.67**	0.04	0.00	0.04	0.25
ScreenType	0.14	0.00	**0.15**	0.00	0.14	0.00	0.04	0.00
SmallKitchenAppliances	0.11	0.40	**0.54**	0.40	0.12	0.40	0.22	0.18
WormsTwoClass	0.13	**1.00**	0.03	0.00	0.57	**1.00**	0.04	0.00
Worms	**0.42**	0.00	0.11	0.00	0.25	0.00	0.12	0.00
StarLightCurves	0.33	0.33	0.33	0.33	0.52	**0.53**	0.52	**0.53**
Haptics	0.19	0.00	0.19	0.19	0.15	**0.25**	0.19	0.19

Table C.11: F1 from all algorithms on all datasets when the scaling factor (f) is 1.2 and number of clusters (k) is chosen by the SSTSC algorithms

Dataset	F1-Measure							
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.41	0.31	**0.57**	0.53	0.41	0.31	0.38	0.40
SonyAIBORobotSurfaceII	0.42	**0.62**	0.25	0.43	0.29	0.50	0.25	0.29
SonyAIBORobotSurface	0.21	0.27	0.40	0.43	0.18	0.25	0.47	**0.57**
DistalPhalanxOutlineCorrect	0.25	0.25	0.25	0.25	**0.29**	**0.29**	0.25	0.25
MiddlePhalanxOutlineCorrect	0.32	0.36	0.46	0.46	0.47	**0.51**	0.43	0.43
PhalangesOutlinesCorrect	**0.41**	**0.41**	0.34	0.34	**0.41**	**0.41**	0.28	0.28
ProximalPhalanxOutlineCorrect	0.35	0.35	**0.41**	**0.41**	0.35	0.35	0.35	0.35
DistalPhalanxOutlineAgeGroup	0.38	**0.45**	0.29	0.29	0.35	0.43	0.37	0.37
MiddlePhalanxOutlineAgeGroup	**0.37**	**0.37**	0.26	0.26	**0.37**	**0.37**	0.29	0.29
ProximalPhalanxOutlineAgeGroup	0.53	0.48	0.37	0.37	**0.54**	0.50	0.29	0.29
TwoLeadECG	0.31	0.34	0.49	**0.50**	0.31	0.34	0.40	0.40
MoteStrain	0.25	0.47	0.37	0.39	0.34	**0.57**	0.48	0.56
ECG200	0.46	**0.53**	0.39	0.34	0.26	0.34	0.48	0.45
CBF	0.49	0.67	**0.75**	0.72	0.42	0.33	0.35	0.36
Two_Patterns	0.36	0.60	0.22	0.20	0.32	**0.67**	0.25	0.44
ECGFiveDays	**0.29**	**0.29**	0.22	0.26	**0.29**	**0.29**	0.25	0.25
ECG5000	0.33	0.32	0.24	0.31	0.14	0.22	0.33	**0.35**
Gun_Point	0.45	0.49	0.48	**0.51**	0.45	0.49	0.29	0.29
wafer	0.53	**0.64**	0.46	0.43	0.60	**0.64**	0.19	0.26
ChlorineConcentration	0.13	0.15	0.11	0.11	0.14	**0.16**	0.13	0.13
Wine	**0.31**	**0.31**	0.18	0.18	0.29	0.29	0.22	0.22
Strawberry	0.32	0.32	0.30	0.30	**0.48**	**0.48**	0.43	0.43
ArrowHead	0.31	0.35	**0.43**	**0.43**	0.28	0.35	0.41	0.41
Trace	0.49	0.67	0.65	**0.72**	0.49	0.67	0.43	0.51
ToeSegmentation1	0.44	**0.50**	0.18	0.14	0.25	0.27	0.26	0.24
Coffee	0.39	0.39	0.30	0.30	0.32	0.32	**0.48**	**0.48**
ToeSegmentation2	0.32	**0.50**	0.37	0.48	0.18	0.25	0.20	0.31
FaceFour	0.38	0.50	**0.61**	0.57	0.23	0.25	0.33	0.38
yoga	0.25	**0.31**	0.19	0.21	0.16	0.22	0.25	0.25
Ham	0.28	0.29	0.32	0.40	0.10	0.00	0.31	**0.57**
Meat	**0.41**	**0.41**	0.35	0.35	0.36	0.36	0.33	0.33
Beef	0.24	0.16	**0.25**	0.18	0.19	0.16	0.23	0.22
FordA	0.53	0.00	0.59	0.69	0.13	0.00	0.15	**1.00**
FordB	0.18	0.00	0.17	**0.40**	0.14	0.25	0.06	0.00
ShapeletSim	0.41	0.50	**0.65**	0.50	0.33	0.00	0.44	0.00
BeetleFly	0.33	**1.00**	0.25	0.19	0.33	0.67	0.16	0.36
BirdChicken	0.22	**0.40**	0.30	0.38	0.20	**0.40**	0.33	0.38
Earthquakes	0.33	0.00	0.67	**0.77**	0.33	0.40	0.13	0.00
Herring	0.30	0.30	**0.37**	**0.37**	0.26	0.26	0.32	0.32
OliveOil	0.14	0.14	**0.27**	**0.27**	0.16	0.16	0.18	0.18
Car	0.49	**0.55**	0.44	0.44	0.49	**0.55**	0.39	0.39
Lighting2	0.45	0.40	0.40	**1.00**	0.22	**1.00**	0.26	0.44
Computers	0.21	0.31	0.41	**0.44**	0.21	0.24	0.21	0.24
LargeKitchenAppliances	0.22	0.44	0.52	**0.68**	0.23	0.35	0.46	0.61
RefrigerationDevices	0.31	0.00	0.25	0.31	0.06	0.00	0.08	**0.33**
ScreenType	**0.18**	0.00	0.16	0.00	**0.18**	0.00	0.06	0.00
SmallKitchenAppliances	0.18	0.44	**0.47**	0.31	0.18	0.44	0.25	0.20
WormsTwoClass	0.20	**1.00**	0.04	0.00	0.57	**1.00**	0.06	0.00
Worms	**0.52**	0.00	0.11	0.00	0.29	0.00	0.17	0.00
StarLightCurves	0.43	0.43	0.43	0.43	**0.55**	**0.55**	**0.55**	**0.55**
Haptics	**0.22**	0.00	0.15	0.15	0.17	0.18	0.15	0.15

Table C.12: AoR from all algorithms on all datasets when the scaling factor (f) is 1.2 and number of clusters (k) is chosen by the SSTSC algorithms

Dataset	Accuracy on Retrieval (AoR)							
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	**1.00**	0.55	**1.00**	0.80	**1.00**	0.60	**1.00**	0.80
SonyAIBORobotSurfaceII	**1.00**	0.65	**1.00**	0.45	0.95	0.65	**1.00**	0.80
SonyAIBORobotSurface	**1.00**	0.75	**1.00**	0.69	0.81	0.50	**1.00**	0.81
DistalPhalanxOutlineCorrect	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
MiddlePhalanxOutlineCorrect	**1.00**	0.90	**1.00**	**1.00**	**1.00**	0.85	**1.00**	**1.00**
PhalangesOutlinesCorrect	**1.00**	**1.00**	**1.00**	0.95	**1.00**	**1.00**	0.95	0.95
ProximalPhalanxOutlineCorrect	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
DistalPhalanxOutlineAgeGroup	**1.00**	0.76	**1.00**	**1.00**	**1.00**	0.71	**1.00**	**1.00**
MiddlePhalanxOutlineAgeGroup	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
ProximalPhalanxOutlineAgeGroup	**1.00**	0.95	**1.00**	**1.00**	**1.00**	0.95	**1.00**	**1.00**
TwoLeadECG	**1.00**	0.90	**1.00**	0.95	**1.00**	0.90	**1.00**	**1.00**
MoteStrain	0.90	0.55	**1.00**	0.70	0.75	0.50	0.95	0.80
ECG200	**1.00**	0.75	**1.00**	0.85	**1.00**	0.75	**1.00**	0.90
CBF	0.86	0.24	**1.00**	0.62	0.62	0.14	0.90	0.86
Two_Patterns	0.85	0.35	**1.00**	0.45	0.85	0.30	0.80	0.60
ECGFiveDays	**1.00**	**1.00**	**1.00**	0.89	**1.00**	**1.00**	**1.00**	**1.00**
ECG5000	0.94	0.56	**1.00**	0.83	0.94	0.61	**1.00**	0.83
Gun_Point	**1.00**	0.90	**1.00**	0.90	**1.00**	0.90	**1.00**	**1.00**
wafer	0.90	0.60	**1.00**	0.70	0.90	0.60	**1.00**	0.85
ChlorineConcentration	0.95	0.52	**1.00**	0.81	**1.00**	0.57	**1.00**	**1.00**
Wine	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Strawberry	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
ArrowHead	**1.00**	0.86	0.95	0.95	**1.00**	0.81	0.95	0.95
Trace	**1.00**	0.80	**1.00**	0.90	**1.00**	0.85	**1.00**	0.95
ToeSegmentation1	0.90	0.50	**1.00**	0.65	0.95	0.60	**1.00**	0.70
Coffee	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
ToeSegmentation2	0.85	0.50	**1.00**	0.70	0.75	0.50	0.95	0.55
FaceFour	**1.00**	0.17	**1.00**	0.67	0.89	0.28	**1.00**	0.67
yoga	0.95	0.80	**1.00**	0.90	**1.00**	0.80	0.95	0.95
Ham	**0.90**	0.25	**0.90**	0.40	**0.90**	0.30	0.75	0.45
Meat	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Beef	**1.00**	0.85	**1.00**	0.85	0.95	0.85	**1.00**	0.85
FordA	0.35	0.00	**0.90**	0.35	0.85	0.15	0.75	0.05
FordB	0.55	0.10	**0.85**	0.30	0.65	0.25	0.60	0.25
ShapeletSim	**0.45**	0.20	0.40	0.15	0.35	0.10	**0.45**	0.00
BeetleFly	0.65	0.25	0.85	0.35	0.60	0.25	**1.00**	0.45
BirdChicken	0.95	0.40	**1.00**	0.60	0.95	0.40	0.95	0.80
Earthquakes	0.15	0.00	**0.65**	0.35	0.35	0.25	0.40	0.10
Herring	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
OliveOil	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Car	**1.00**	0.90	**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**
Lighting2	**0.95**	0.25	0.85	0.05	0.90	0.20	**0.95**	0.30
Computers	**0.95**	0.50	0.70	0.35	**0.95**	0.55	0.85	0.55
LargeKitchenAppliances	**0.95**	0.43	0.81	0.52	0.86	0.52	0.90	0.62
RefrigerationDevices	0.57	0.00	**0.90**	0.29	0.67	0.14	0.86	0.24
ScreenType	0.33	0.10	**0.67**	0.33	0.29	0.10	**0.67**	0.24
SmallKitchenAppliances	0.86	0.33	0.76	0.43	**0.90**	0.33	0.86	0.62
WormsTwoClass	0.40	0.05	0.65	0.15	0.65	0.15	**0.80**	0.30
Worms	0.70	0.15	**0.80**	0.20	**0.80**	0.20	0.75	0.20
StarLightCurves	**1.00**	0.90	**1.00**	0.90	**1.00**	0.90	**1.00**	0.90
Haptics	**0.95**	0.40	**0.95**	**0.95**	**0.95**	0.45	**0.95**	**0.95**

Table C.13: AoD from all algorithms on all datasets when the scaling factor (f) is 1.2 and number of clusters (k) is chosen by the SSTSC algorithms

Dataset	Accuracy on Detection (AoD)							
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	0.84	0.92	0.90	**0.93**	0.86	0.92	0.89	0.93
SonyAIBORobotSurfaceII	0.86	0.92	0.81	0.92	0.85	0.92	0.89	**0.93**
SonyAIBORobotSurface	0.81	0.88	0.82	0.88	0.75	0.89	0.88	**0.92**
DistalPhalanxOutlineCorrect	0.89	0.89	0.89	0.89	0.90	0.90	**0.93**	**0.93**
MiddlePhalanxOutlineCorrect	0.88	0.89	0.86	0.86	0.86	0.88	**0.94**	**0.94**
PhalangesOutlinesCorrect	0.92	0.92	0.94	**0.95**	0.93	0.93	0.93	0.93
ProximalPhalanxOutlineCorrect	0.87	0.87	0.86	0.86	0.87	0.87	**0.94**	**0.94**
DistalPhalanxOutlineAgeGroup	0.84	0.89	0.87	0.87	0.84	0.89	**0.94**	**0.94**
MiddlePhalanxOutlineAgeGroup	0.94	0.94	**0.94**	**0.94**	0.94	0.94	0.94	0.94
ProximalPhalanxOutlineAgeGroup	0.93	0.94	0.92	0.92	0.92	0.92	**0.94**	**0.94**
TwoLeadECG	0.89	0.90	0.91	0.91	0.89	0.90	**0.95**	**0.95**
MoteStrain	0.76	0.87	0.83	0.88	0.75	0.87	0.89	**0.92**
ECG200	0.83	0.88	0.85	0.87	0.83	0.90	0.91	**0.93**
CBF	0.67	0.88	0.86	0.91	0.64	0.85	0.91	**0.92**
Two_Patterns	0.72	0.86	0.74	0.85	0.69	0.87	0.81	**0.95**
ECGFiveDays	0.91	0.91	0.91	0.94	0.91	0.91	**0.96**	**0.96**
ECG5000	0.76	0.90	0.83	0.86	0.79	0.88	0.88	**0.91**
Gun_Point	0.88	0.90	0.88	0.90	0.88	0.90	**0.91**	**0.91**
wafer	0.79	0.87	0.83	0.87	0.80	0.87	0.90	**0.92**
ChlorineConcentration	0.85	0.94	0.88	0.91	0.85	0.93	**0.94**	**0.94**
Wine	0.86	0.86	0.88	0.88	0.87	0.87	**0.92**	**0.92**
Strawberry	0.90	0.90	0.91	0.91	0.90	0.90	**0.95**	**0.95**
ArrowHead	0.90	0.95	0.92	0.92	0.88	0.95	**0.98**	**0.98**
Trace	0.87	0.89	0.90	0.91	0.87	0.89	0.93	**0.93**
ToeSegmentation1	0.75	**0.88**	0.81	0.87	0.79	0.87	0.82	0.88
Coffee	0.91	0.91	0.90	0.90	0.92	0.92	**0.96**	**0.96**
ToeSegmentation2	0.73	0.85	0.82	0.90	0.77	0.87	0.79	**0.90**
FaceFour	0.70	**0.96**	0.83	0.89	0.71	0.89	0.85	0.90
yoga	0.82	0.89	0.91	**0.93**	0.81	0.89	0.88	0.90
Ham	0.72	0.88	0.78	0.89	0.72	0.84	0.73	**0.93**
Meat	0.91	0.91	0.92	0.92	0.92	0.92	**0.98**	**0.98**
Beef	0.90	0.92	0.87	0.93	0.91	0.92	0.91	**0.97**
FordA	0.62	**0.95**	0.72	0.86	0.64	0.86	0.65	0.92
FordB	0.63	0.82	0.70	0.88	0.67	**0.89**	0.69	0.88
ShapeletSim	0.69	**0.92**	0.67	0.89	0.67	0.87	0.60	0.00
BeetleFly	0.65	0.85	0.66	0.85	0.65	0.86	0.73	**0.91**
BirdChicken	0.78	0.88	0.82	0.87	0.78	0.88	0.87	**0.92**
Earthquakes	0.71	0.92	0.69	0.83	0.70	**0.96**	0.61	0.95
Herring	0.85	0.85	0.86	0.86	0.86	0.86	**0.92**	**0.92**
OliveOil	0.86	0.86	0.87	0.87	0.88	0.88	**0.94**	**0.94**
Car	0.84	0.85	0.87	0.87	0.84	0.85	**0.92**	**0.92**
Lighting2	0.73	0.92	0.64	0.87	0.70	**0.96**	0.74	0.91
Computers	0.74	0.87	0.71	0.86	0.78	0.88	0.79	**0.92**
LargeKitchenAppliances	0.72	0.87	0.78	0.86	0.72	0.87	0.83	**0.91**
RefrigerationDevices	0.70	0.95	0.73	0.92	0.68	**0.96**	0.69	0.91
ScreenType	0.64	0.84	0.69	0.84	0.61	0.84	0.67	**0.88**
SmallKitchenAppliances	0.69	0.85	0.76	**0.86**	0.69	0.85	0.79	0.85
WormsTwoClass	0.62	**0.98**	0.68	0.83	0.66	0.97	0.71	0.89
Worms	0.66	0.84	0.66	0.88	0.66	0.84	0.72	**0.90**
StarLightCurves	0.88	0.89	0.88	0.90	0.87	0.88	0.88	**0.90**
Haptics	0.78	**0.92**	0.87	0.87	0.79	0.92	0.87	0.87

Table C.14: Excess from all algorithms on all datasets when the scaling factor (f) is 1.2 and number of clusters (k) is chosen by the SSTSC algorithms

Dataset	Excess Rate							
	E-AA-Z		E-AA-L		E-SA-Z		E-SA-L	
	40%	80%	40%	80%	40%	80%	40%	80%
ItalyPowerDemand	**0.00**	0.45	**0.00**	0.20	**0.00**	0.40	0.05	0.24
SonyAIBORobotSurfaceII	0.23	0.50	**0.09**	0.59	0.24	0.48	**0.09**	0.27
SonyAIBORobotSurface	0.27	0.45	0.16	0.42	0.41	0.64	**0.11**	0.28
DistalPhalanxOutlineCorrect	**0.00**	**0.00**	**0.00**	**0.00**	0.05	0.05	**0.00**	**0.00**
MiddlePhalanxOutlineCorrect	0.05	0.14	0.05	0.05	0.05	0.19	**0.00**	**0.00**
PhalangesOutlinesCorrect	**0.00**	**0.00**	**0.00**	0.05	**0.00**	**0.00**	**0.00**	**0.00**
ProximalPhalanxOutlineCorrect	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
DistalPhalanxOutlineAgeGroup	0.05	0.27	**0.00**	**0.00**	0.05	0.32	**0.00**	**0.00**
MiddlePhalanxOutlineAgeGroup	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
ProximalPhalanxOutlineAgeGroup	**0.00**	0.05	**0.00**	**0.00**	**0.00**	0.05	**0.00**	**0.00**
TwoLeadECG	**0.00**	0.10	**0.00**	0.05	**0.00**	0.10	**0.00**	**0.00**
MoteStrain	0.28	0.56	**0.00**	0.30	0.35	0.57	0.05	0.20
ECG200	**0.00**	0.25	**0.00**	0.15	**0.00**	0.25	**0.00**	0.10
CBF	0.33	0.81	0.16	0.48	0.41	0.86	**0.00**	0.05
Two_Patterns	0.37	0.74	**0.09**	0.59	0.35	0.77	0.20	0.40
ECGFiveDays	**0.00**	**0.00**	0.10	0.19	**0.00**	**0.00**	**0.00**	**0.00**
ECG5000	0.23	0.55	**0.00**	0.17	0.19	0.48	**0.00**	0.17
Gun_Point	0.09	0.18	**0.00**	0.10	0.09	0.18	**0.00**	**0.00**
wafer	0.25	0.50	**0.00**	0.30	0.22	0.48	**0.00**	0.15
ChlorineConcentration	**0.00**	0.45	**0.00**	0.19	**0.00**	0.43	**0.00**	**0.00**
Wine	**0.00**	**0.00**	0.09	0.09	0.09	0.09	**0.00**	**0.00**
Strawberry	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
ArrowHead	0.05	0.18	**0.00**	**0.00**	0.13	0.29	**0.00**	**0.00**
Trace	**0.00**	0.20	**0.00**	0.10	**0.00**	0.15	**0.00**	0.05
ToeSegmentation1	0.31	0.62	0.13	0.43	0.27	0.54	**0.09**	0.36
Coffee	**0.00**	**0.00**	**0.00**	**0.00**	0.09	0.09	**0.00**	**0.00**
ToeSegmentation2	0.35	0.62	**0.13**	0.39	0.40	0.60	0.14	0.50
FaceFour	0.22	0.87	**0.00**	0.33	0.27	0.77	**0.00**	0.33
yoga	0.21	0.33	**0.00**	0.10	0.17	0.33	0.10	0.10
Ham	**0.25**	0.79	**0.25**	0.67	**0.25**	0.75	0.35	0.61
Meat	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Beef	**0.00**	0.15	**0.00**	0.15	**0.00**	0.11	0.05	0.19
FordA	0.68	1.00	**0.28**	0.72	0.35	0.88	0.32	0.95
FordB	0.56	0.92	**0.29**	0.75	0.52	0.81	0.40	0.75
ShapeletSim	0.64	0.84	0.69	0.88	0.68	0.91	**0.57**	1.00
BeetleFly	0.50	0.81	0.32	0.72	0.50	0.79	**0.13**	0.61
BirdChicken	0.10	0.62	**0.00**	0.40	0.14	0.64	0.05	0.20
Earthquakes	0.81	1.00	**0.54**	0.75	0.71	0.79	0.64	0.91
Herring	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
OliveOil	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Car	**0.00**	0.10	**0.00**	**0.00**	**0.00**	0.10	**0.00**	**0.00**
Lighting2	0.30	0.81	0.23	0.95	0.25	0.83	**0.05**	0.70
Computers	0.24	0.60	0.33	0.67	**0.17**	0.52	0.19	0.48
LargeKitchenAppliances	0.35	0.71	0.23	0.50	0.40	0.63	**0.05**	0.35
RefrigerationDevices	0.43	1.00	**0.17**	0.74	0.46	0.88	0.31	0.81
ScreenType	0.68	0.91	0.42	0.71	0.74	0.91	**0.39**	0.78
SmallKitchenAppliances	0.40	0.77	0.20	0.55	0.37	0.77	**0.14**	0.38
WormsTwoClass	0.62	0.95	0.46	0.88	0.46	0.88	**0.27**	0.73
Worms	0.42	0.88	0.38	0.85	0.38	0.85	**0.38**	0.83
StarLightCurves	**0.00**	0.10	**0.00**	0.10	**0.00**	0.10	**0.00**	0.10
Haptics	**0.00**	0.58	**0.00**	**0.00**	**0.00**	0.53	**0.00**	**0.00**

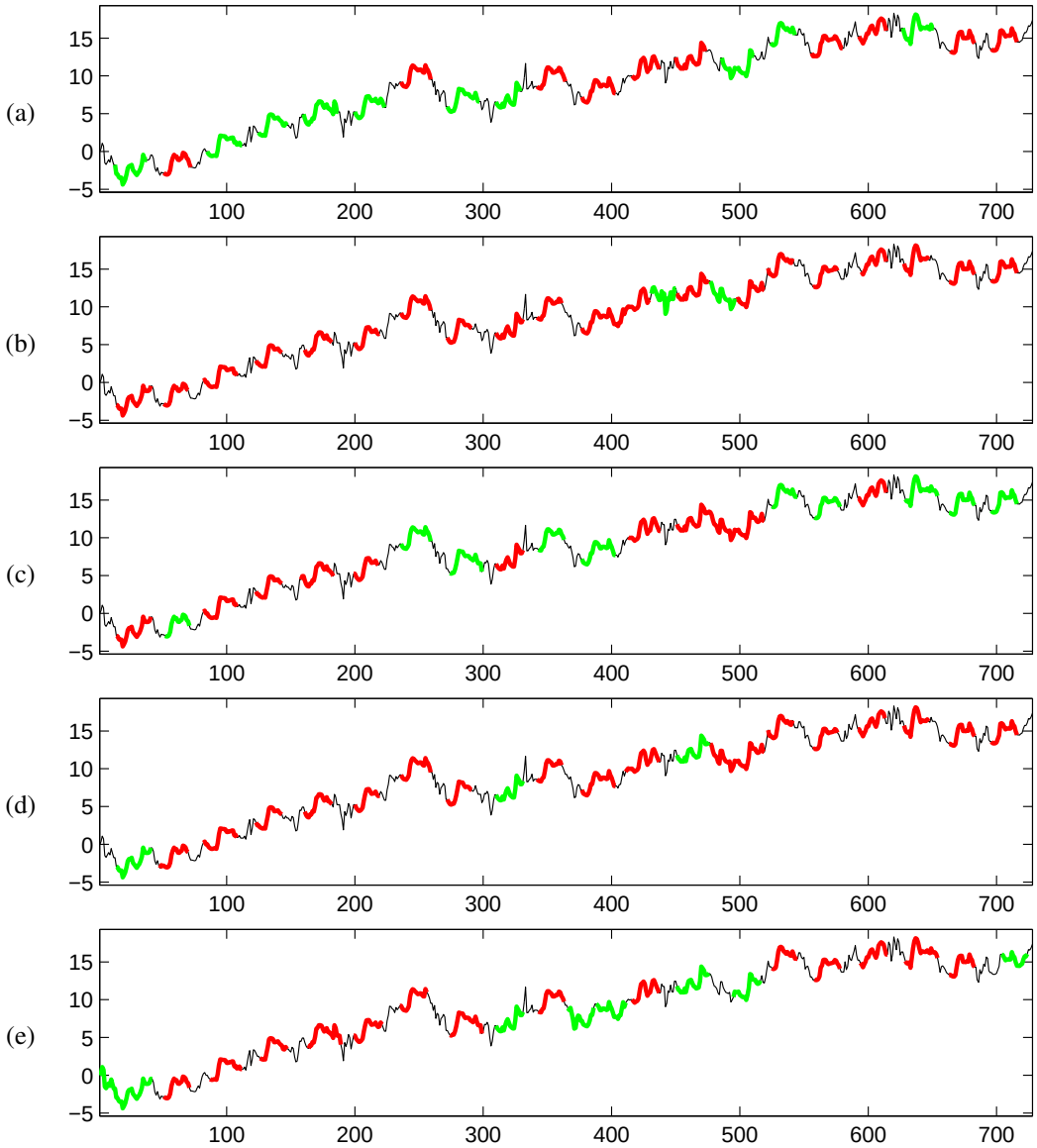

Figure C.1: ItalyPowerDemand dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

