การทำให้เป็นบรรทัดฐานและการวัดความเหมือนของอนุกรมเวลาที่ทนทานต่อสเกลที่แปรเปลี่ยน

นายอริยวัฒน์ ชนบดีเฉลิมรุ่ง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2560

Robust Scale-Invariant Normalization and Similarity Measurement for Time Series Data

Mr. Ariyawat Chonbodeechalermroong

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2017

Thesis Title	Robust Scale-Invariant Normalization and Similarity Measurement for Time Series Data
By	Mr. Ariyawat Chonbodeechalermroong
Field of Study	Computer Engineering
Thesis Advisor	Associate Professor Chotirat Ratanamahatana, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

....Dean of the Faculty of Engineering
(Associate Professor Supot Teachavorasinskun, D.Eng.)

THESIS COMMITTEE

....Chairman
(Duangdao Wichadakul, Ph.D.)

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

....External Examiner
(Haemwaan Sivaraks, Ph.D.)

อริยวัฒน์ ชนบดีเฉลิมรุ่ง : การทำให้เป็นบรรทัดฐานและการวัดความเหมือนของอนุกรมเวลาที่ทนทานต่อสเกลที่แปรเปลี่ยน (Robust Scale-Invariant Normalization and Similarity Measurement for Time Series Data) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร.โชติรัตน์ รัตนามหัทธนะ, 51 หน้า.

การจำแนกประเภท (Classification) เป็นหนึ่งในงานที่แพร่หลายในการทำเหมือง อนุกรมเวลา (Time Series mining) ไดนามิกไทม์วอร์ปปิง (Dynamic Time Warping) และการหาลำดับย่อยร่วมยาวสุด (Longest Common Subsequence) เป็นอัลกอริทึมที่ใช้วัดความเหมือนของอนุกรมเวลาแบบปรับแนวไม่เชิงเส้นที่ใช้กันอย่างแพร่หลาย เพียงแต่ว่าวิธีทั้งสองนี้เหมาะสมกับอนุกรมเวลาที่มีสเกลแนวแอมพลิจูดที่ใกล้เคียงกันเท่านั้น ในขณะที่ข้อมูลจากเซนเซอร์และข้อมูลอนุกรมเวลาส่วนมากที่พบได้จริงบนโลกมักจะมีสัญญาณรบกวน ค่าข้อมูลขาดหาย ค่าข้อมูลผิดปกติ รวมไปถึงการแปรเปลี่ยนของสเกลทั้งสองแกน ซึ่งการทำให้เป็นบรรทัดฐานแบบซี (Z-normalization) ไม่สามารถรับมือได้อย่างเหมาะสม งานวิจัยนี้นำเสนออัลกอริทึมการทำให้เป็นบรรทัดฐานด้วยลักษณะท้องถิ่น (Local Feature Normalization: LFN) และลักษณะท้องถิ่นที่ถูกสเกล (Local Scaling Feature: LSF) ซึ่งสามารถทำอนุกรมเวลาที่มีสัญญาณรบกวน สัญญาณหาย และการบิดงอในแกนเวลา ให้เป็นบรรทัดฐานได้ นอกจากนี้ลักษณะท้องถิ่นที่ถูกสเกลยังถูกมาใช้ประโยชน์ในการจับคู่อนุกรมเวลาที่มีลำดับย่อยหลาย ๆ ลำดับย่อยที่มีสเกลหลากหลาย อัลกอริทึมนี้ถูกเรียกว่าการหาคุณลักษณะท้องถิ่นที่ถูกสเกลร่วมที่ยาวที่สุด (Longest Common Local Scaling Feature: LCSF) เทียบกับการทำให้เป็นบรรทัดฐานแบบซีในด้านการจำแนกประเภทแล้ว LFN ทำงานได้อย่างดียิ่งโดยเฉพาะชุดข้อมูลที่มีสัญญาณรบกวน สำหรับชุดข้อมูลสังเคราะห์และแอปพลิเคชันจริงสำหรับการออกกำลังกายฟื้นฟูข้อมือด้วยเซนเซอร์จากโทรศัพท์มือถือ LCSF สามารถให้ผลที่ดีกว่าอัลกอริทึมอื่น ๆ อย่างมีนัยสำคัญ อย่างไรก็ตาม LCSF มีข้อเสียร้ายแรงในด้านความเร็วและจำนวนพารามิเตอร์ ในที่สุดงานวิจัยนี้จึงเสนออัลกอริทึมที่ชื่อว่าไดนามิกไทม์วอร์ปปิงแบบสเกลท้องถิ่น (local scaling Dynamic Time warping: LSDTW) ที่เร็วกว่าและมีจำนวนพารามิเตอร์น้อยกว่า LCSF แต่สามารถก้าวข้าม LCSF และอัลกอริทึมที่ทันสมัยอื่น ๆ ได้