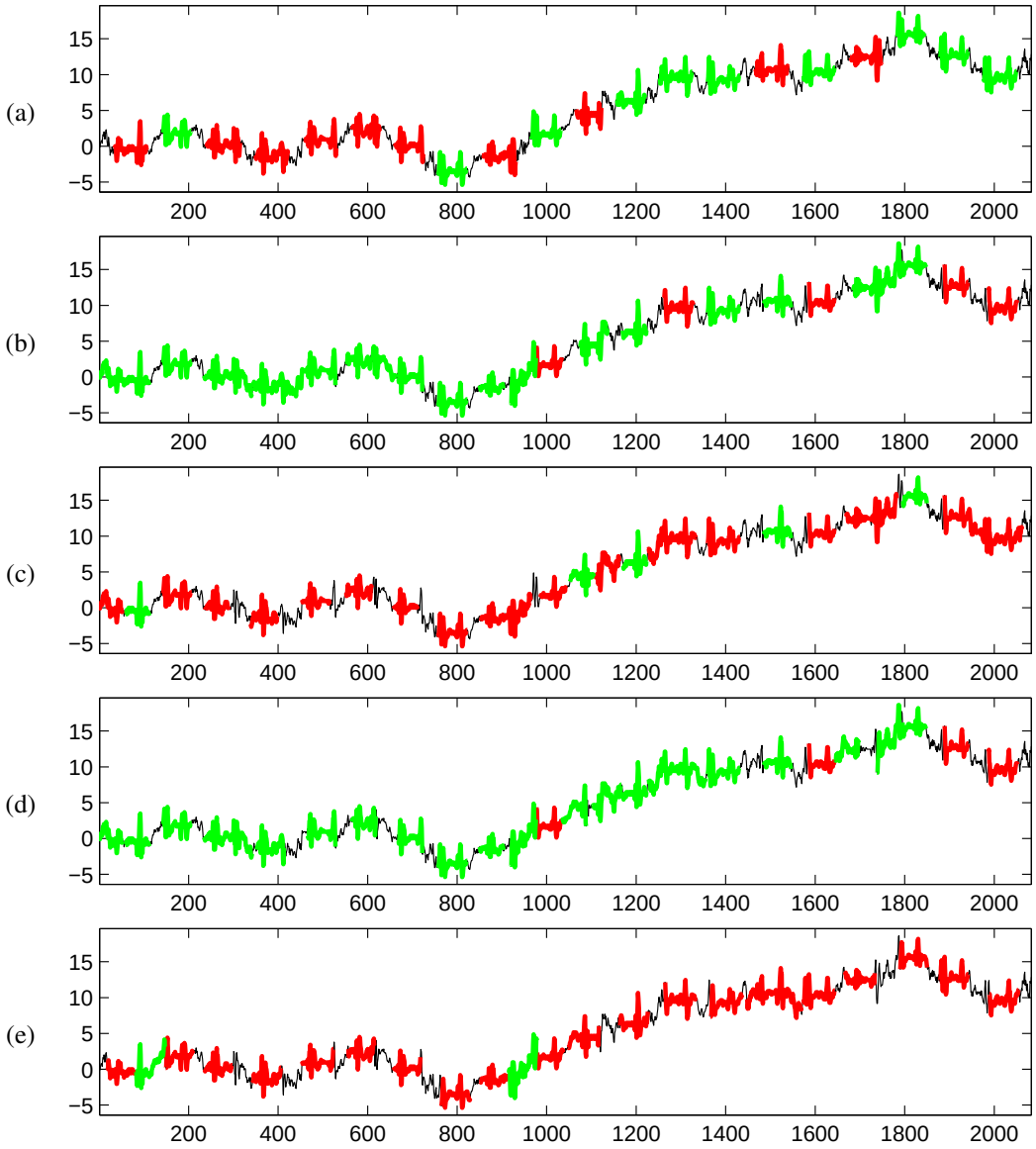

Figure C.2: SonyAIBORobotSurfaceII dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

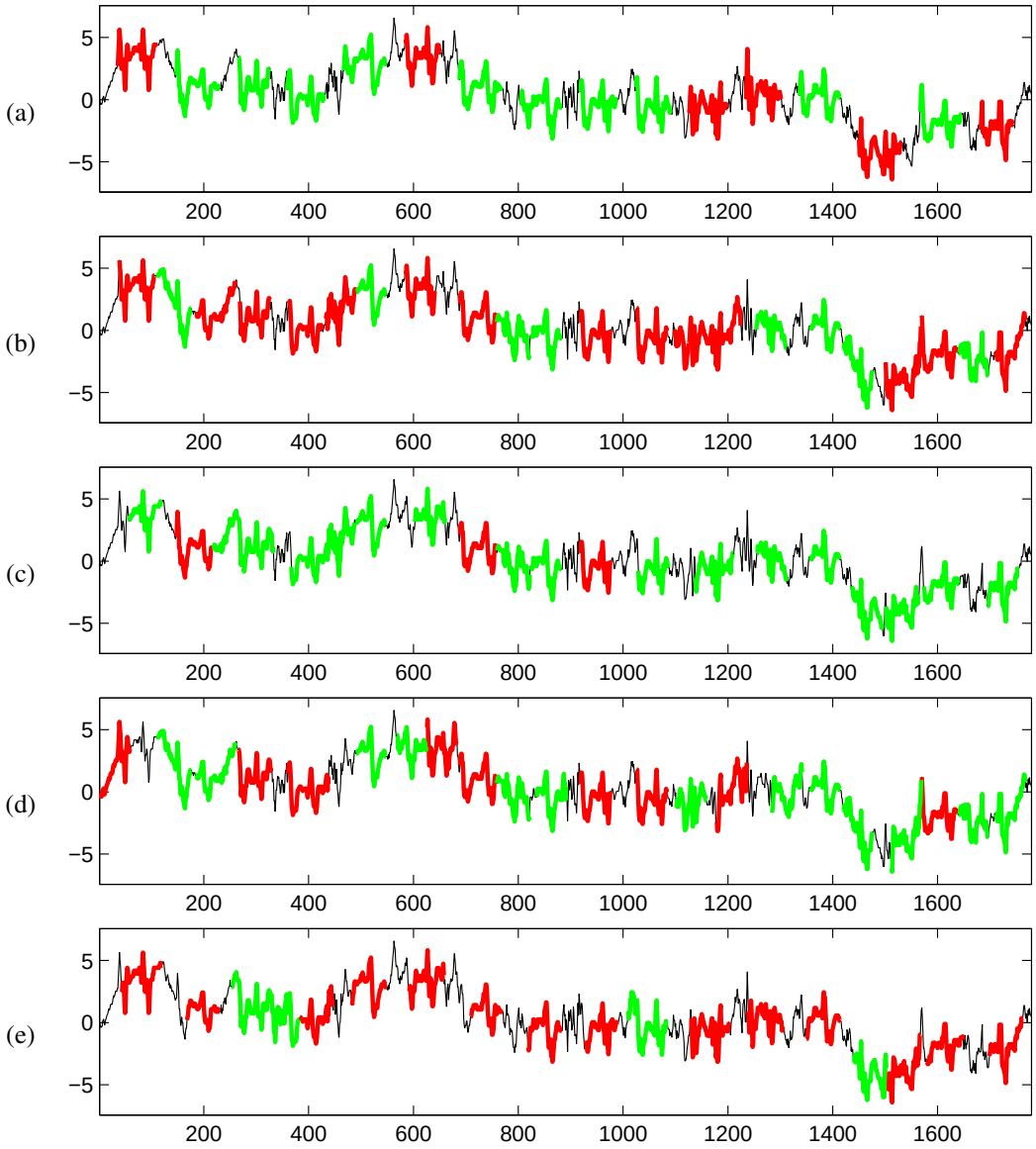

Figure C.3: SonyAIBORobotSurface dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

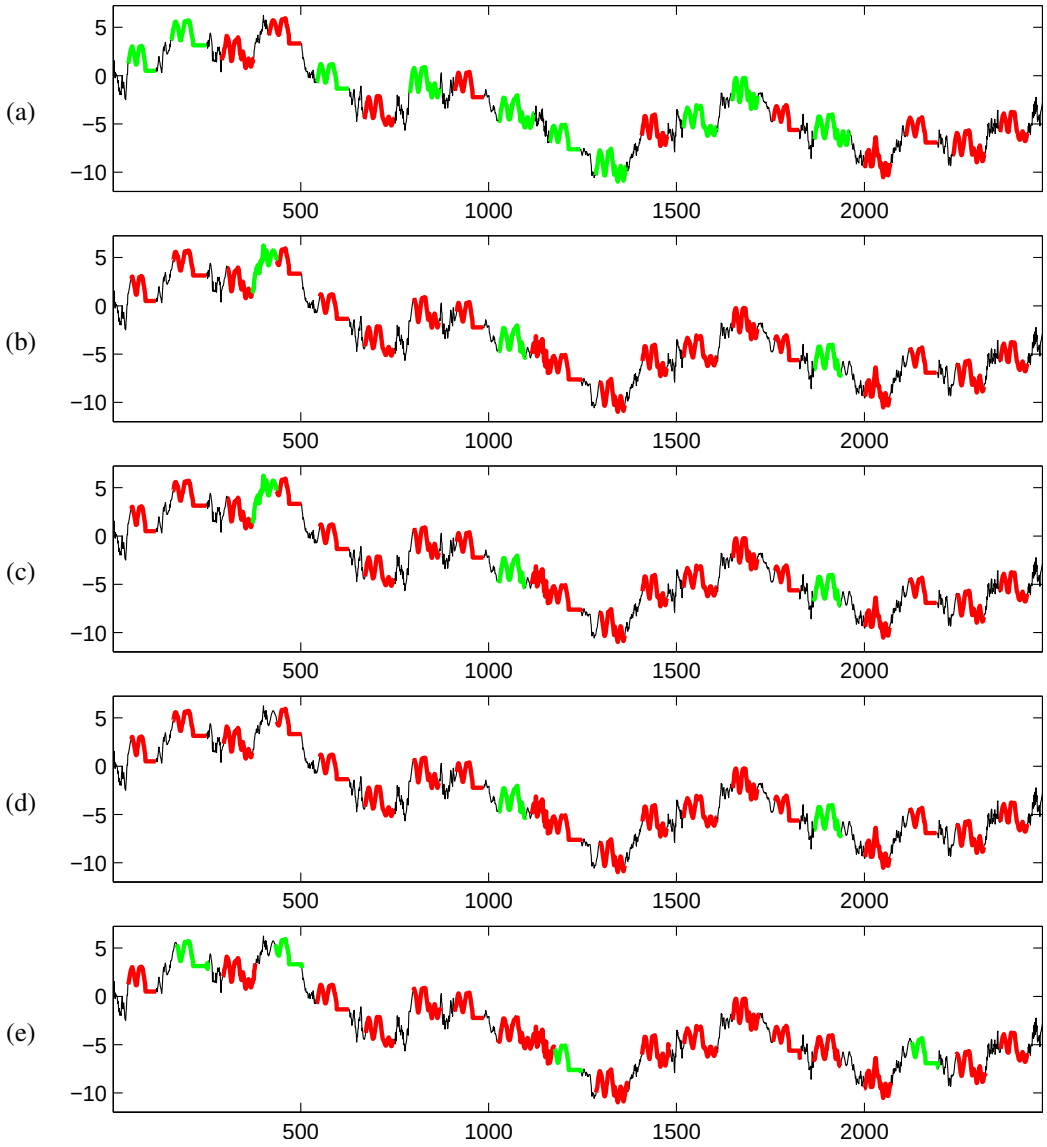

Figure C.4: DistalPhalanxOutlineCorrect dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

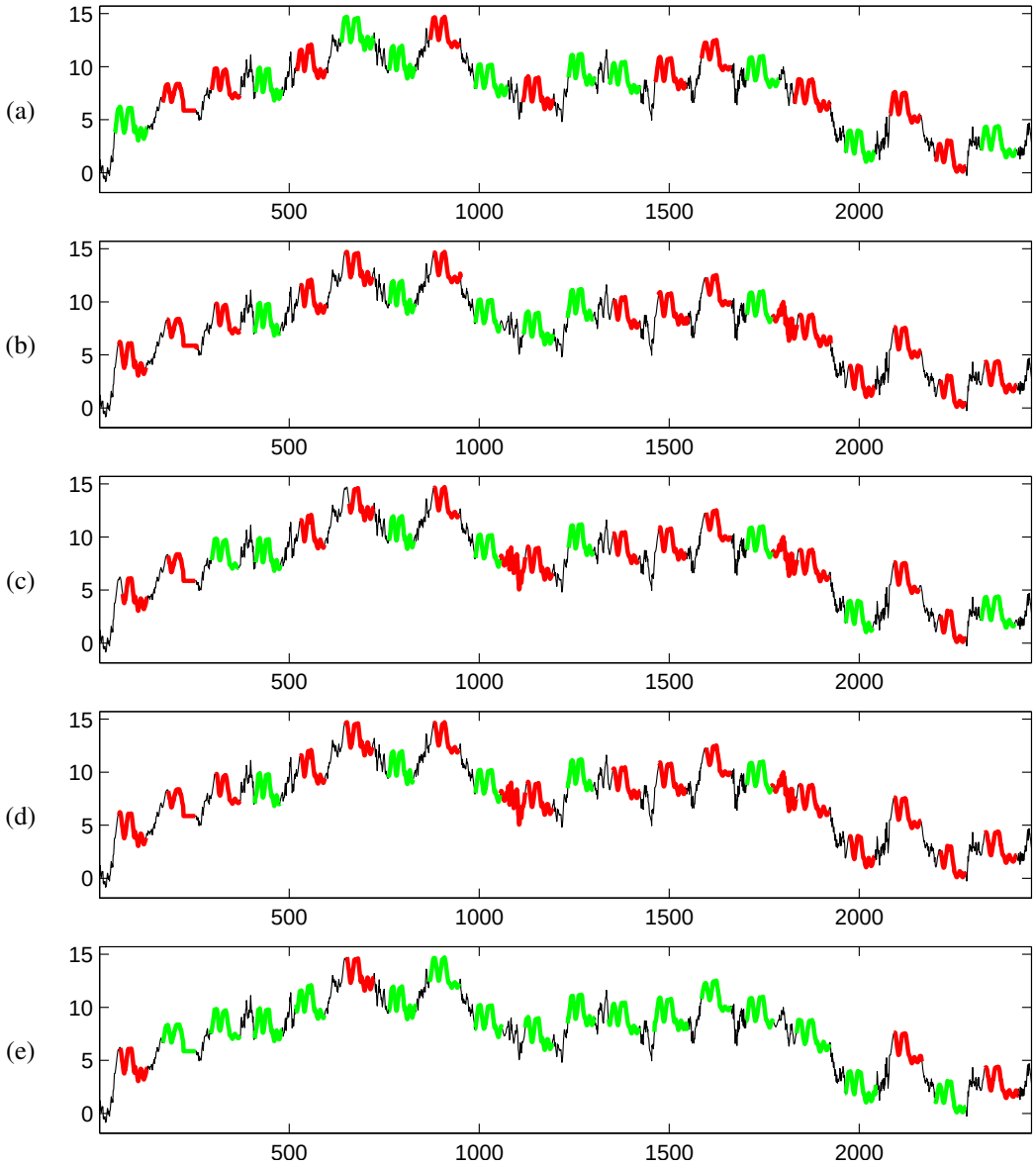

Figure C.5: MiddlePhalanxOutlineCorrect dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

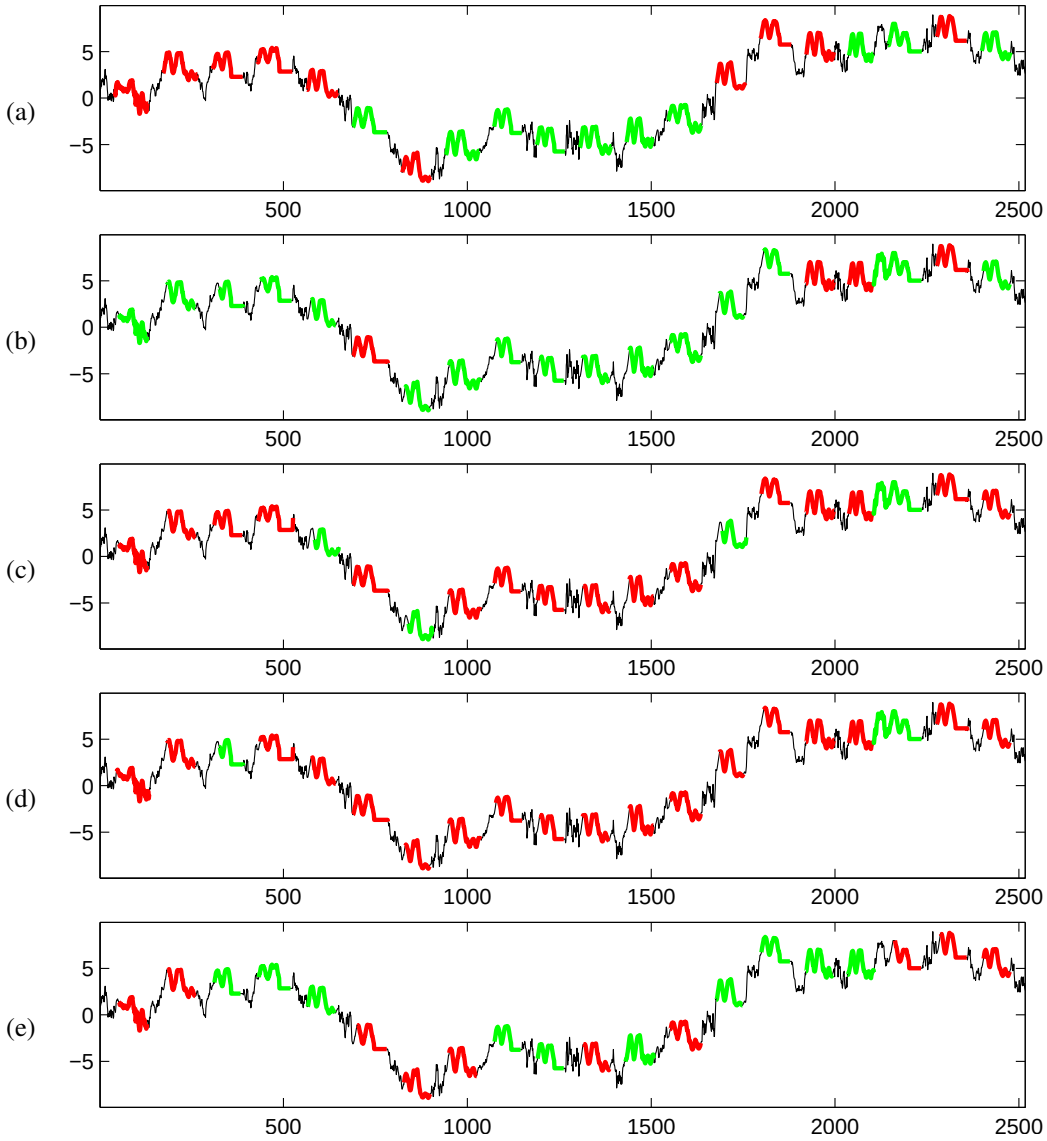

Figure C.6: PhalangesOutlinesCorrect dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

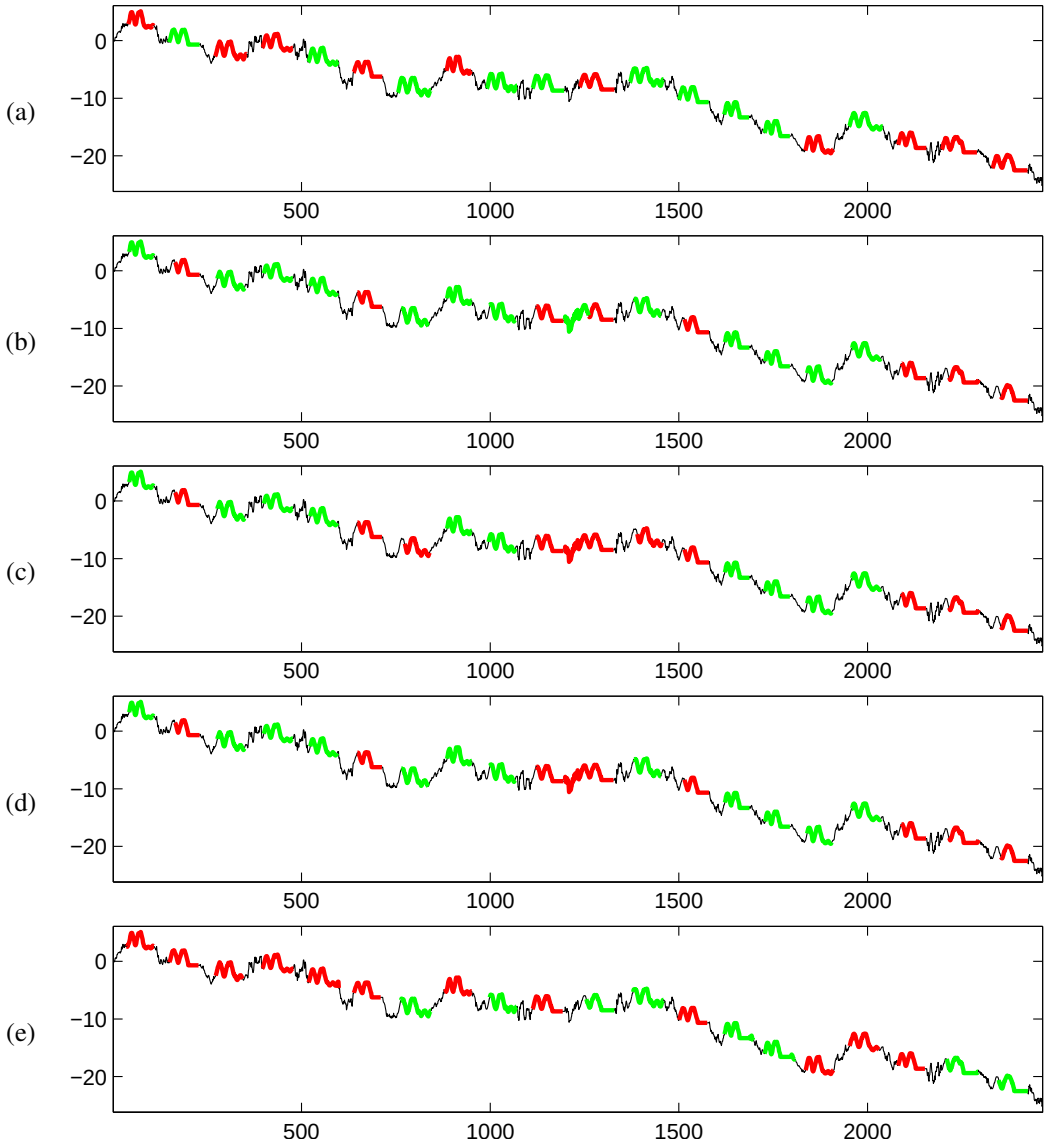

Figure C.7: ProximalPhalanxOutlineCorrect dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

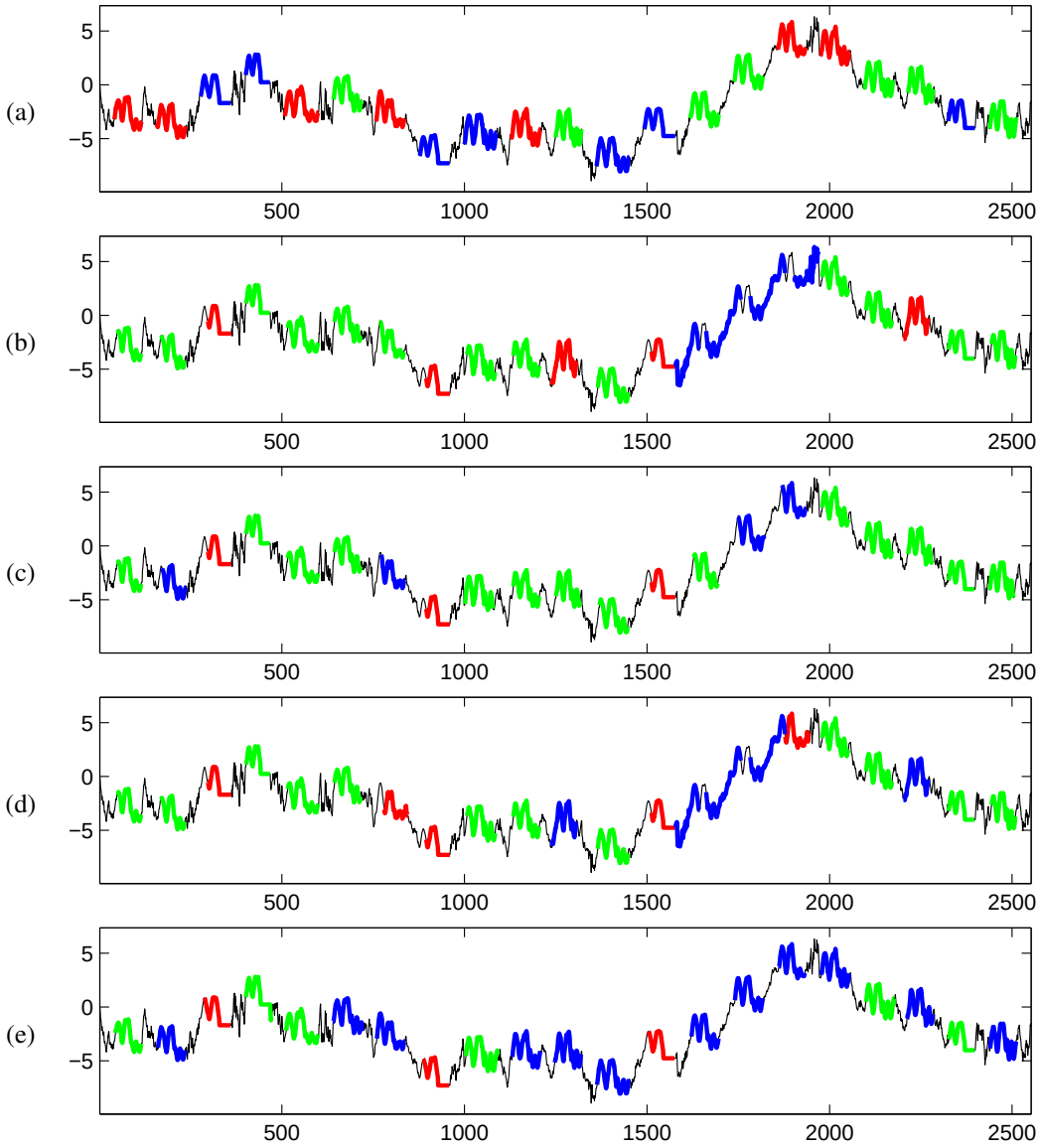

Figure C.8: DistalPhalanxOutlineAgeGroup dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

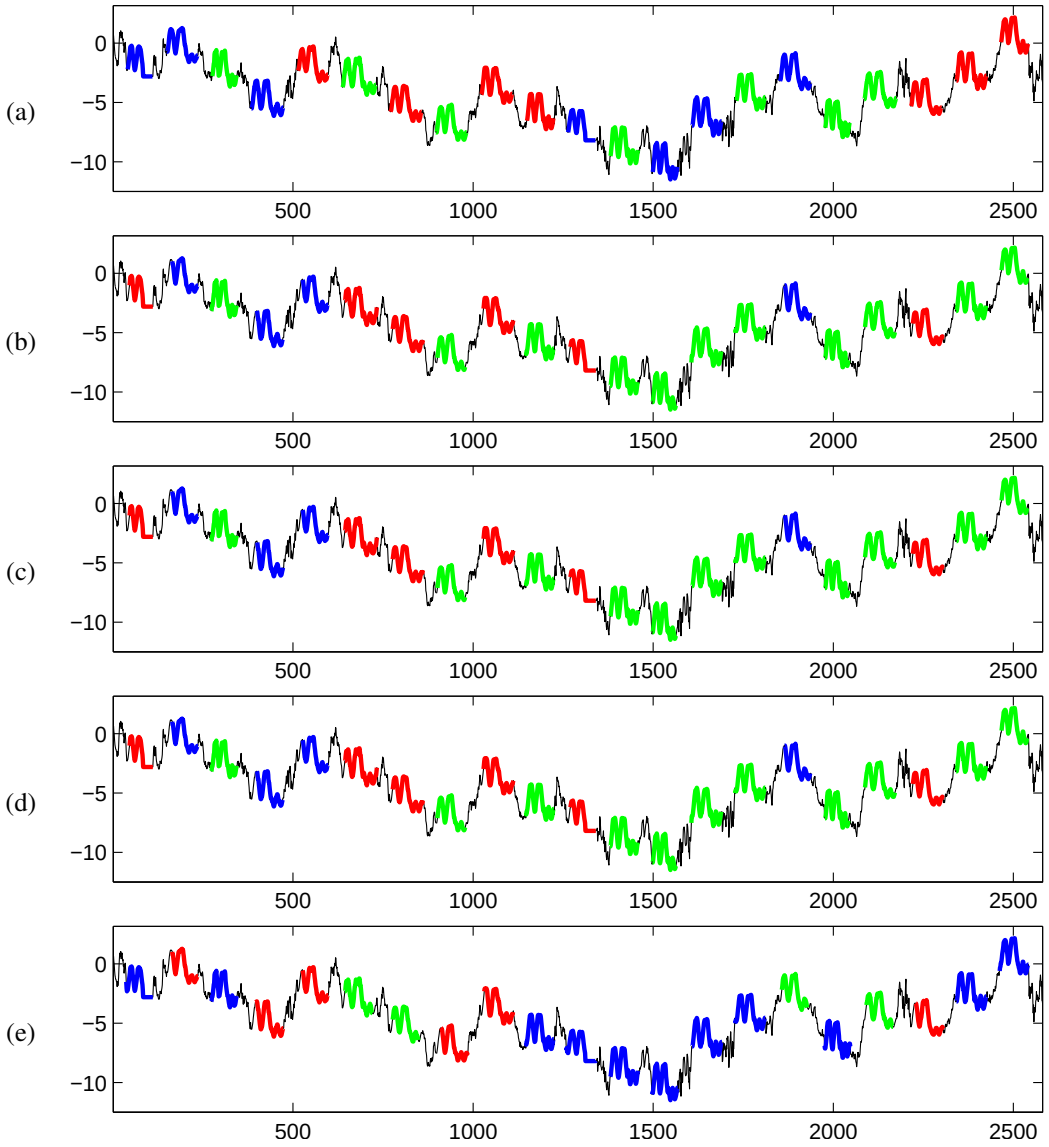

Figure C.9: MiddlePhalanxOutlineAgeGroup dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

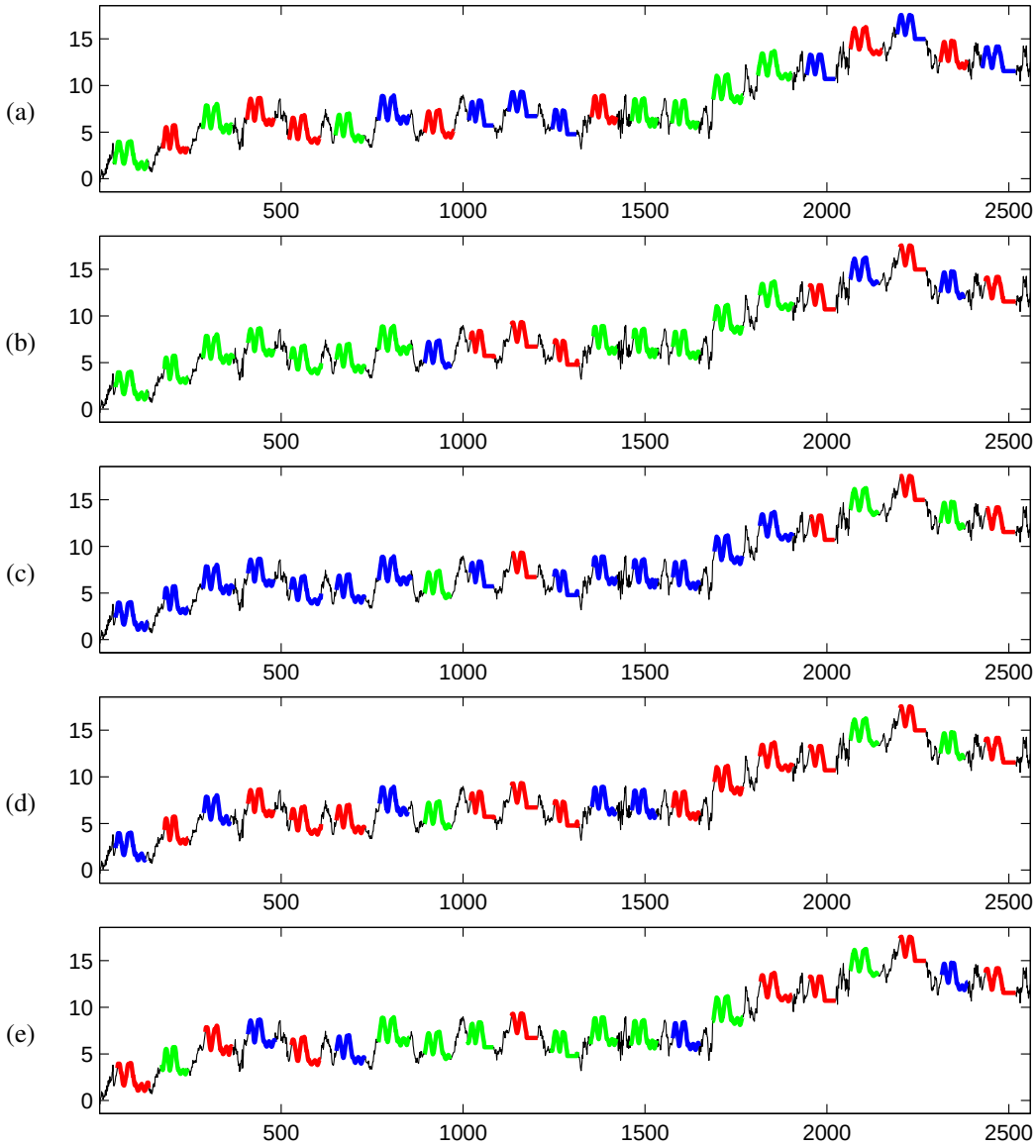

Figure C.10: ProximalPhalanxOutlineAgeGroup dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