ภาควิชา	วิศวกรรมคอมพิวเตอร์	ลายมือชื่อนิสิต	....
สาขาวิชา	วิศวกรรมคอมพิวเตอร์	ลายมือชื่อ อ.ที่ปรึกษาหลัก	....
ปีการศึกษา	2560		

6070375521 : MAJOR COMPUTER ENGINEERING
KEYWORDS: TIME SERIES FEATURES / DISTANCE FUNCTION / DYNAMIC TIME WARPING / TIME SERIES NORMALIZATION / LONGEST COMMON SUBSEQUENCE / MULTIPLE-SUBSEQUENCE-WITH-MULTIPLE-SCALE TIME SERIES

ARIYAWAT CHONBODEECHALERMROONG: Robust Scale-Invariant Normalization and Similarity Measurement for Time Series Data. ADVISOR: ASSOC. PROF. CHOTIRAT RATANAMAHATANA, Ph.D., 51 pp.

Classification is one of the most prevalent tasks in time series mining. Dynamic Time Warping and Longest Common Subsequence are well-known and widely used algorithms to measure similarity between two time series sequences using non-linear alignment. However, these algorithms work best when the time series pair has similar amplitude scaling. Unfortunately, sensor data and most real-world time series data usually contain noise, missing values, outlier, and variability or scaling in both axes, which is not suitable for the widely used Z-normalization. This research introduces the Local Feature Normalization (LFN) and its Local Scaling Feature (LSF), which can be used to robustly normalize noisy/warped/missing-valued time series. In addition, LSF is utilized to help matching time series containing multiple subsequences with a variety of scales; this algorithm is called Longest Common Local Scaling Feature (LCSF). Compared to the use of Z-normalized data, our classification results show that our proposed LFN is impressively robust, especially on high-error and noisy datasets. On both synthetic and real application data for wrist strengthening rehabilitation exercise using a mobile phone sensor, our LCSF similarity measure also significantly outperforms other existing methods by a large margin. However, LCSF has the serious drawback on speed and number of parameters. Finally, this thesis proposes local scaling Dynamic Time warping (LSDTW), which has faster speed and fewer parameters than LCSF, but LSDTW can impressively outperform LCSF and other state-of-the-art approaches.

Department:	Computer Engineering	Student's Signature	....
Field of Study:	Computer Engineering	Advisor's Signature	....
Academic Year:	2017		

ACKNOWLEDGEMENTS

Firstly, I would like to thank my adviser, Assoc. Prof. Chotirat Ratanamahatana, Ph.D., for any contributions to this thesis.

Secondly, I would like to thank the Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University for the precious scholarship for my master's degree.

Finally, I would like to thank Mrs. Shella Bibb Baccus for the help in proofreading this Thesis.

CONTENTS

Page

THAI ABSTRACT iv

ENGLISH ABSTRACT v

ACKNOWLEDGEMENTS vi

CONTENTS vii

LIST OF FIGURES 1

LIST OF TABLES 3

CHAPTER 1 INTRODUCTION 4

1.1 Objectives 9

1.2 Scopes of work 9

1.3 Publications 10

CHAPTER 2 BACKGROUND AND RELATED WORKS 11

2.1 Dynamic Time Warping (DTW) 12

2.2 Derivative Dynamic Time Warping (DDTW) 13

2.3 Adaptive Feature Based Dynamic Time Warping 14

2.4 Longest Common Subsequence Similarity Measure (LCSS) 15

CHAPTER 3 PROPOSED ALGORITHM 16

3.1 Local Scaling Feature (LSF) and Local Feature Normalization (LFN) 16

3.2 Longest Common Local Scaling Feature (LCSF) 21

3.3 Local Scaling Dynamic Time Warping (LSDTW) 23

CHAPTER 4 EXPERIMENTS 29

4.1 UCR datasets 29

4.1.1 Small-size UCR datasets 29

4.1.2 Larger-size UCR datasets 34

4.2 Synthetic Data 37

4.3 Real World data and Applications 40

CHAPTER 5 CONCLUSION 43

REFERENCES 45

APPENDIX 47

Page

Local Scaling Dynamic Time Warping Origin (LSDTWO) 48

VITA 51

LIST OF FIGURES

Fig. 1-1 Incorrect scaling caused by Z-normalization in sequences containing (a) amplitude scaling, (b) time scaling (warping), (c) noise, and (d) combination of three types of variability. (e) Due to incorrect scaling, high magnitude points of the top sine wave graph cannot be matched correctly using DTW; however, it cannot skip points, and therefore matches the rest of square wave part although the distances are high. (f) Though LCSS can skip data points, the square waves of the two sequences are too different for a relatively small ε parameter to match, and cannot be correctly aligned. (g) The proposed LCSF can correctly align the sequences. 7

Fig. 3-1 LFN's mechanism is finding the proper candidates a_i and b_j, and then uses this candidates to normalize (divide) the original time series.............................17

Fig. 3-2 An illustration of potential score calculation of a_i and b_j; the left/right subsequences will be scored using LCSS such that two data points are matched if their scale is equal to a_i/b_j. 19

Fig. 3-3 The crossing of high *PS* score LSF; the alignment lines shown the high *PS* score LSF, however, the dash line is crossing some other lines. 21

Fig. 3-4 Alignment from DTW of two Z-normalized time series consisting of two subsequences: a sine wave and a square wave, which have extremely different scaling on both X and Y axes. Evidently, it produces incorrect alignments. 26

Fig. 3-5 Alignment from DDTW of two Z-normalized time series consisting of two subsequences: a sine wave and a square wave, which have extremely different scaling on both X and Y axes. Evidently, it produces incorrect alignments. 27

Fig. 3-6 Alignment from FBDTW of two Z-normalized time series consisting of two subsequences: a sine wave and a square wave, which have extremely different scaling on both X and Y axes. Evidently, it produces incorrect alignments. 27

Fig. 3-7 The proposed LCSF can align quite correctly but LCSF is based on LCSS that does not allow duplicate matching such that some data points the lower time series' sine wave, which are longer than the upper time series' sine wave, are skipped. 28

Fig. 3-8 The proposed LSDTW can correctly align these two time series pair, matching the sine wave from one sequence to another sine wave of another sequence, and matching the square wave from one sequence to another square wave of another sequence. 28

Fig. 4-1 Examples of the four classes of synthetic data 39

Fig. 4-2 The three wrist strengthening exercises for wrist injuries with sampled time series sequences showing for each class; a mobile phone is tied and taped on middle and ring fingers. 41

Fig. A-1 The comparison between a) LSDTWO and b) LSDTW 50

LIST OF TABLES

Table 4-1 Classification accuracy on small size UCR datasets (* observed as noisy dataset) 31

Table 4-2 Classification accuracy and the improvement over DTW 35

Table 4-3 Classification accuracy on synthetic datasets 39

Table 4-4 Accuracy on wrist strengthening rehabilitation exercise classification 42

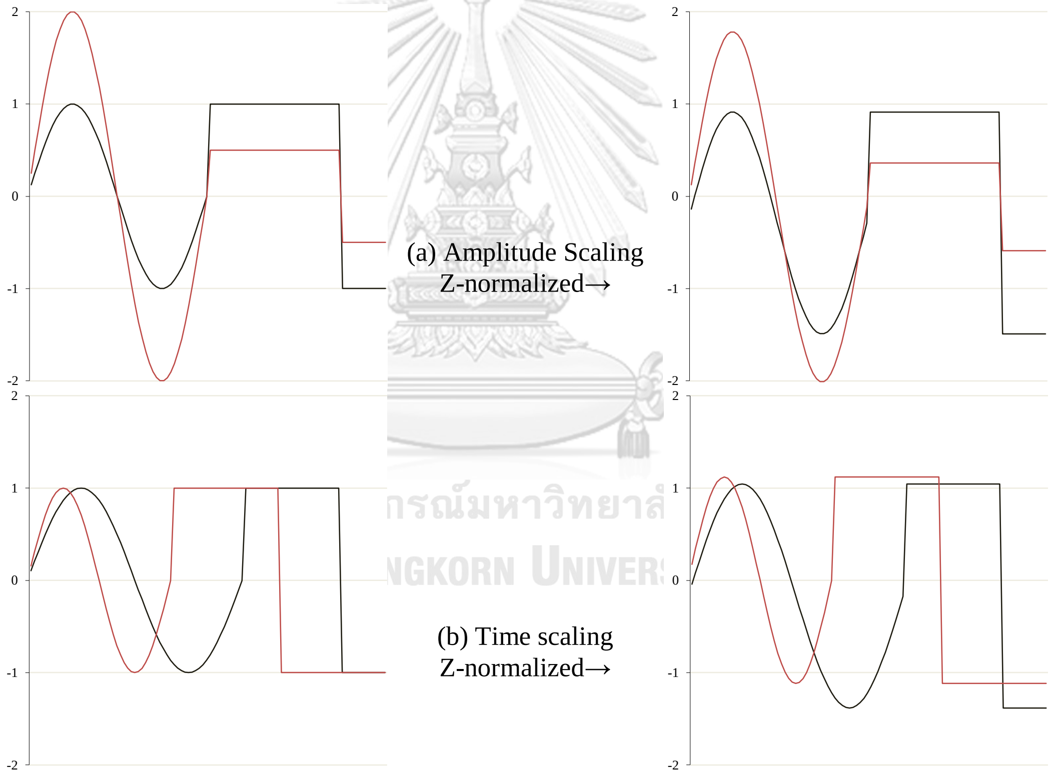
Table A-1 Preliminary experiments classification 48