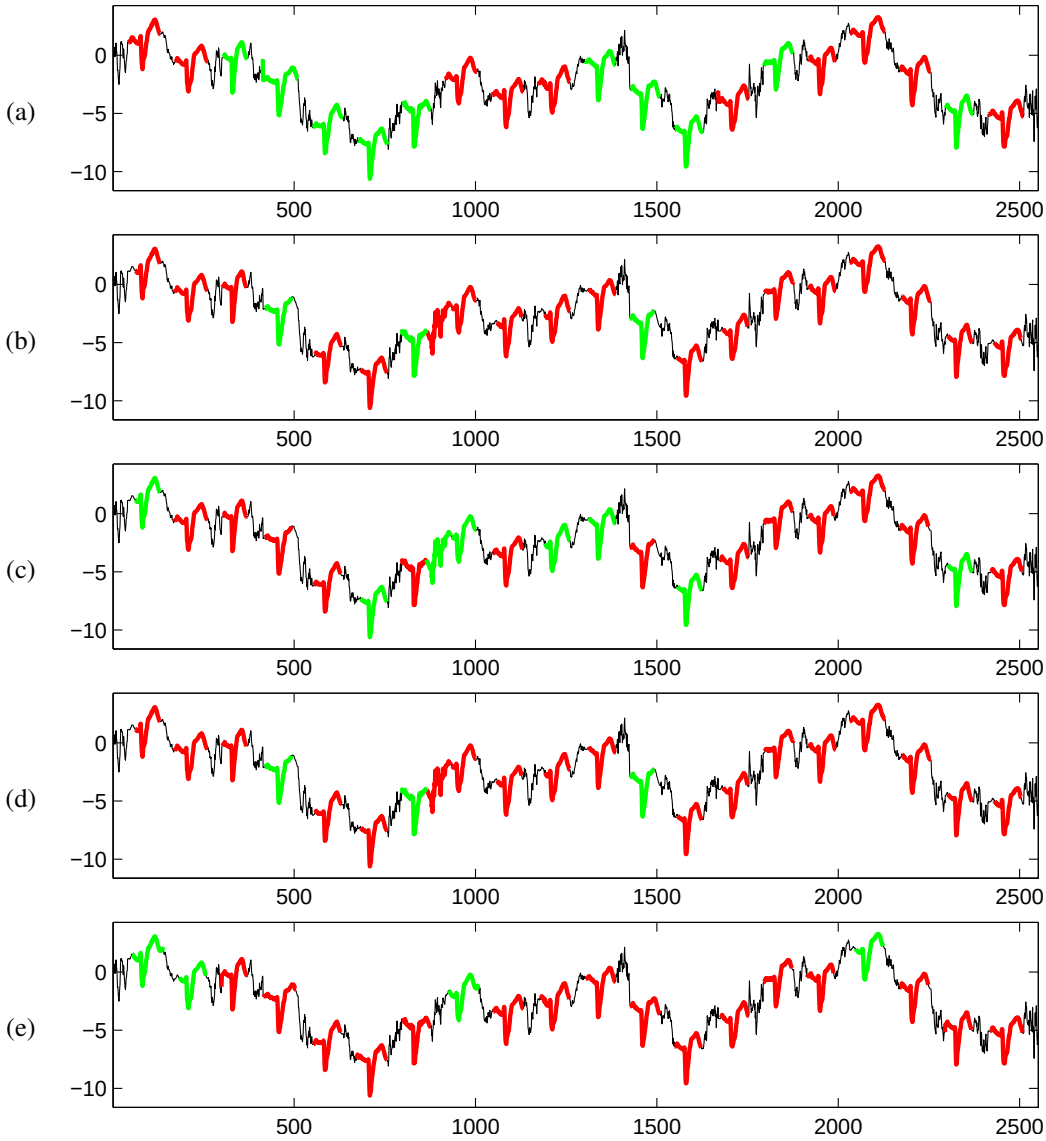

Figure C.11: TwoLeadECG dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

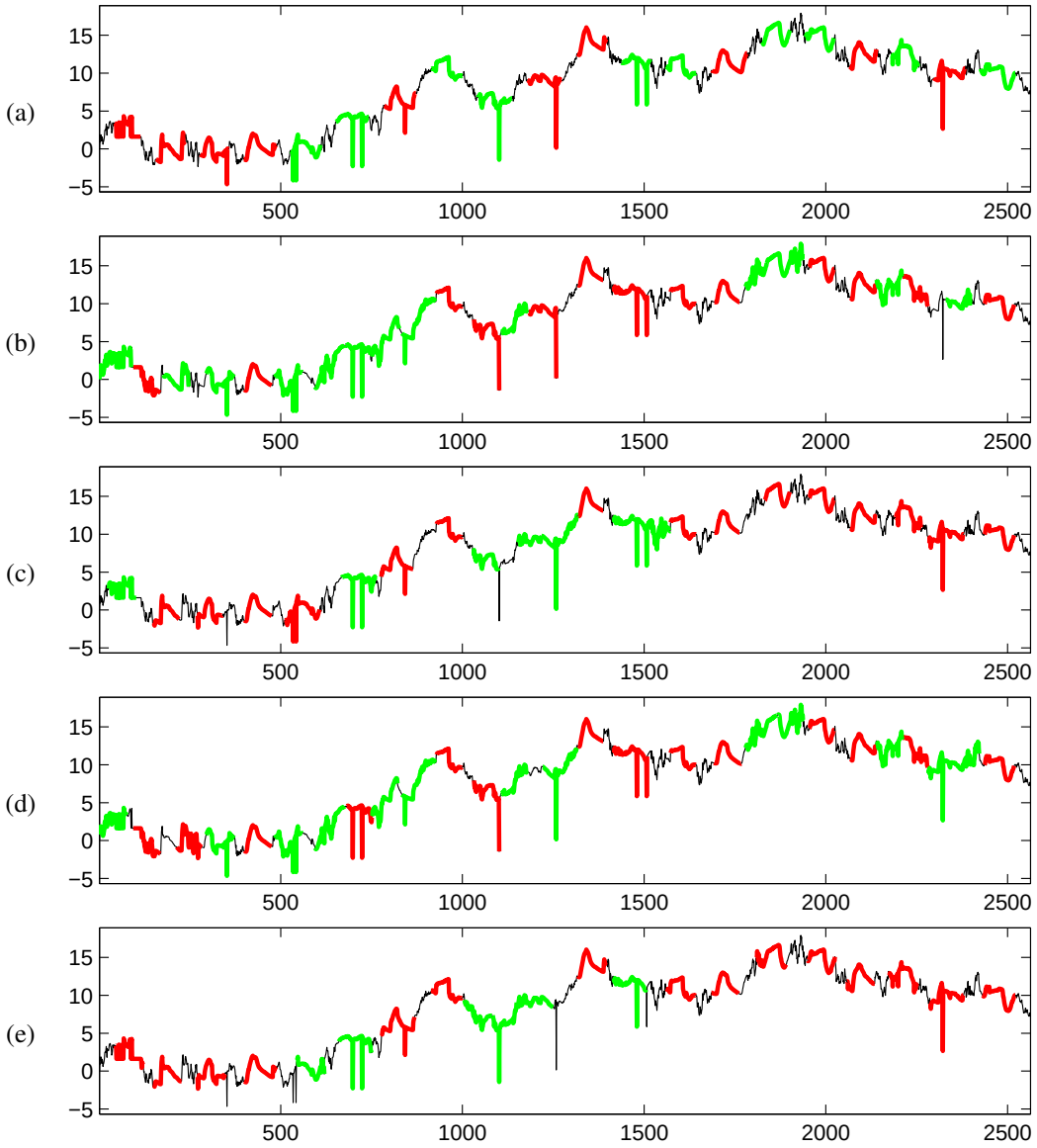

Figure C.12: MoteStrain dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

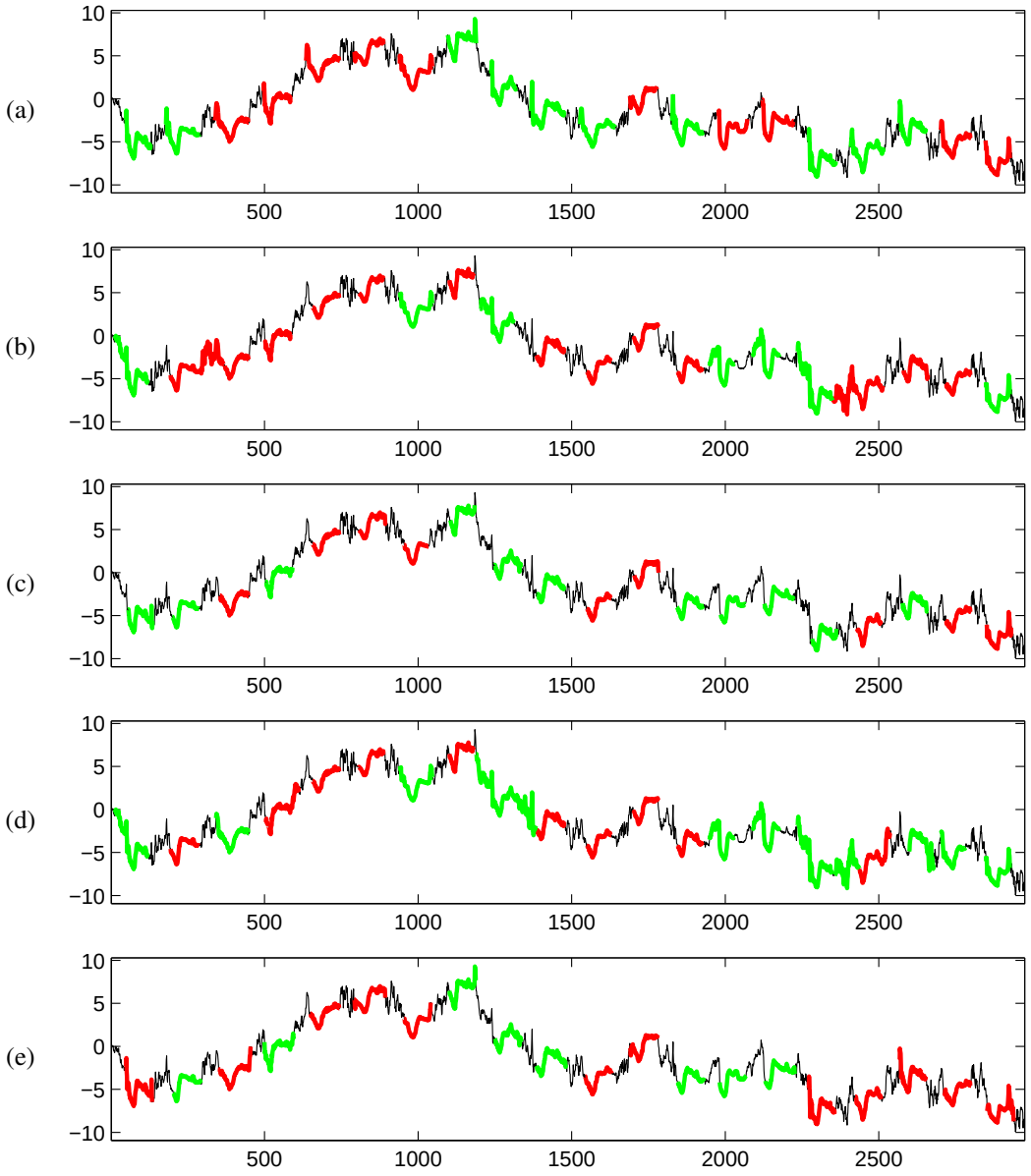

Figure C.13: ECG200 dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

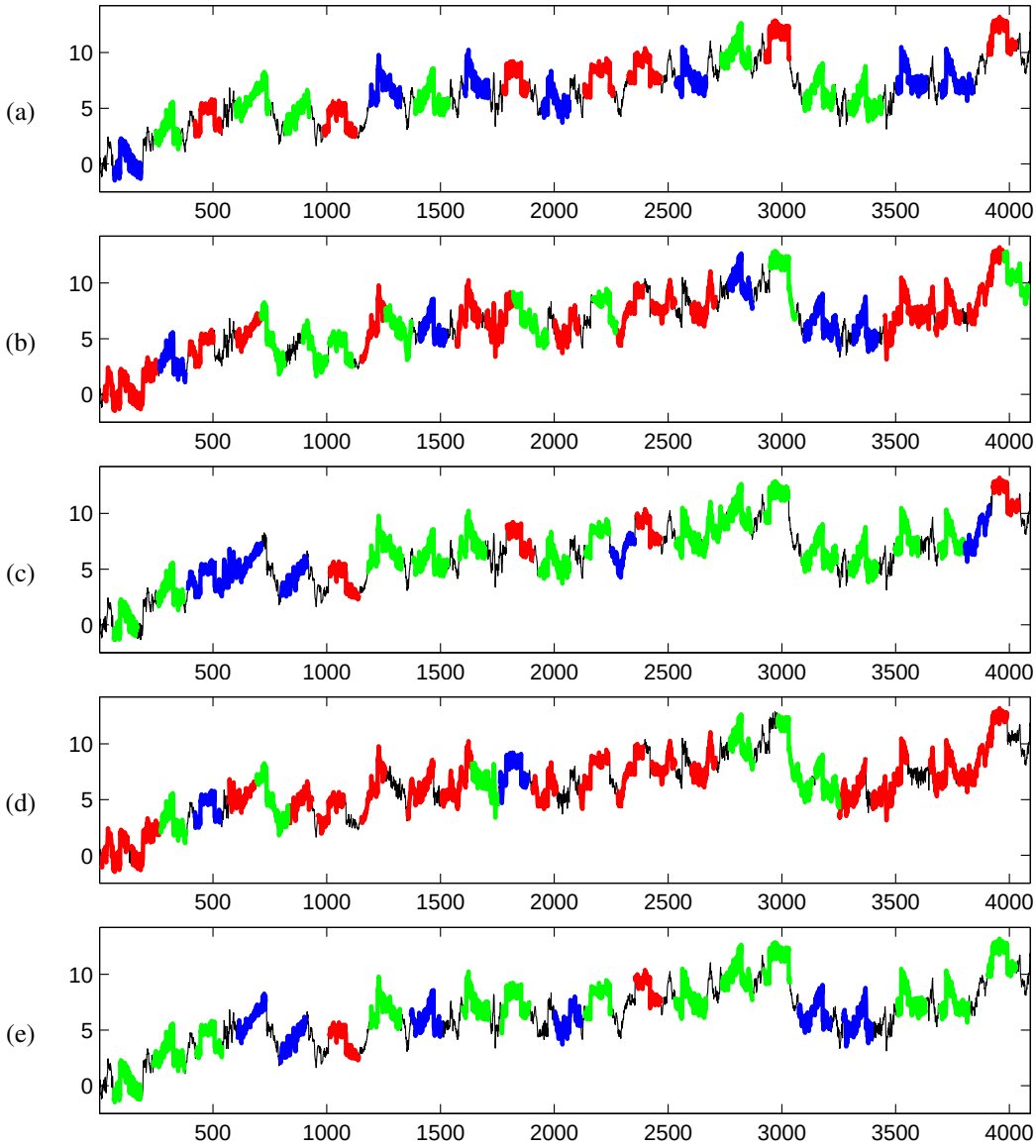

Figure C.14: CBF dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

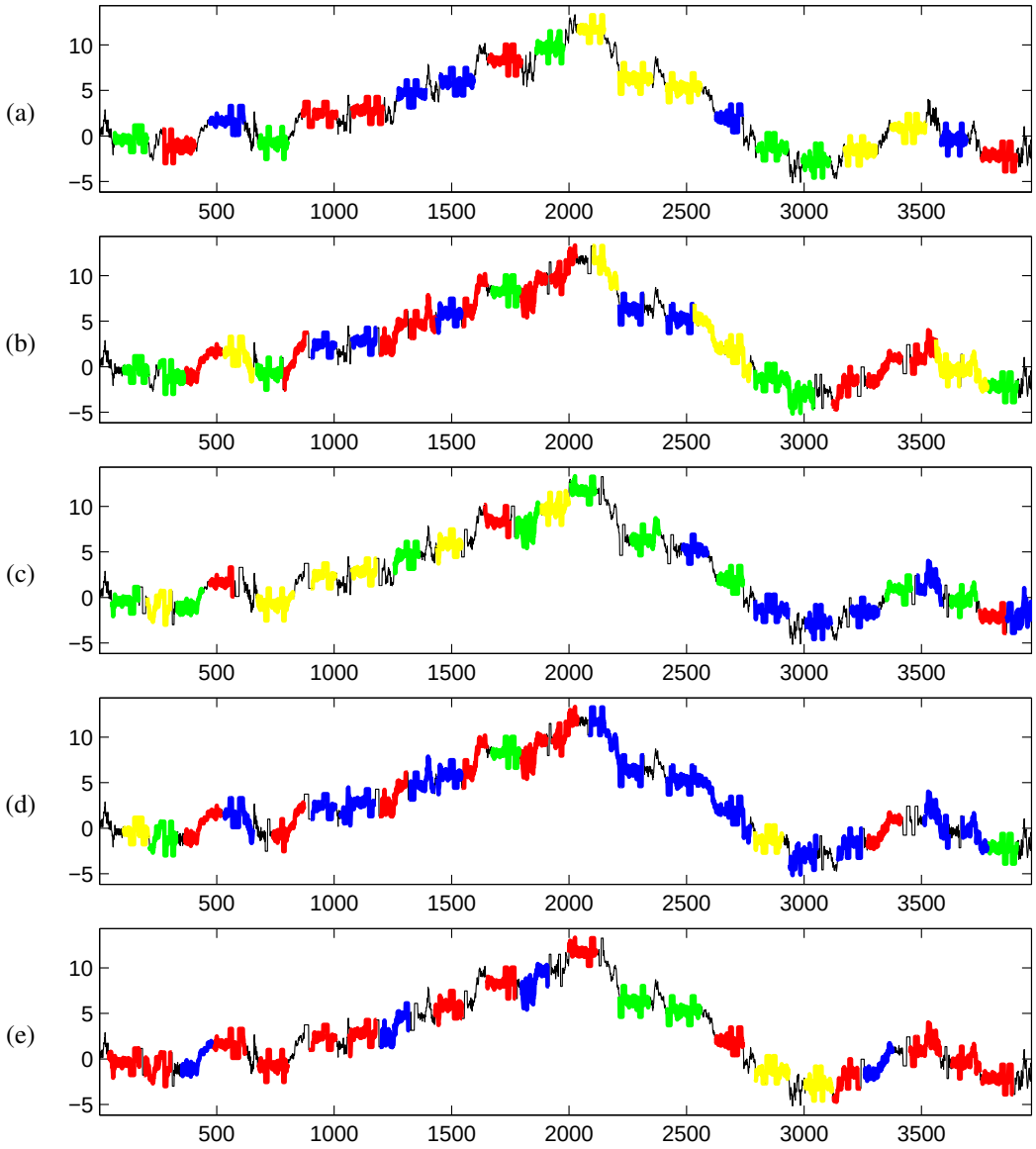

Figure C.15: Two_Patterns dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

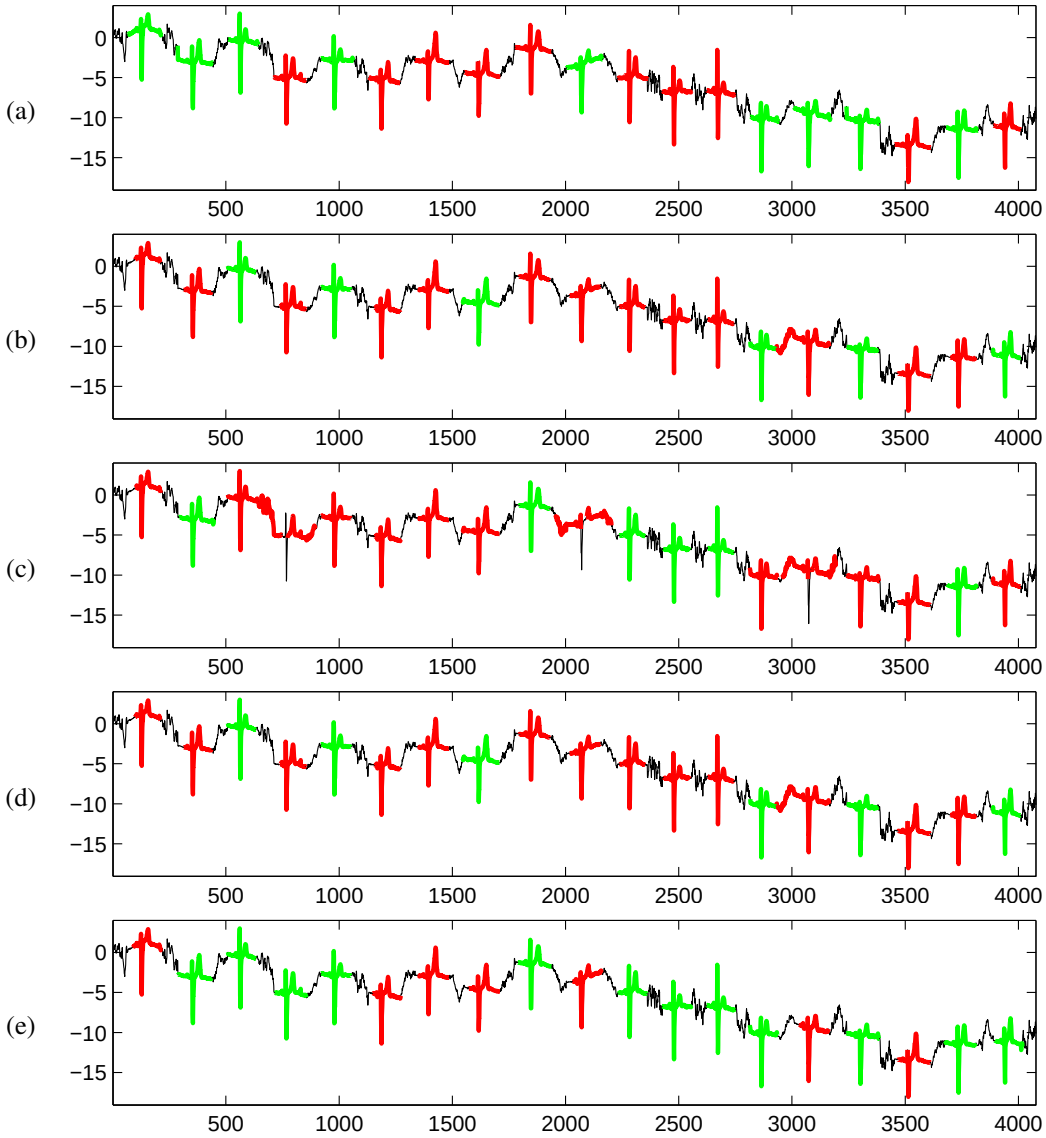

Figure C.16: ECGFiveDays dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

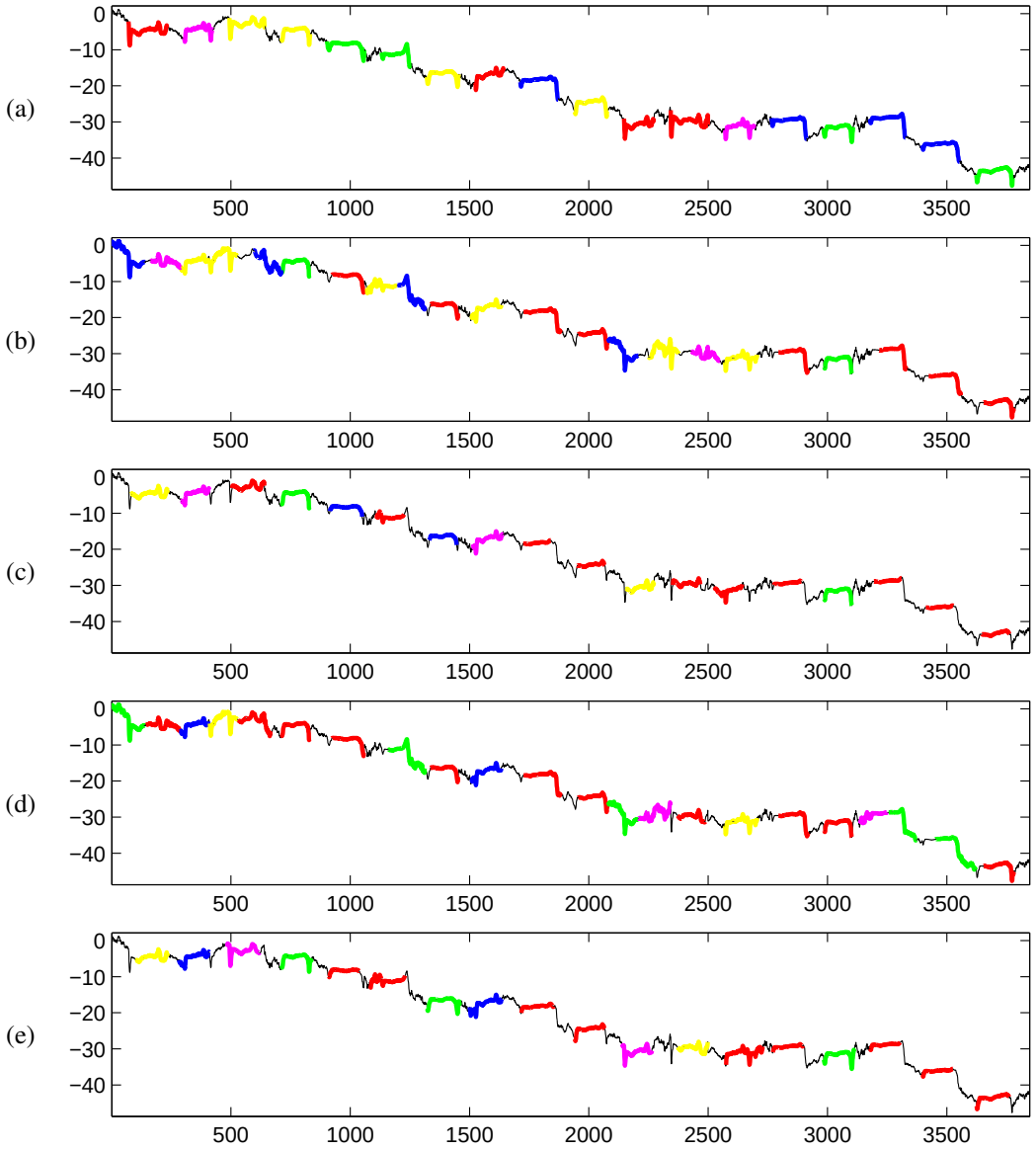

Figure C.17: ECG5000 dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

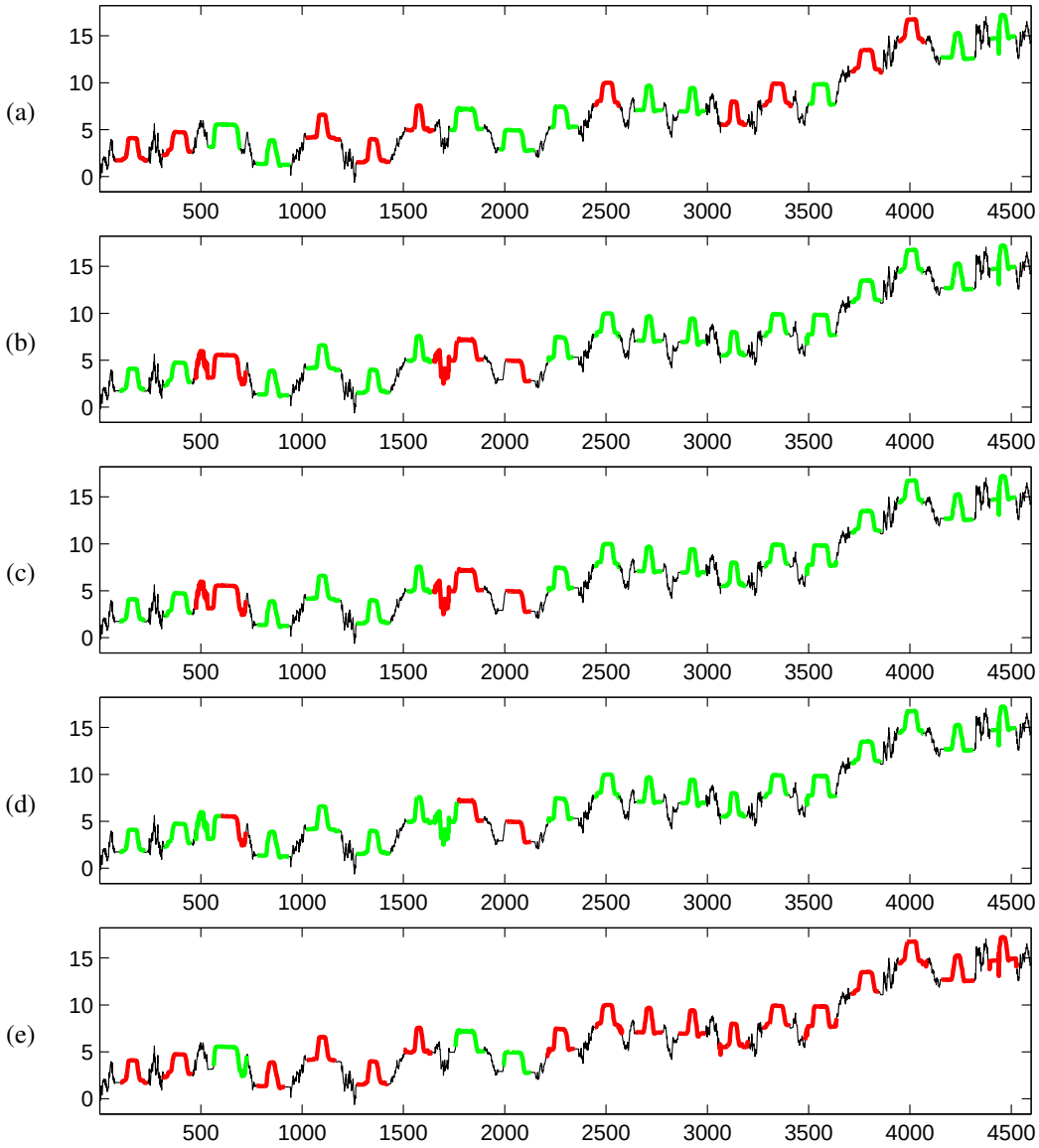

Figure C.18: Gun_Point dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

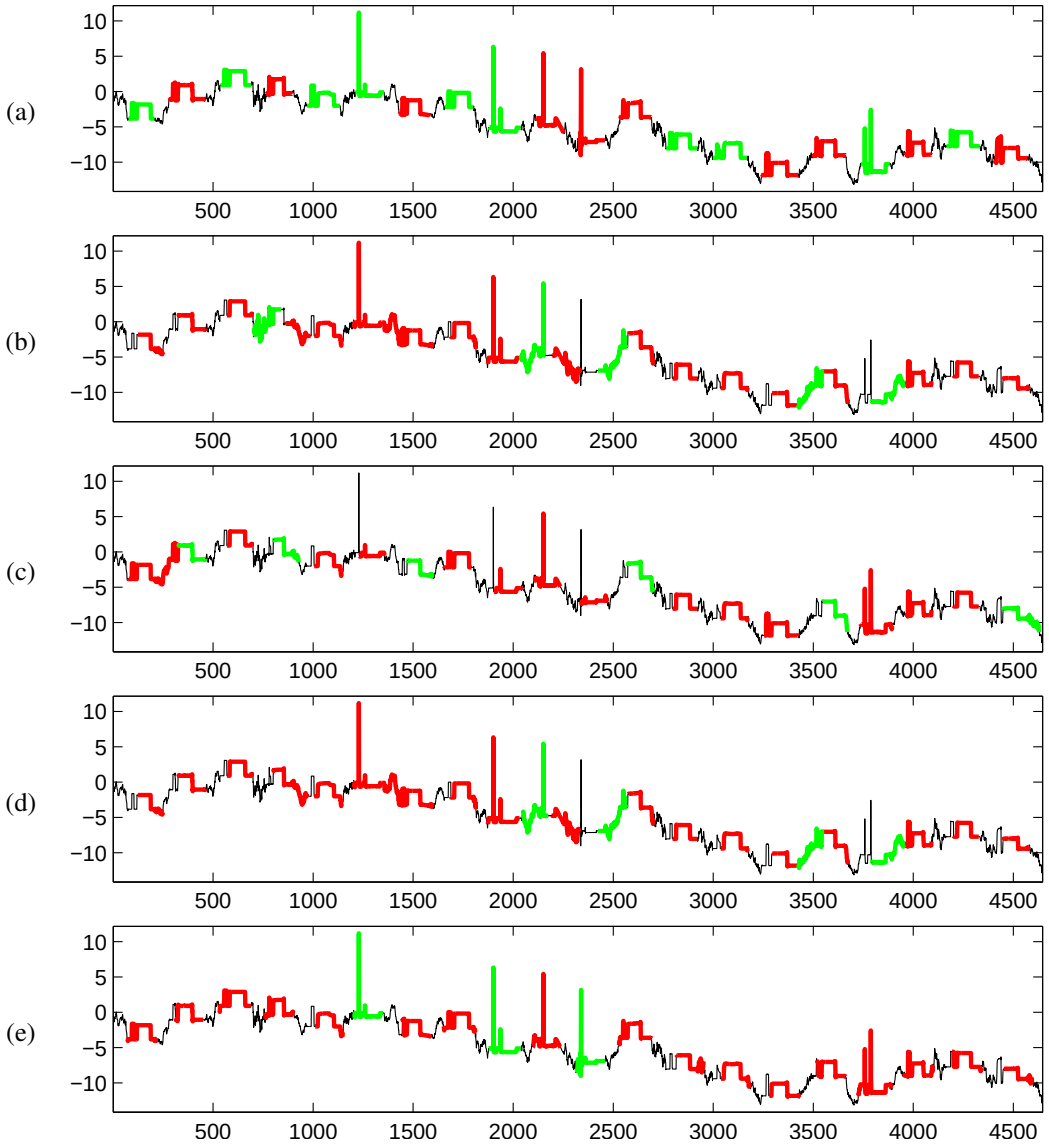

Figure C.19: wafer dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

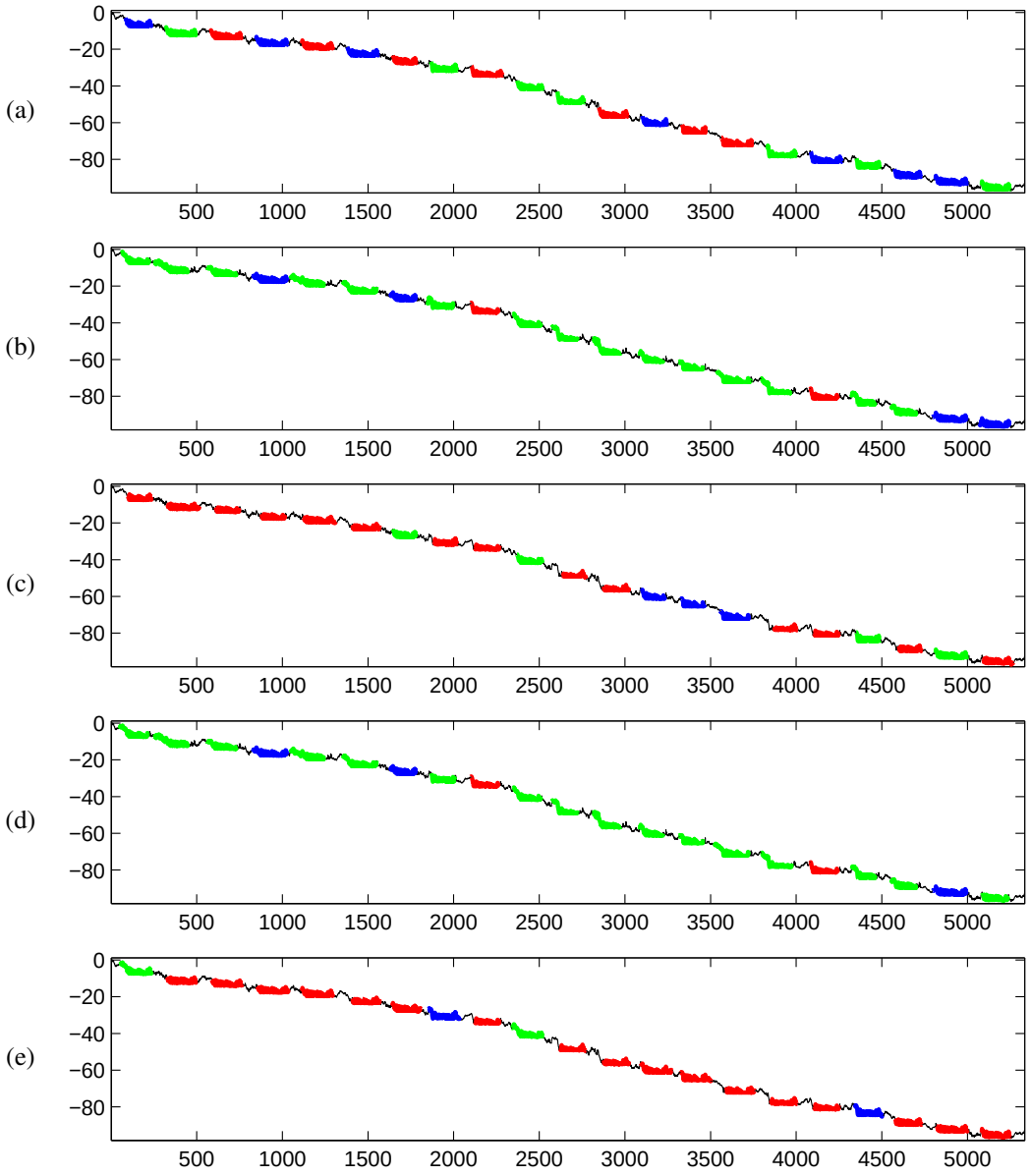

Figure C.20: ChlorineConcentration dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

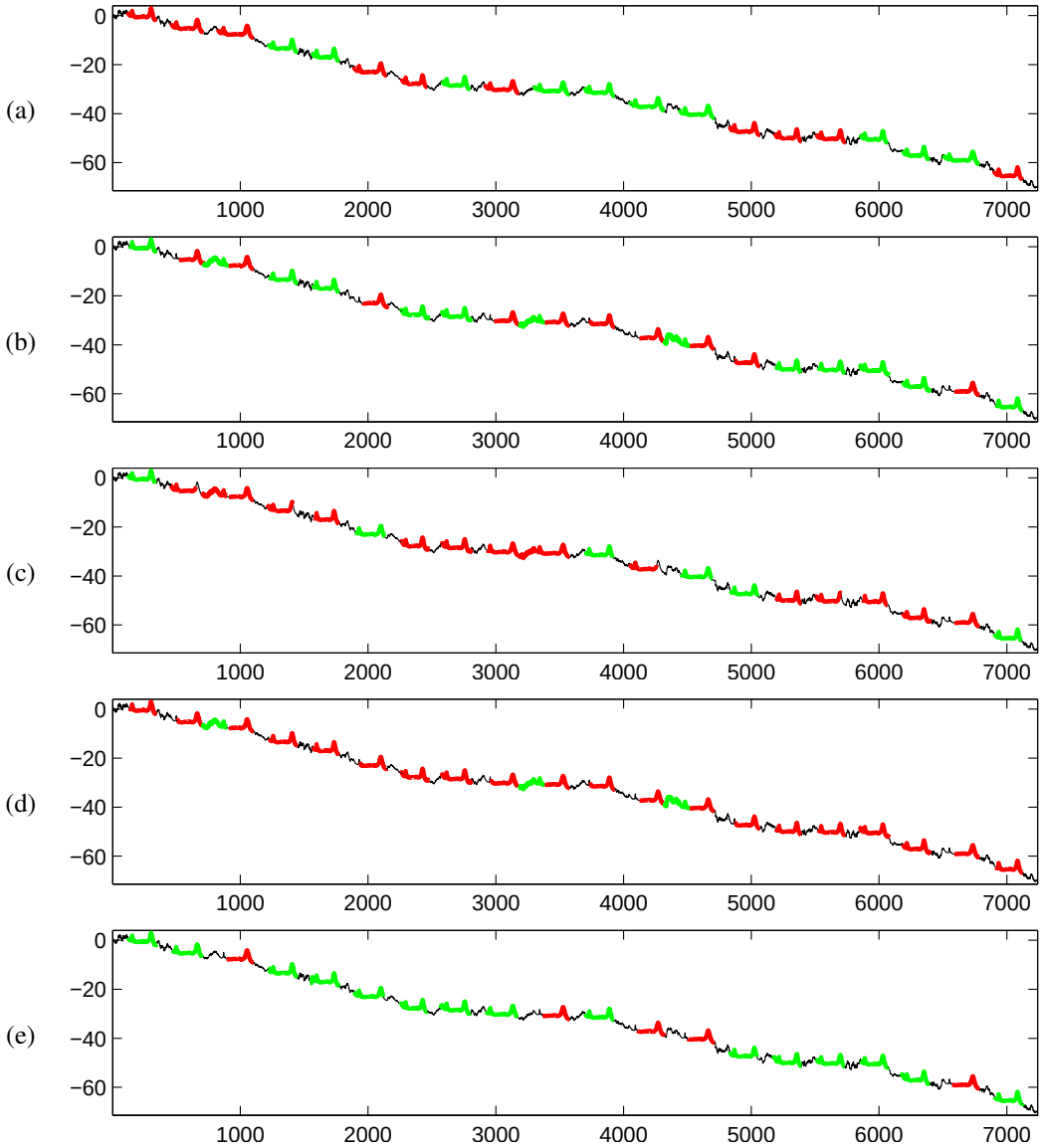

Figure C.21: Wine dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

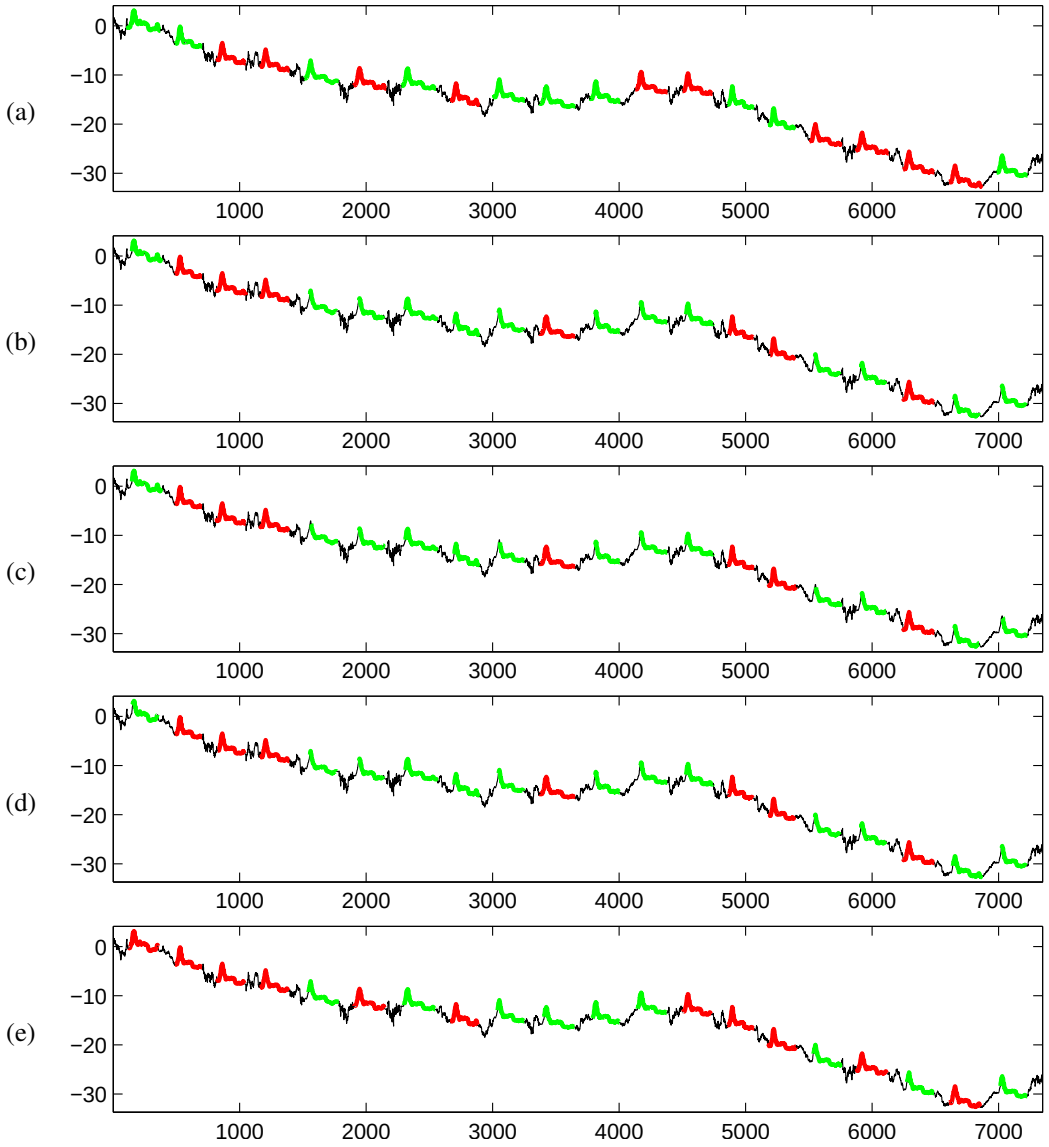

Figure C.22: Strawberry dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

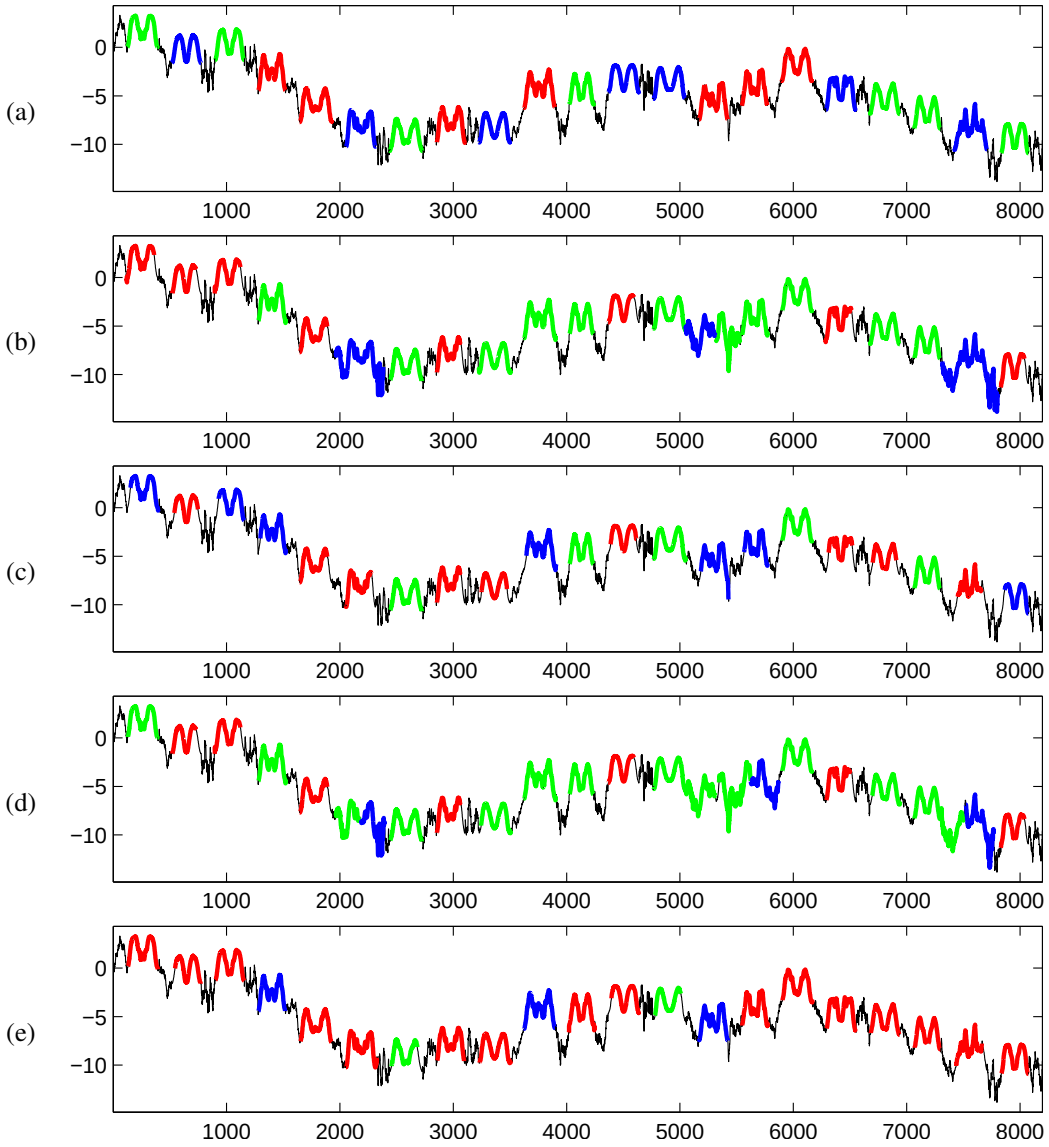

Figure C.23: ArrowHead dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

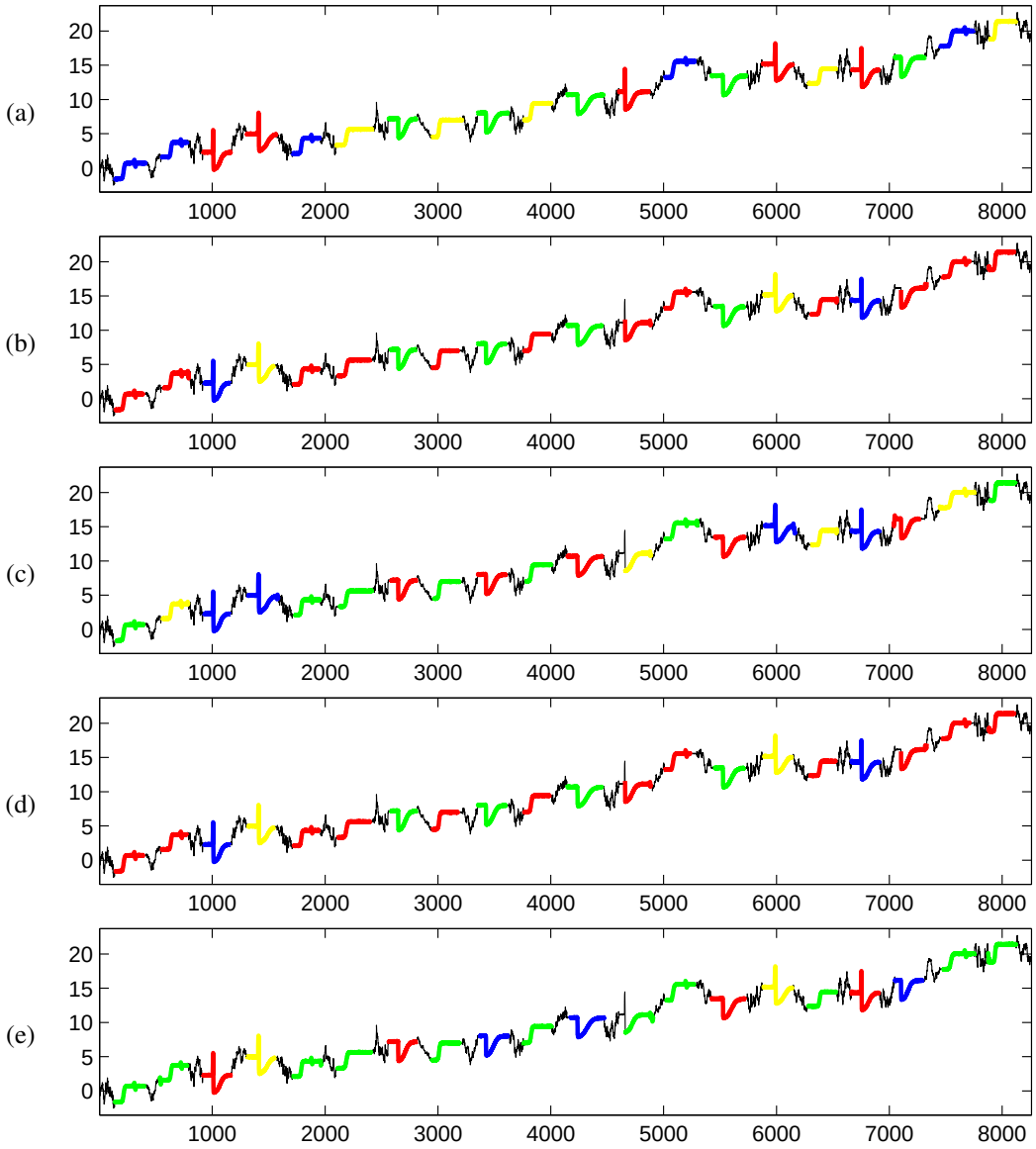

Figure C.24: Trace dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

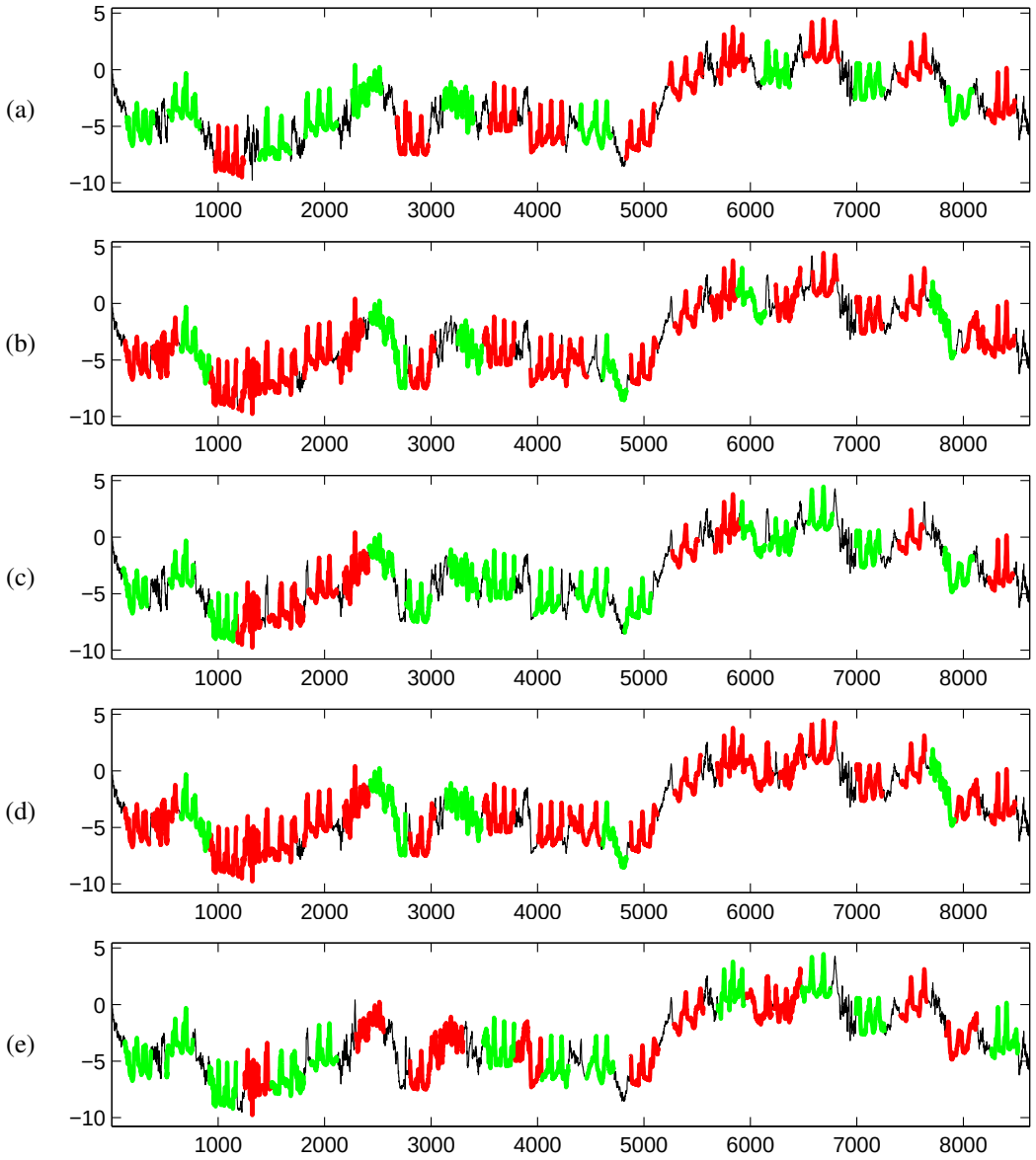

Figure C.25: ToeSegmentation1 dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

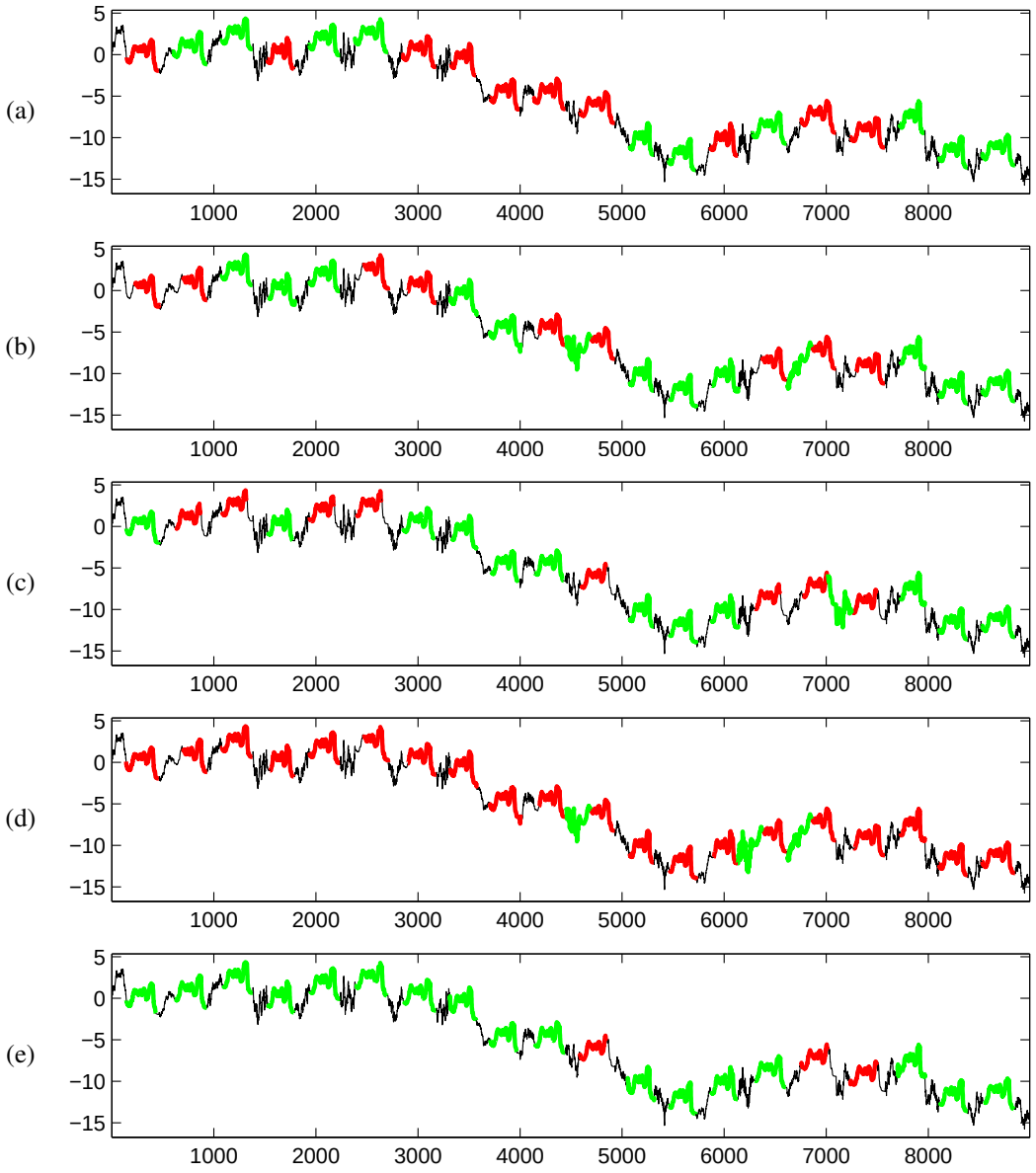

Figure C.26: Coffee dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

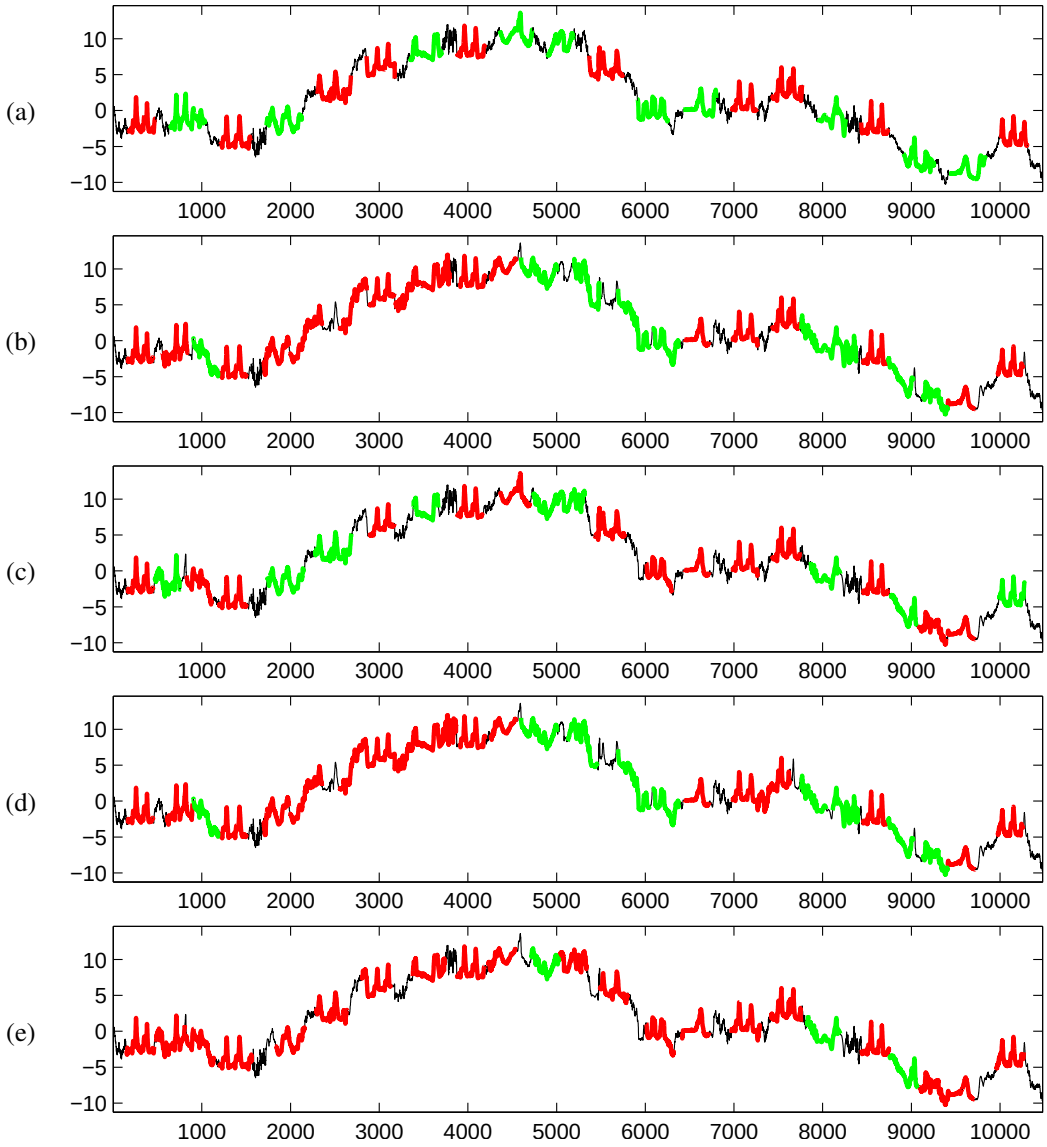

Figure C.27: ToeSegmentation2 dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

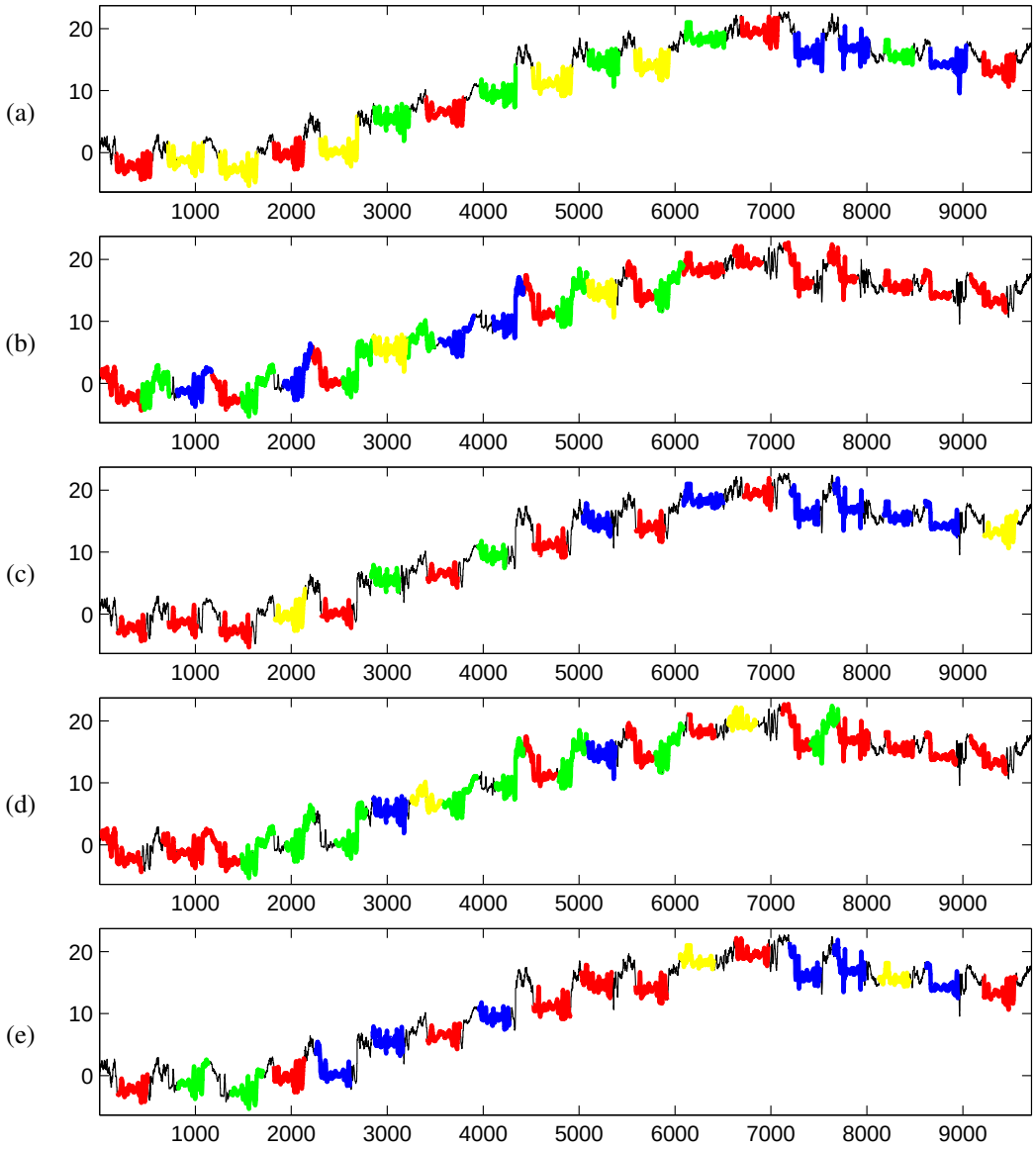

Figure C.28: FaceFour dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

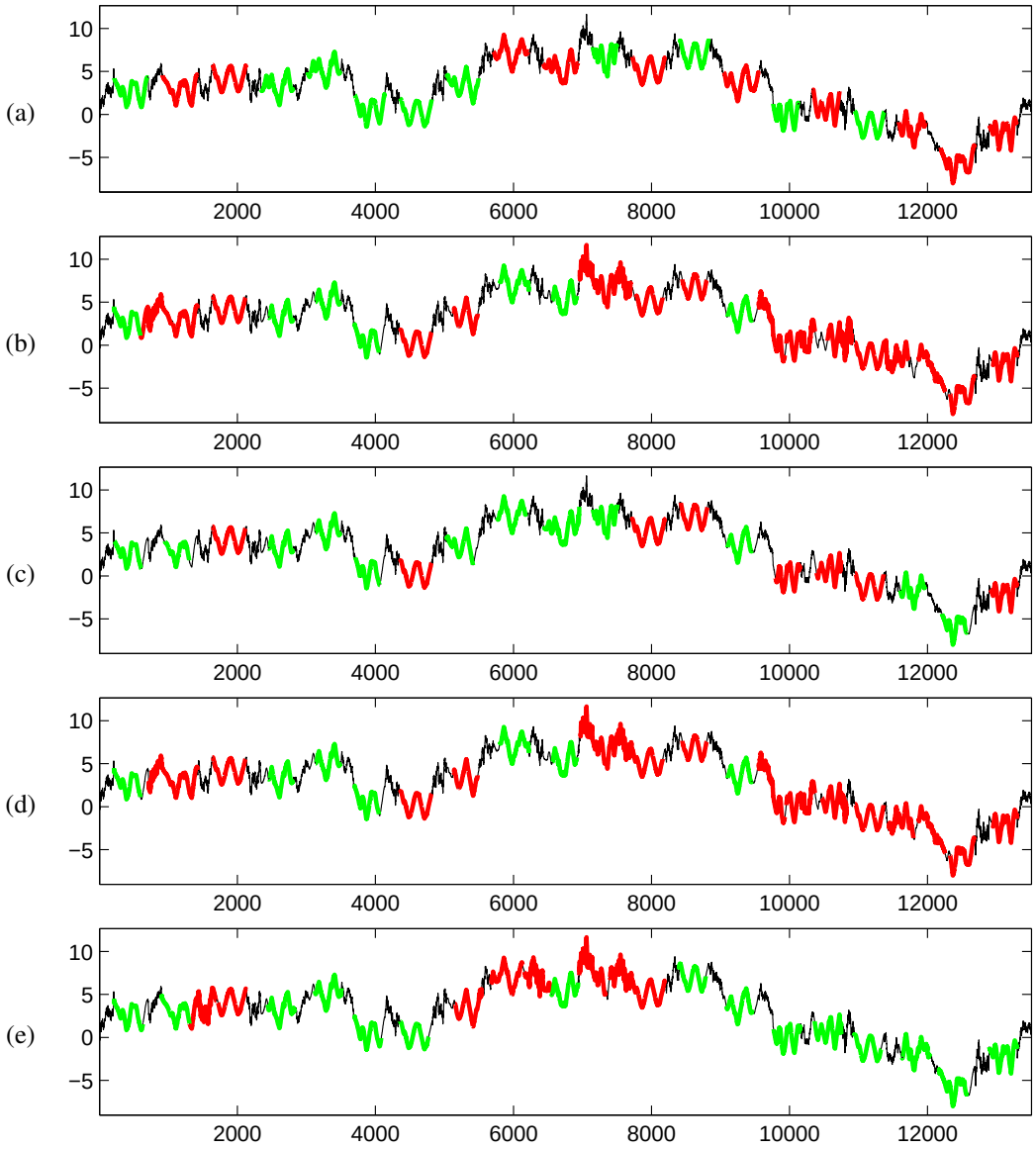

Figure C.29: yoga dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

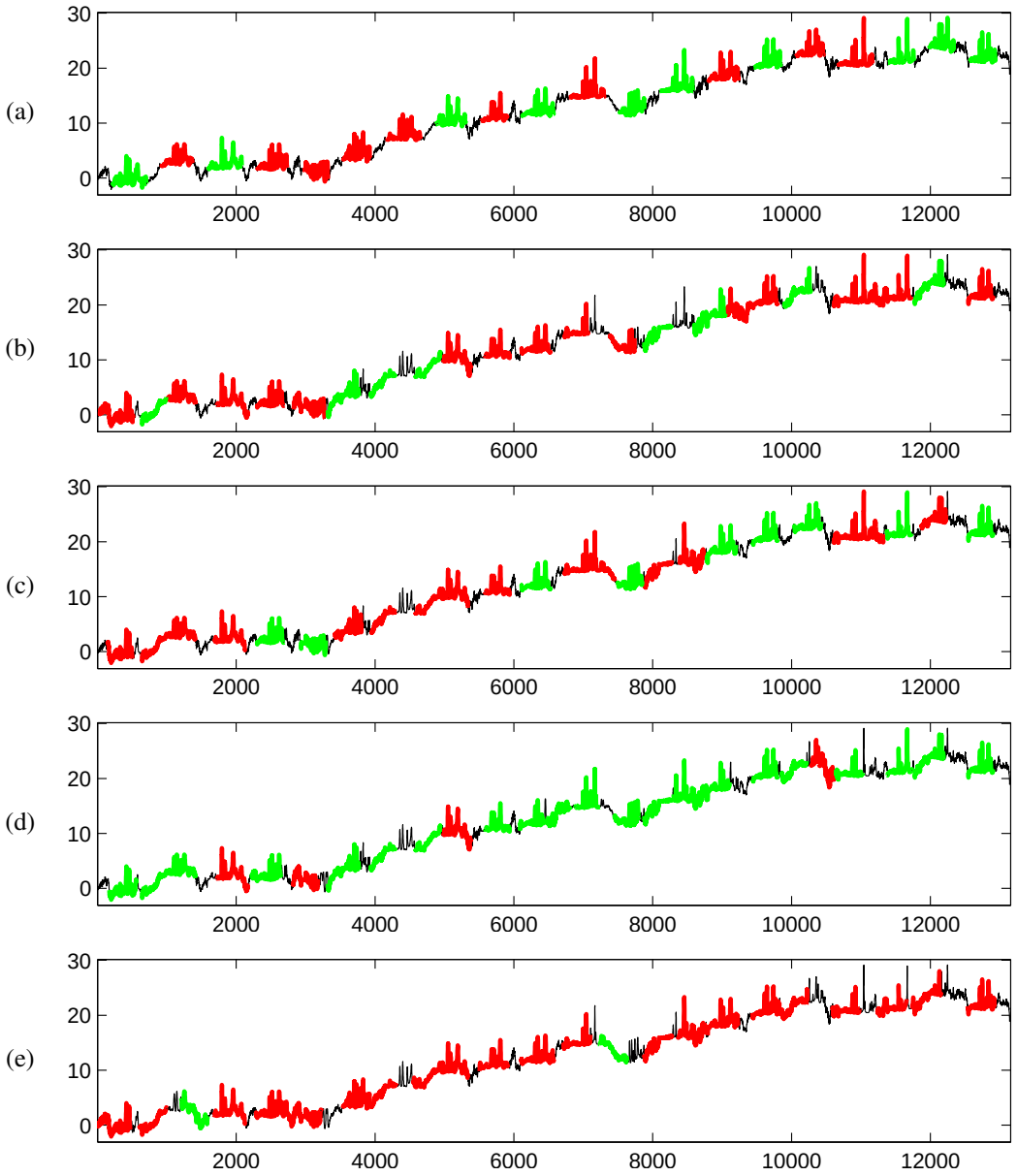

Figure C.30: Ham dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

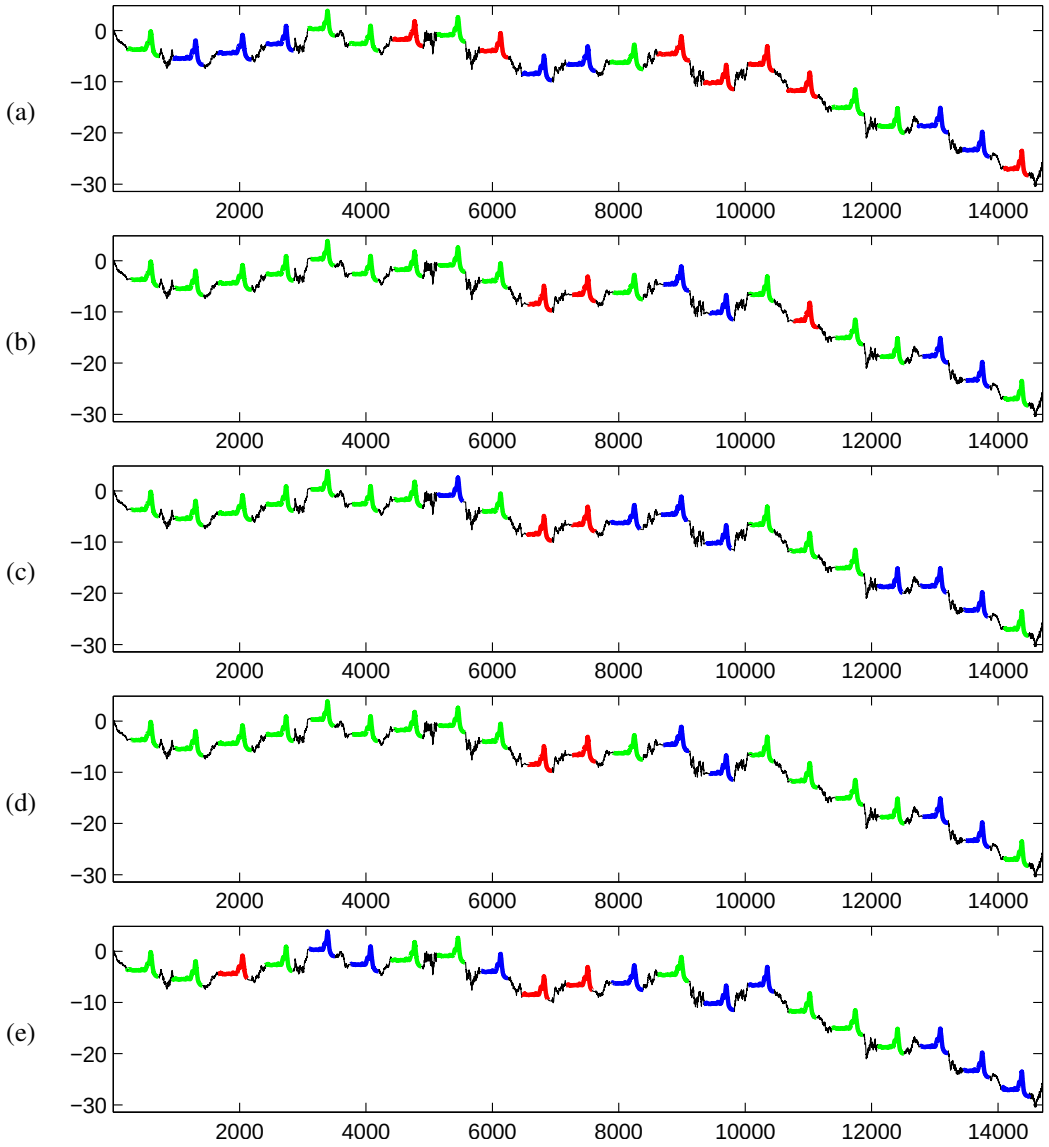

Figure C.31: Meat dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

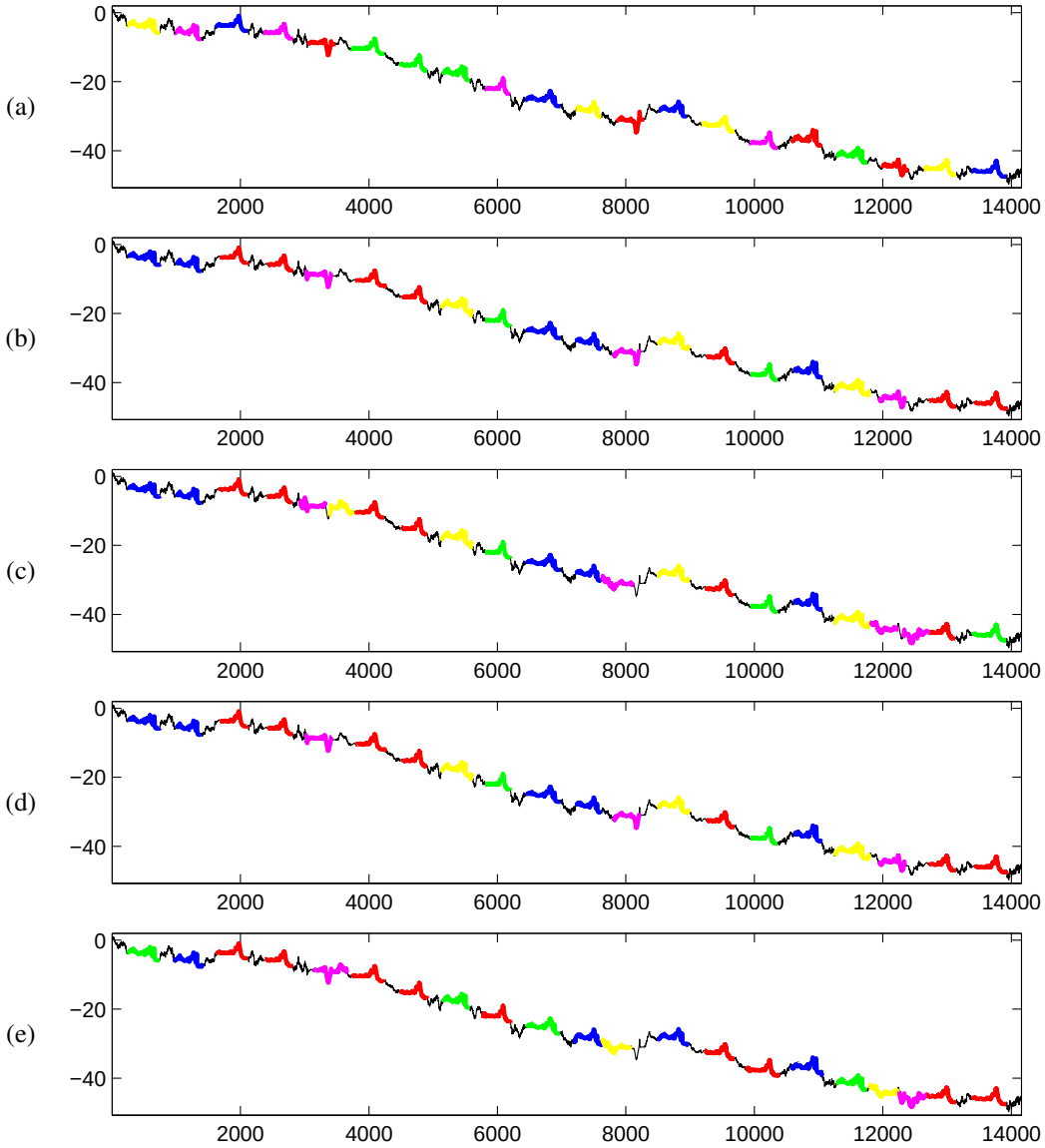

Figure C.32: Beef dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

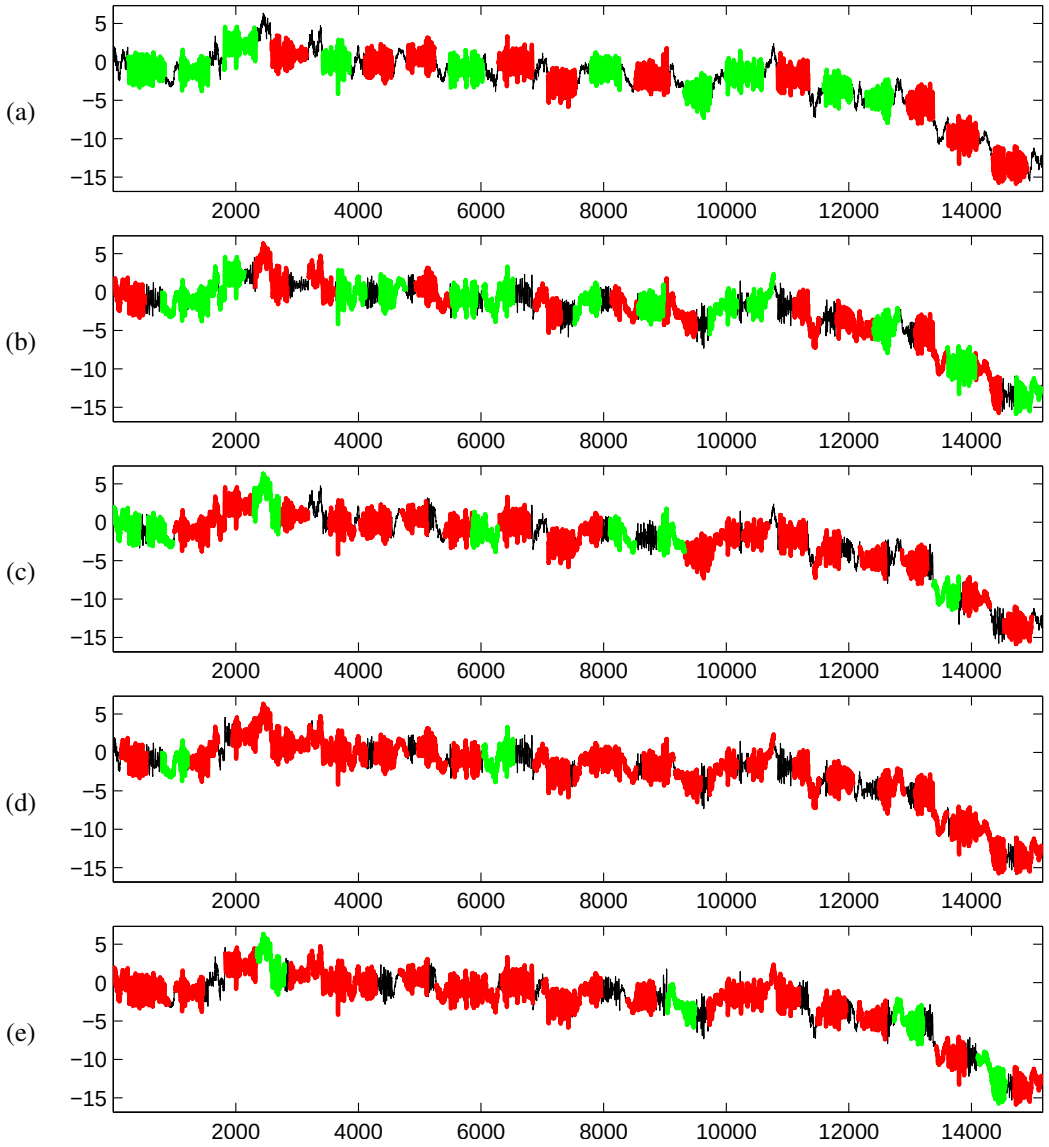

Figure C.33: FordA dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

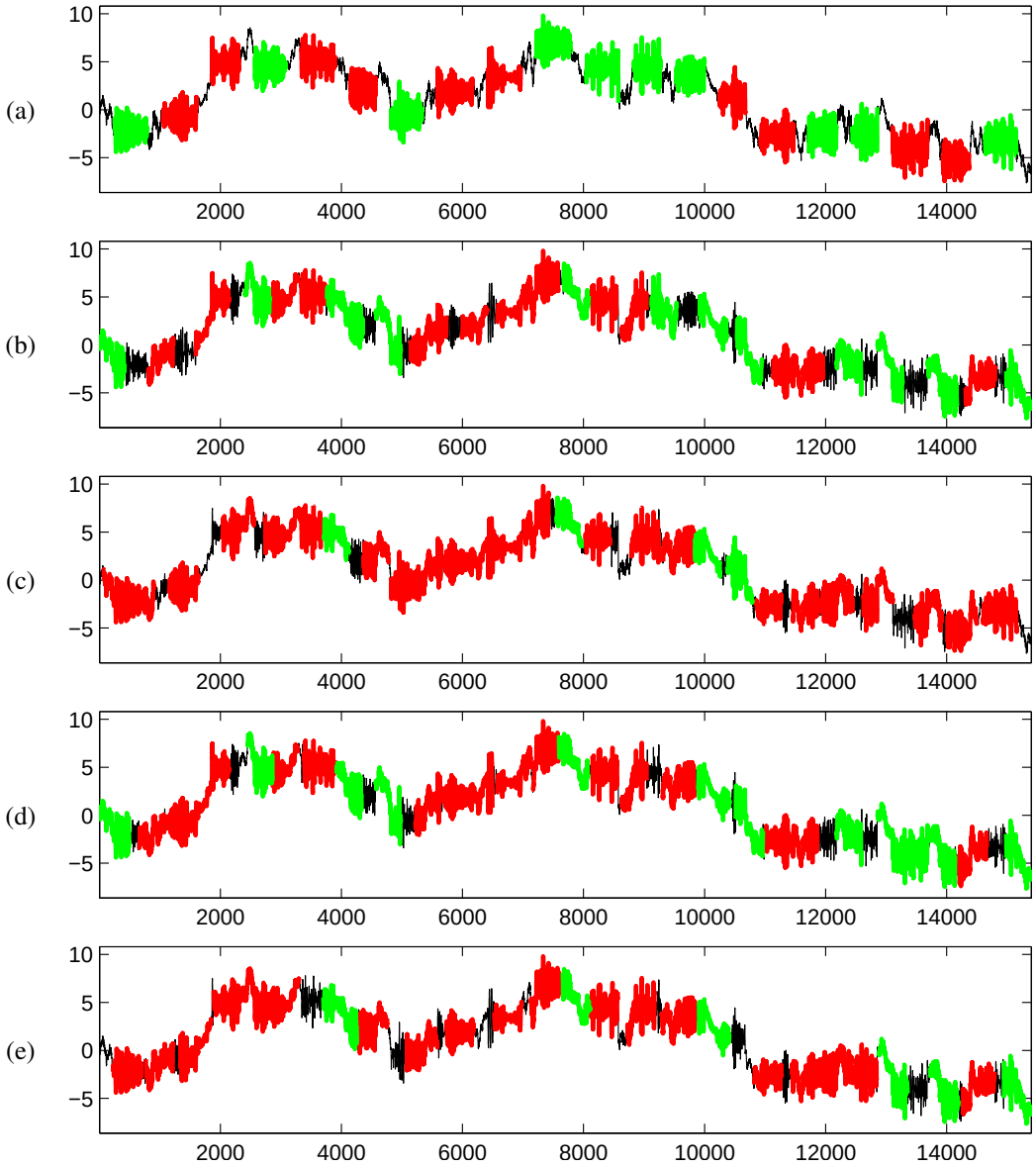

Figure C.34: FordB dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

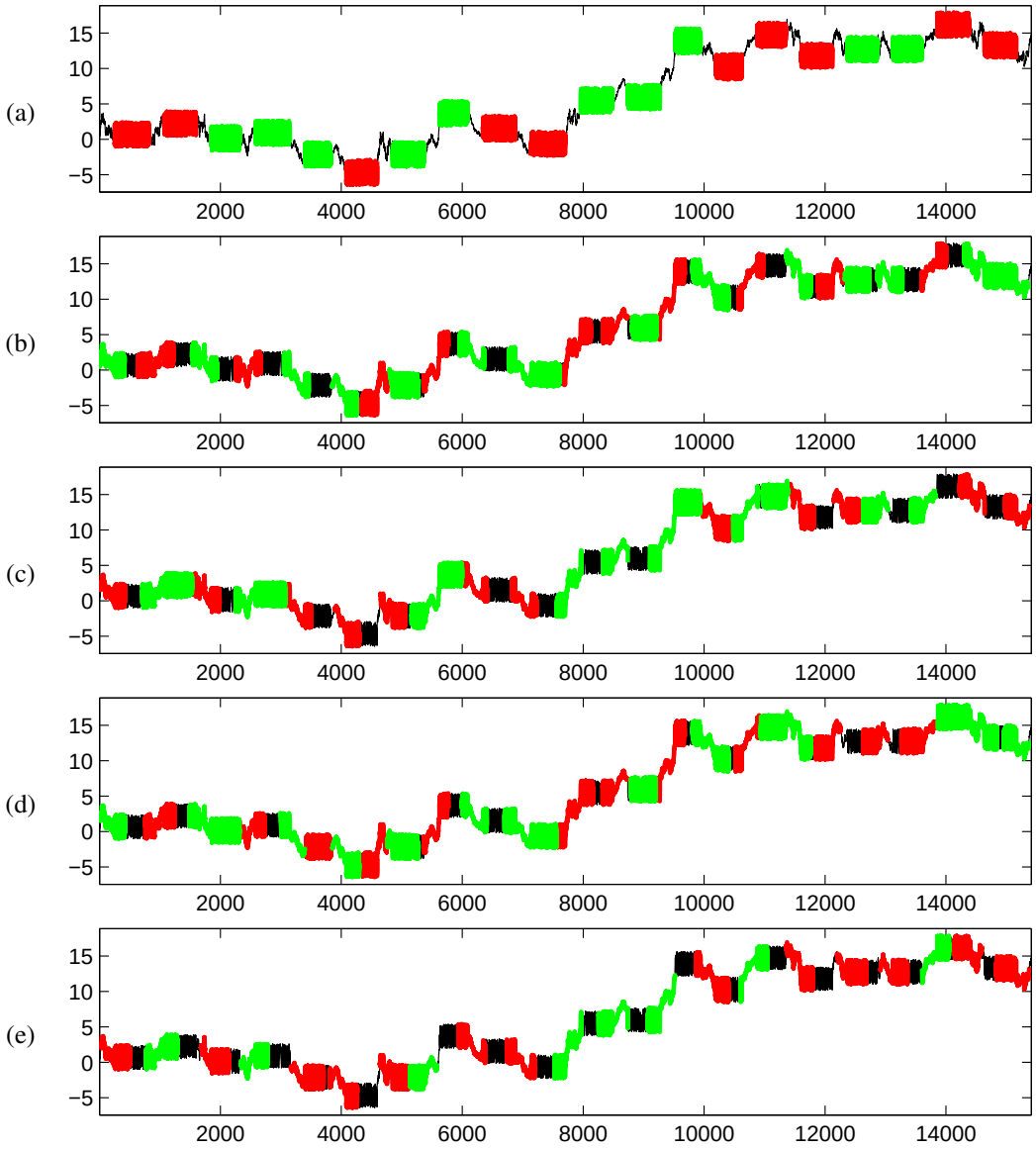

Figure C.35: ShapeletSim dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

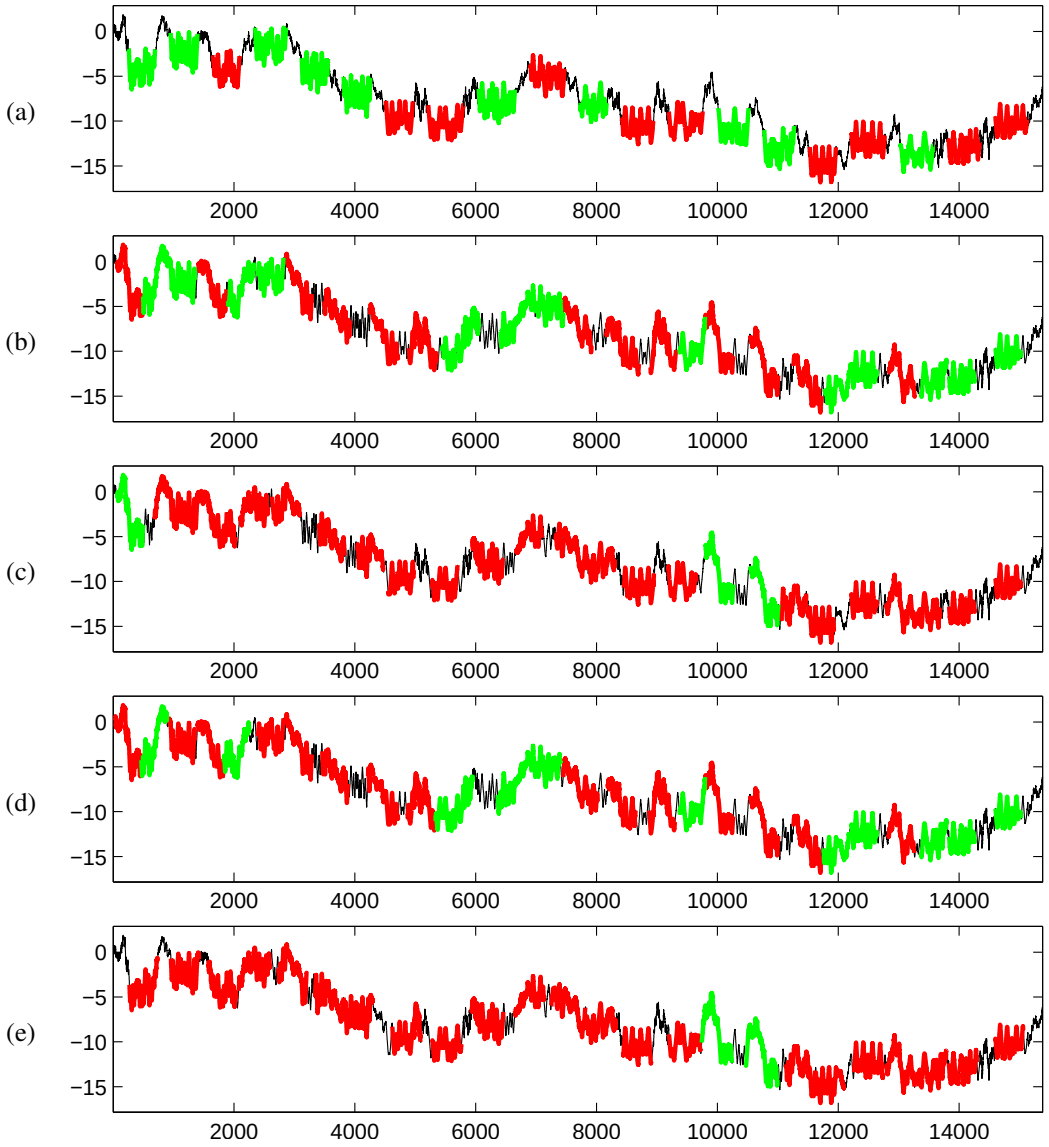

Figure C.36: BeetleFly dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

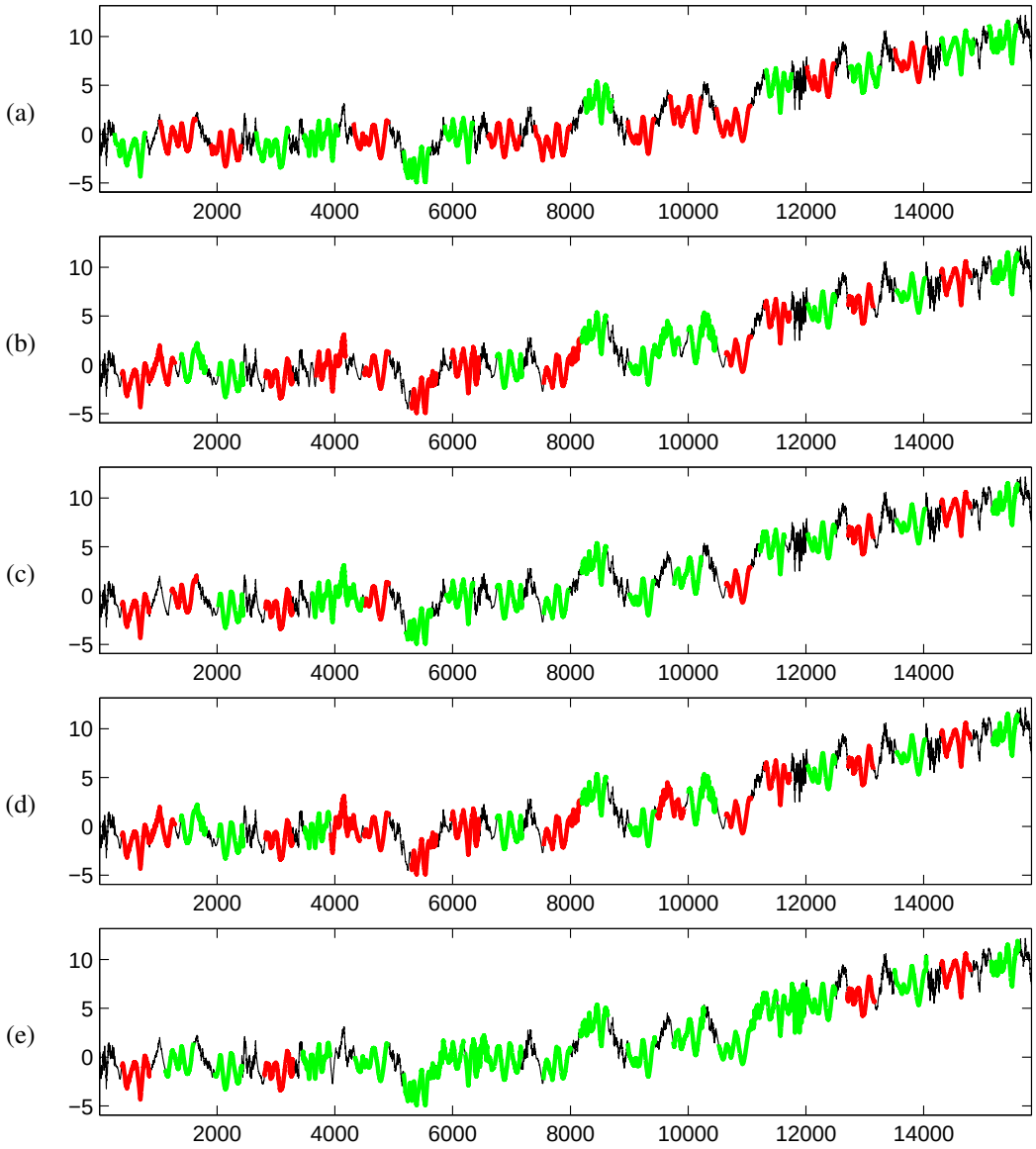

Figure C.37: BirdChicken dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

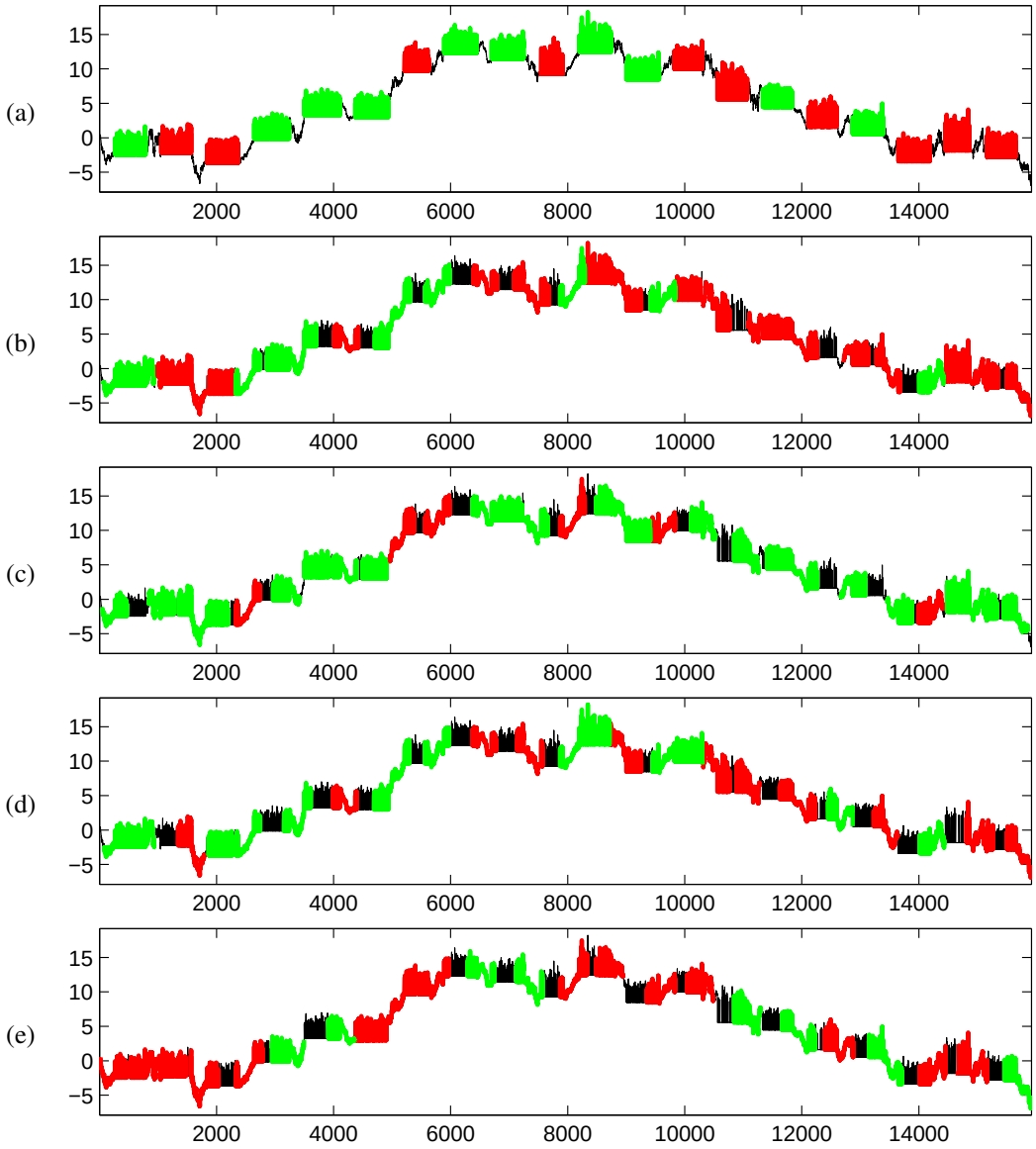

Figure C.38: Earthquakes dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

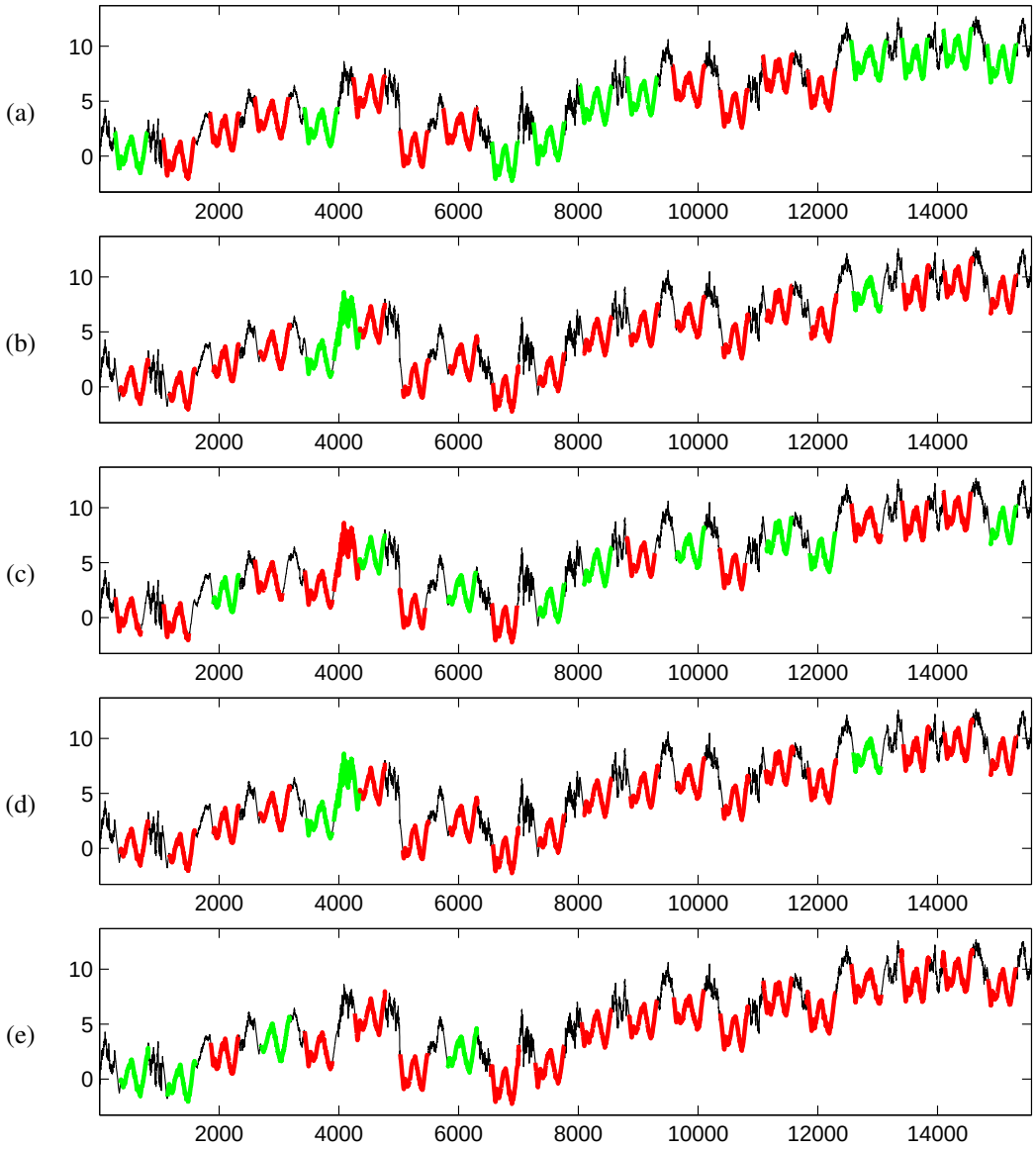

Figure C.39: Herring dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

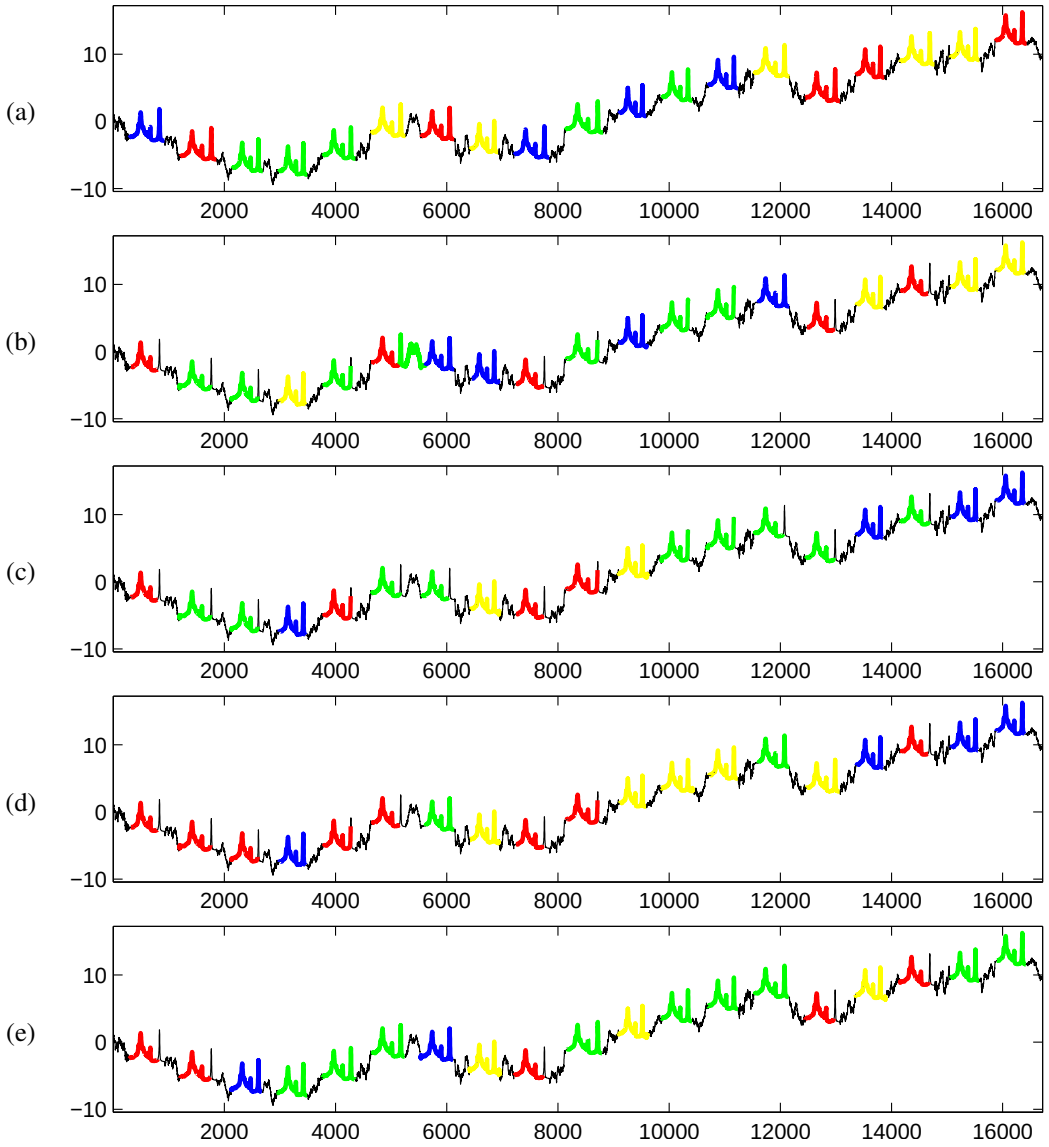

Figure C.40: OliveOil dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

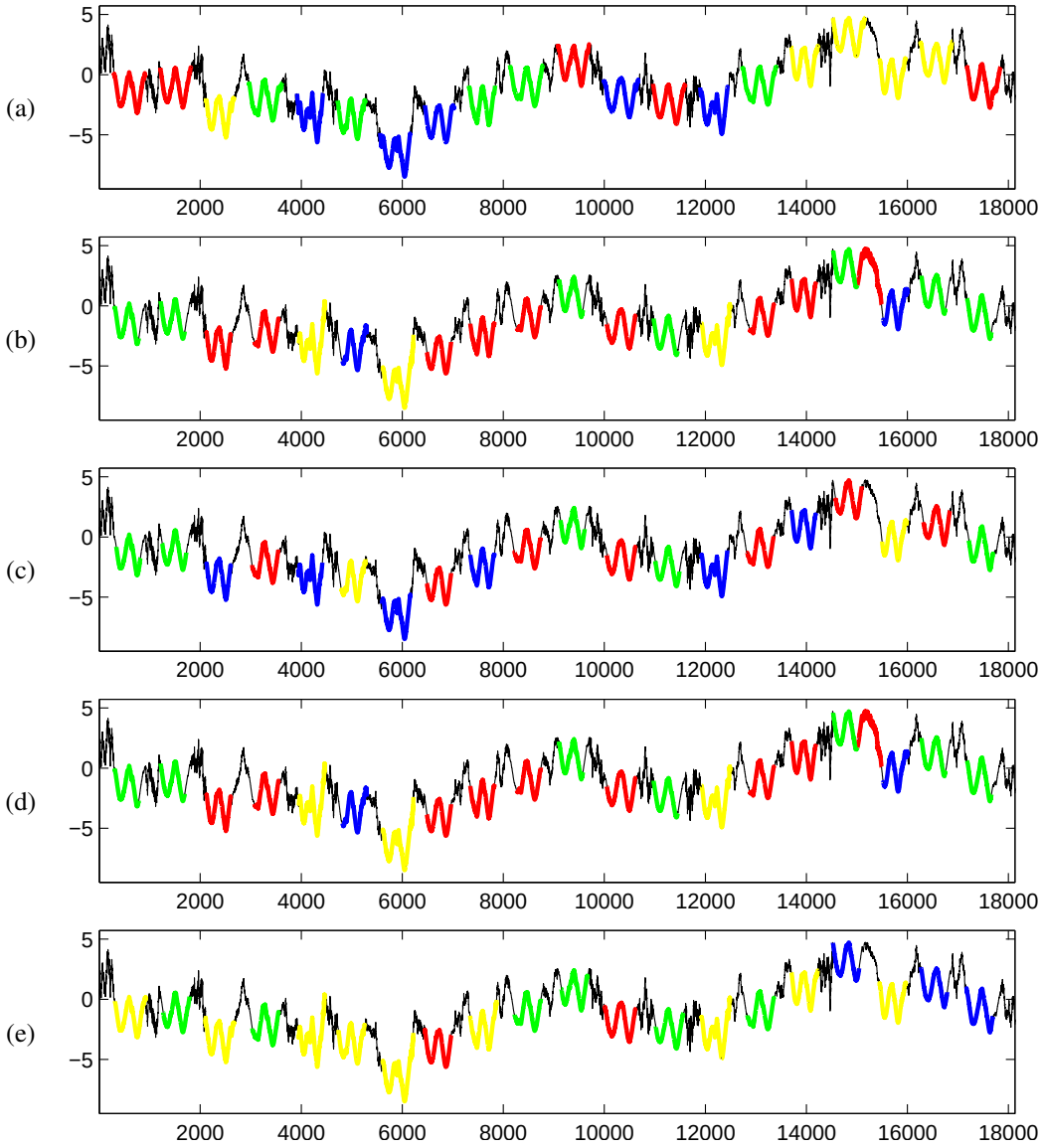

Figure C.41: Car dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

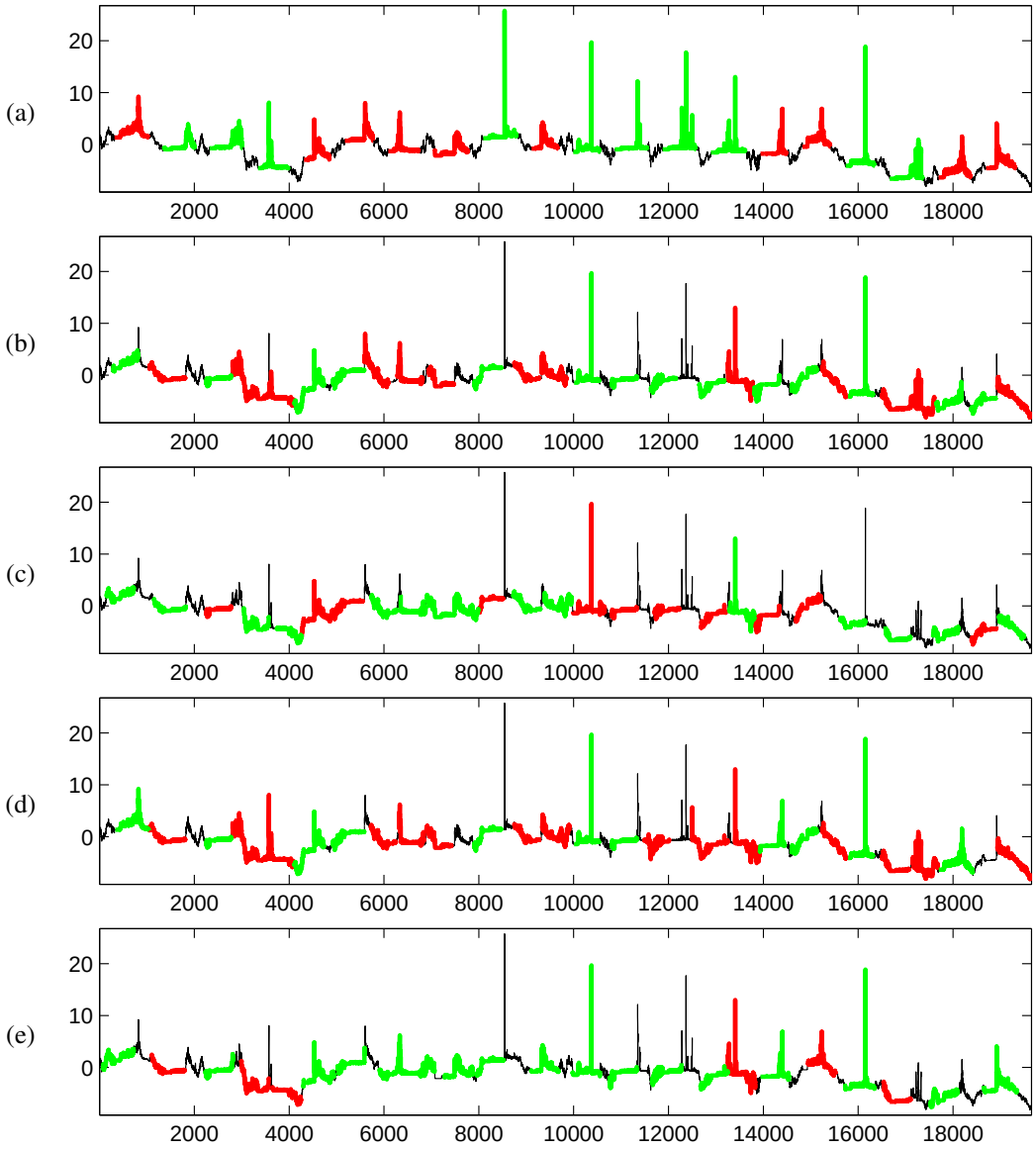

Figure C.42: Lighting2 dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

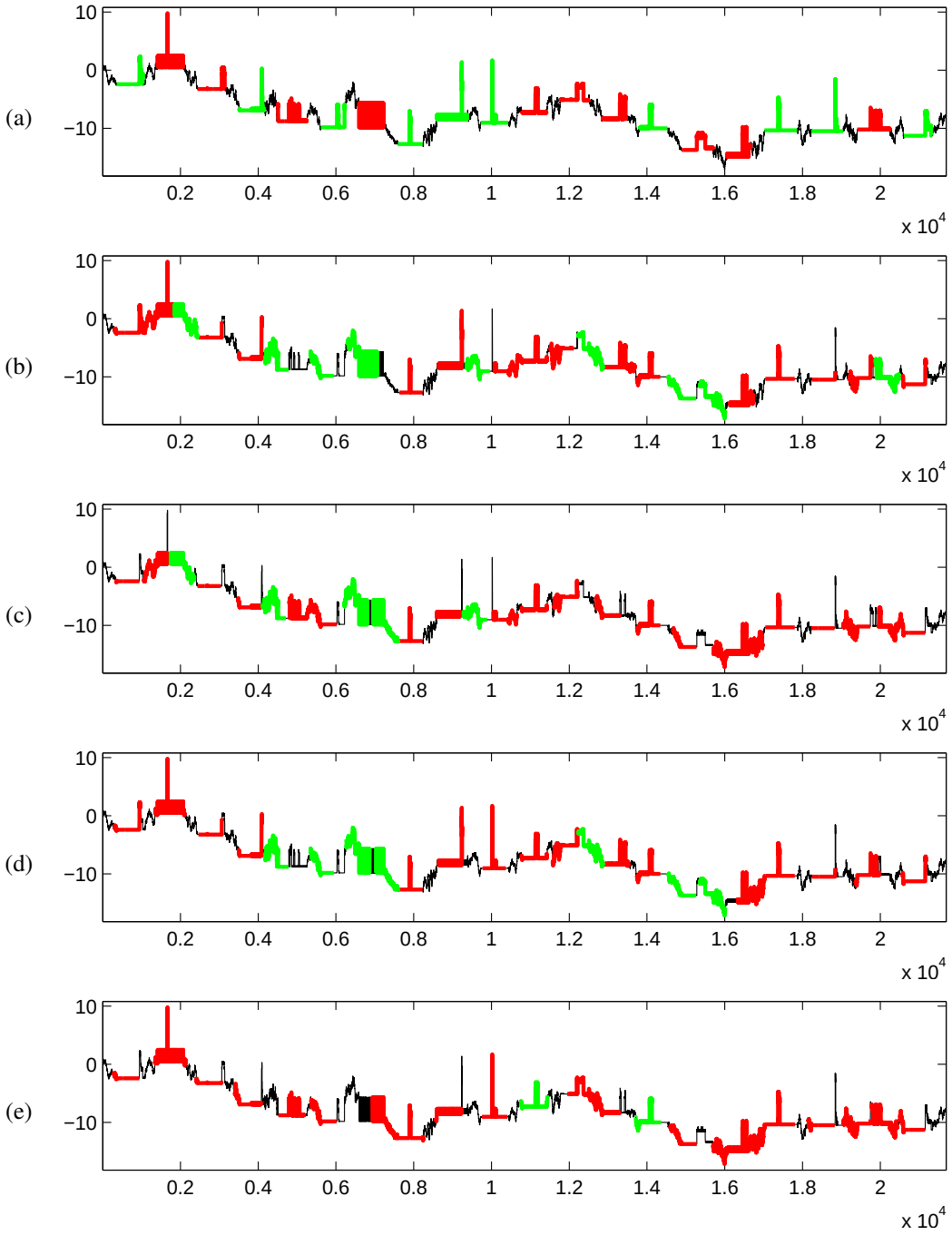

Figure C.43: Computers dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

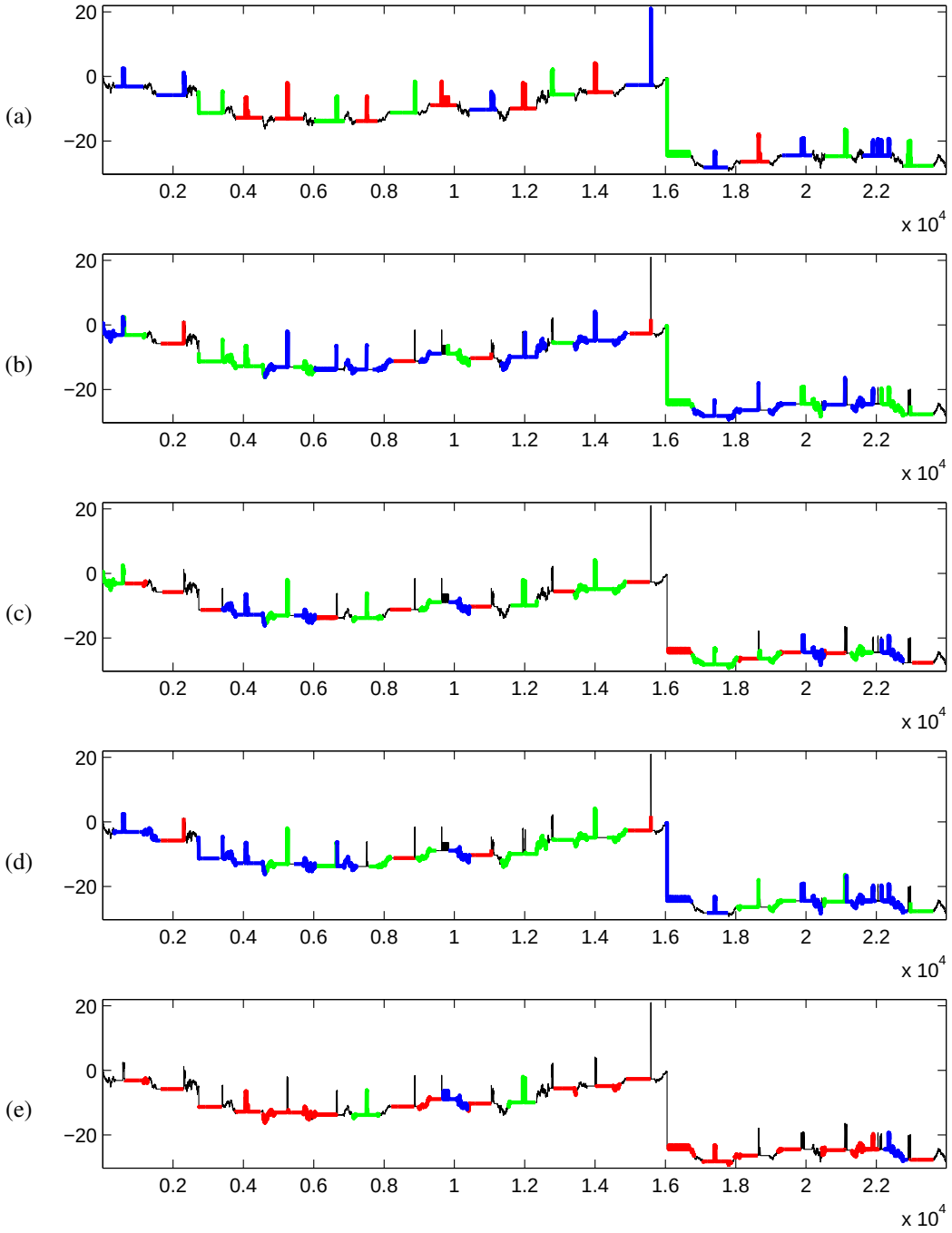

Figure C.44: LargeKitchenAppliances dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

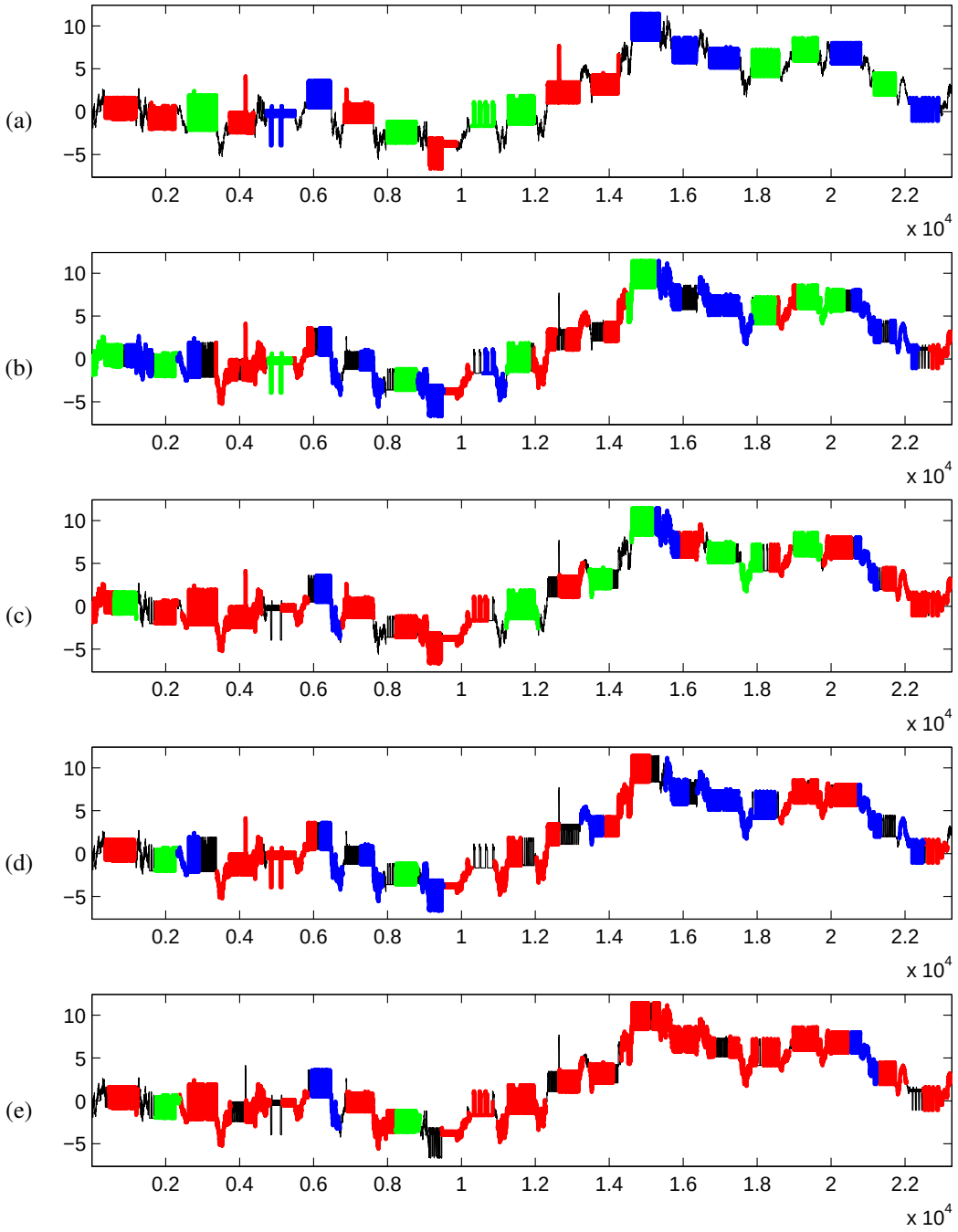

Figure C.45: RefrigerationDevices dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

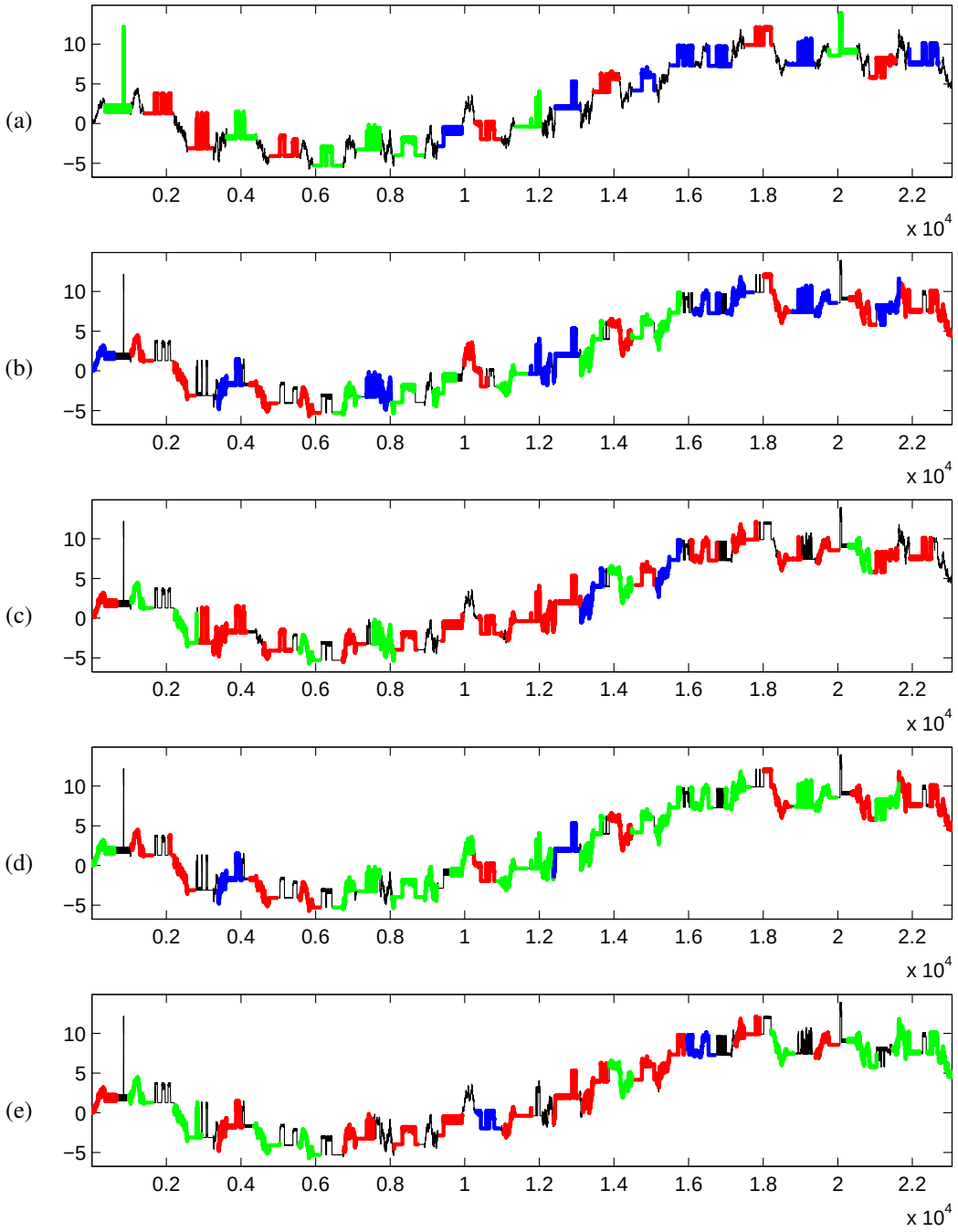

Figure C.46: ScreenType dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

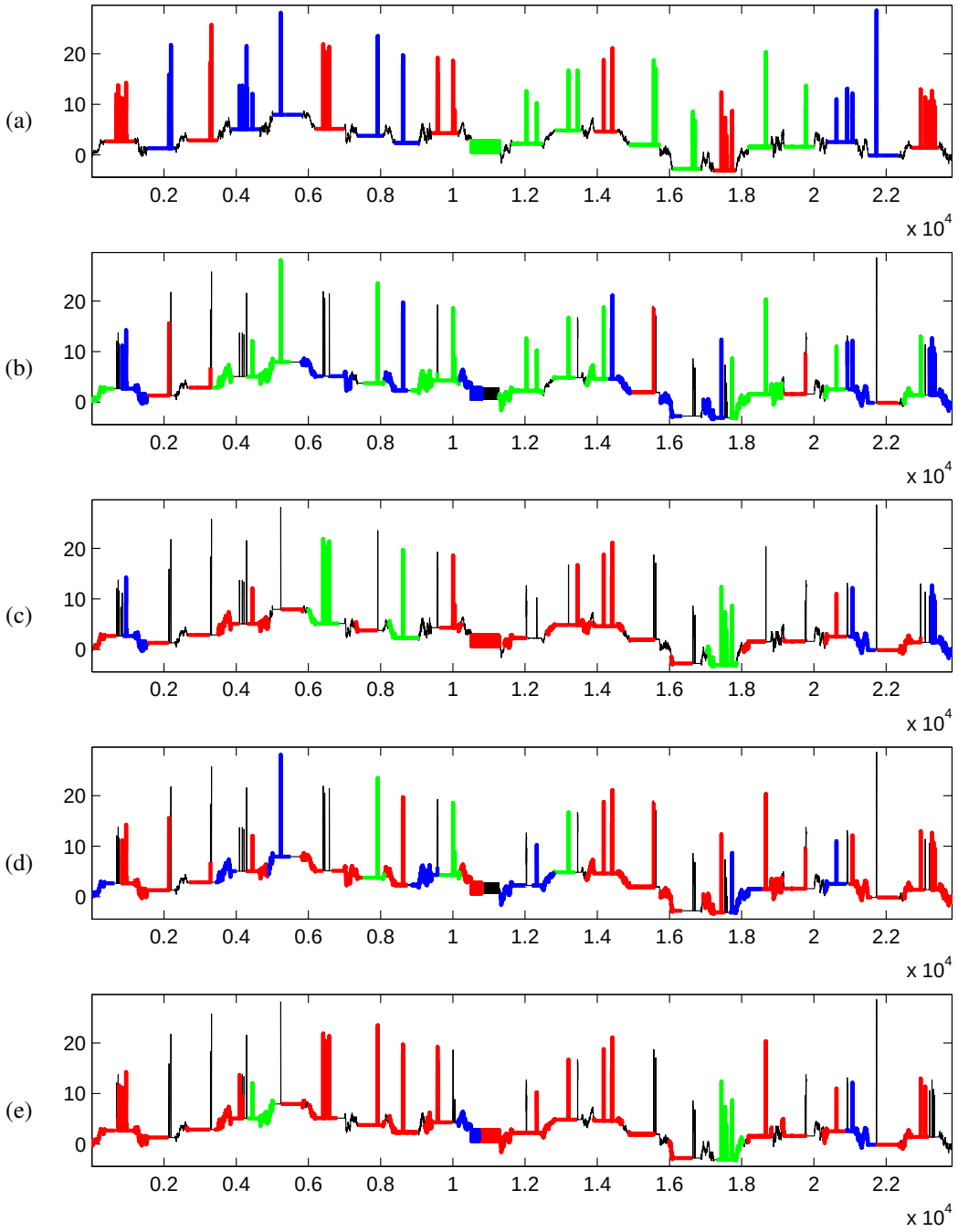

Figure C.47: SmallKitchenAppliances dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

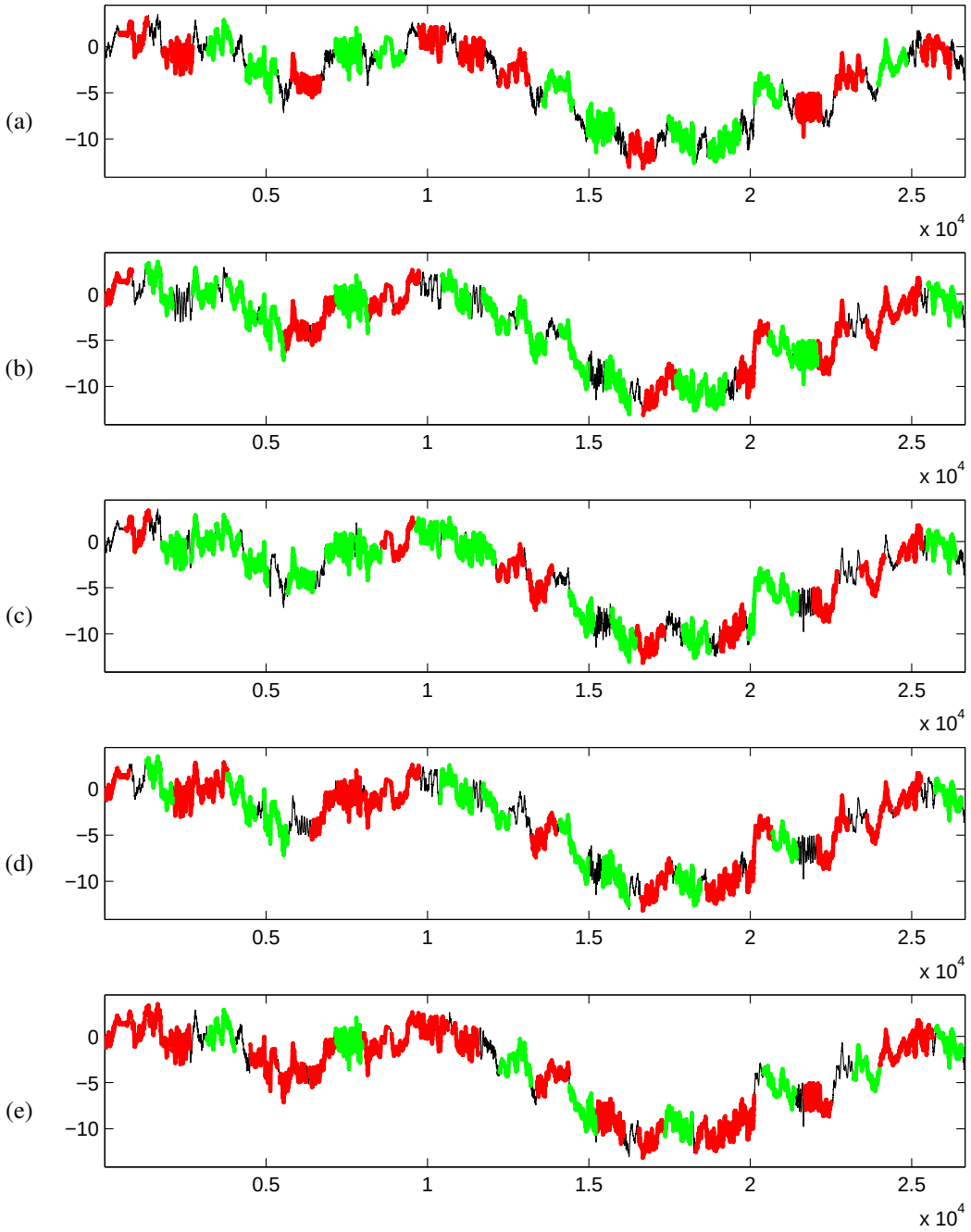

Figure C.48: WormsTwoClass dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

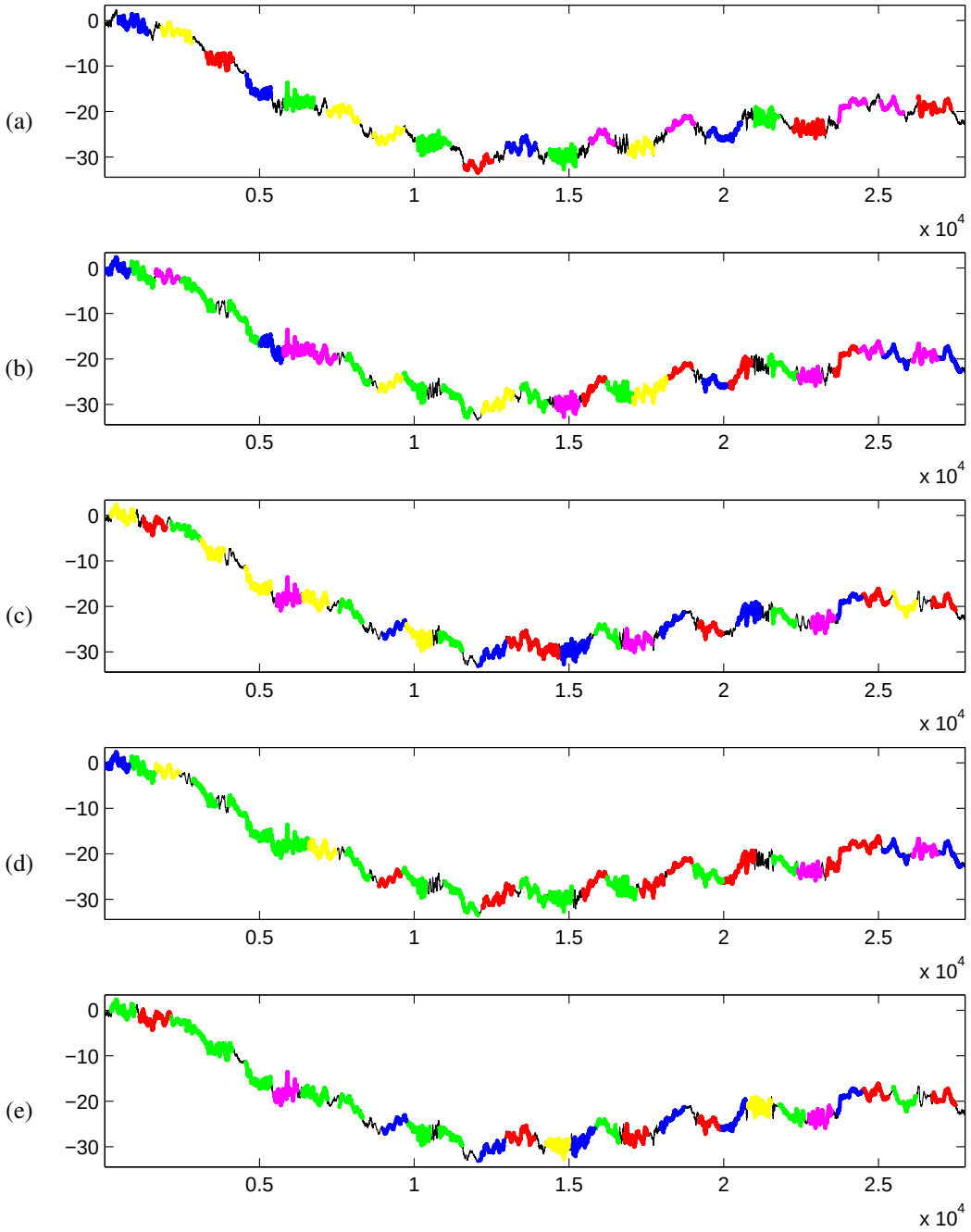

Figure C.49: Worms dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

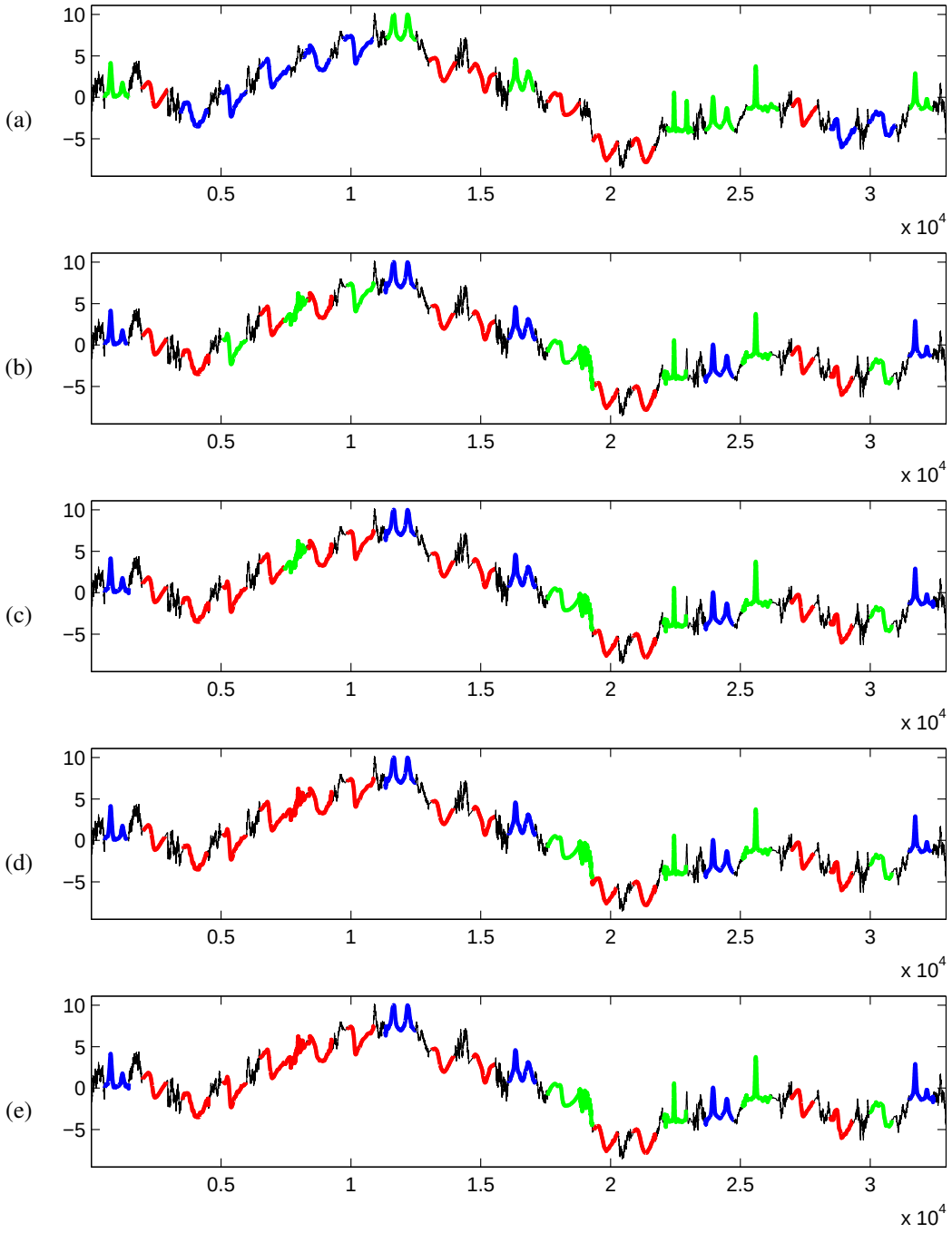

Figure C.50: StarLightCurves dataset: (a) Input time series labeled by classes of planted data with scaling factor $f = 1.2$. (b), (c), (d) and (e) are output from SSTSC with scaling factor $f = 1.2$ by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

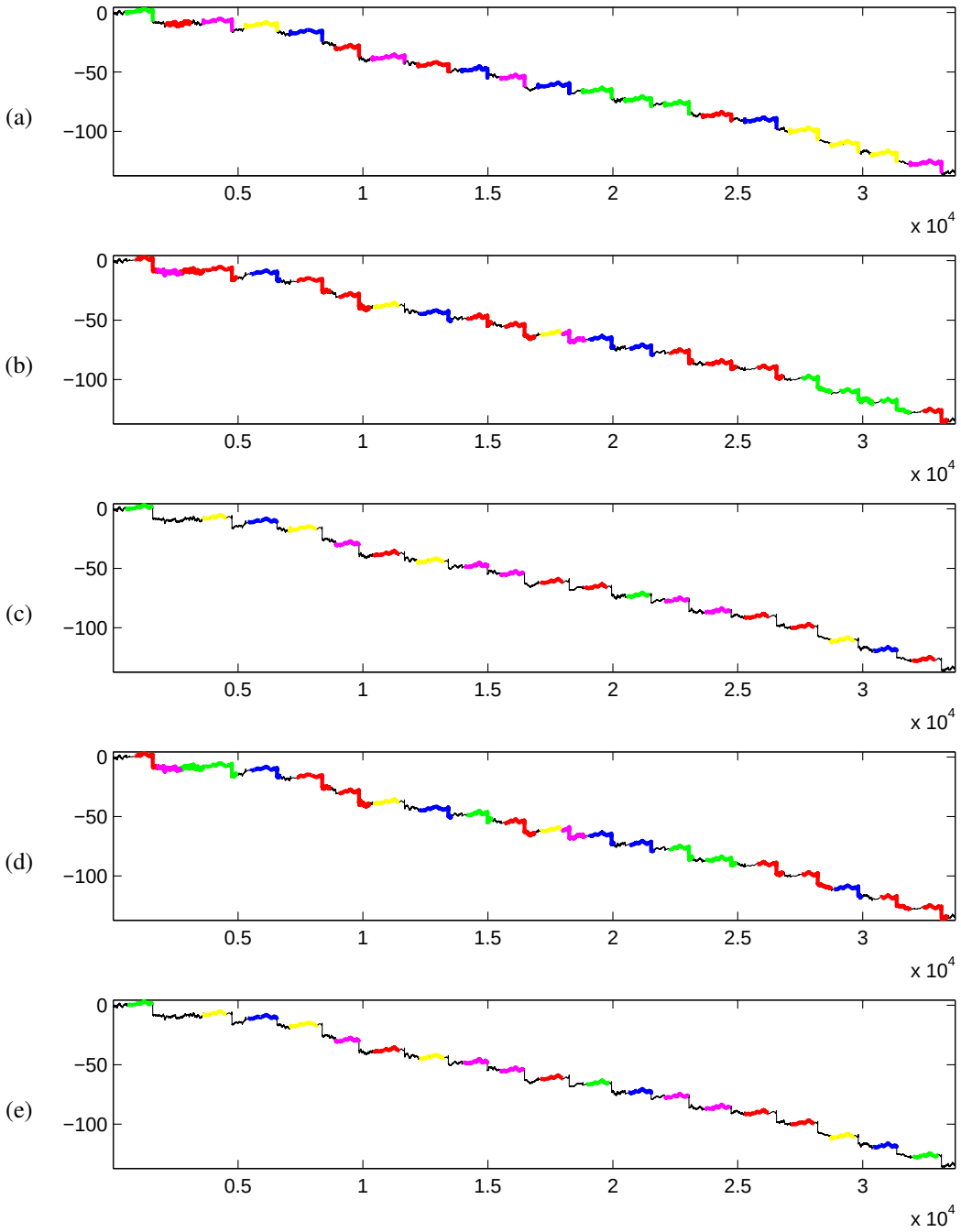

Figure C.51: Haptics dataset: (a) Input time series labeled by classes of planted data with scaling factor f = 1.2. (b), (c), (d) and (e) are output from SSTSC with scaling factor f = 1.2 by using E-AA-Z, E-AA-L, E-SA-Z and E-SA-L, respectively.

Biography

Sura Rodpongpun was born in Bangkok, Thailand, on December 26th, 1985. He graduated from Suankularb Wittayalai School in 2004. Then, he received B.Eng. in Computer Engineering from Mahidol University, Thailand, in 2008 with second class honor. His doctorate has been under supervision of Asst. Prof. Dr. Chotirat Ann Ratanamahatana. During his Ph.D. study, he was a visiting researcher at Graduate School of Biomedical Engineering, University of New South Wales, under supervision of Associate Professor Stephen James Redmond for one year (March 2014 to March 2015), and he was granted scholarships from the Thailand Research Fund and Chulalongkorn University given through the Royal Golden Jubilee Ph.D. Program (PHD/0319/2551 to S. Rodpongpun) (June 2010 to July 2016). His research interests include but not limited to time series data mining, machine learning, and artificial intelligent.