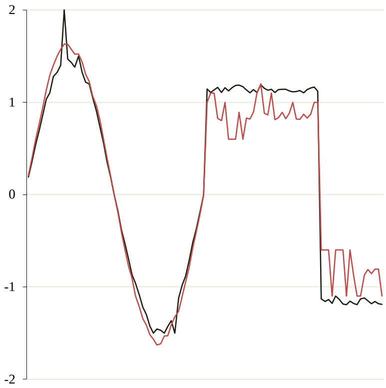
CHAPTER 1
INTRODUCTION

Dynamic Time Warping (DTW) (Sakoe & Chiba, 1978; Vlachos, Hadjieleftheriou, Gunopulos, & Keogh, 2003) and Longest Common Subsequence (LCSS) (Das, Gunopulos, & Mannila, 1997) are the two well-known similarity measures for time series data with the ability to warp on the X-axis using non-linear alignment. However, the main obstacle of these algorithms is the variability of scales in the amplitude; they work at its best when the time series pair has similar amplitude scaling. Therefore, time series normalization becomes critical as a minor adjustment of scale can actually double the error rates (Rakthanmanon et al., 2012).

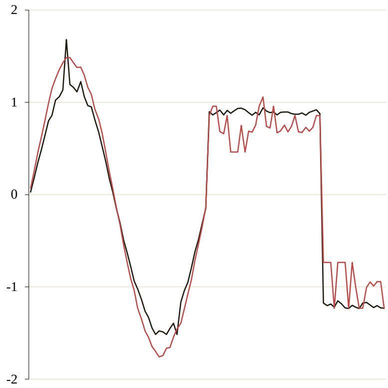
Currently, Z-normalization is one of the most widely used techniques to normalize time series data. Unfortunately, Z-normalization can produce incorrect scaling on data containing noise, missing data, outlier, subsequence scaling, and even the variability in the time axis. As demonstrated in **Fig. 1-1** (a) – (d), each of the sequences contains a sine wave followed by a square wave with different variability added to the sequences, i.e., (a) Y-axis scaling (amplitude), (b) X-axis scaling (warping), (c) noise, and (d) combination of all three; Z-normalization produces inaccurate scaling, which in turn causes inaccuracies in classification. Finding similarity between two Z-normalized sequences in **Fig. 1-1** (d) using DTW or LCSS can lead to incorrect alignment. As shown in **Fig. 1-1** (e), due to incorrect scaling, high-magnitude points of the top sine wave cannot be matched correctly through DTW, and the rest of the square wave is wrongly matched, overestimating the cumulative distances. In **Fig. 1-1** (f), on the other hand, though LCSS can skip data points, the square waves of the two sequences are too different for a relatively small ε parameter to match, and cannot be correctly aligned.

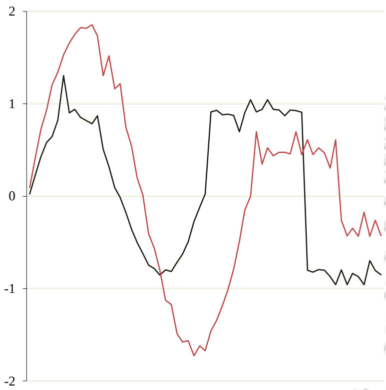
To resolve this problem, this work introduces a Local Feature Normalization (LFN) that first discovers a Local Scaling Feature (LSF) to normalize time series data more accurately, especially for time warped and noisy data. The Longest Common Local Scaling Feature (LCSF) similarity measurement is then proposed to effectively match time series sequences that contain subsequences with a variety of scales, as demonstrated in **Fig. 1-1** (g).



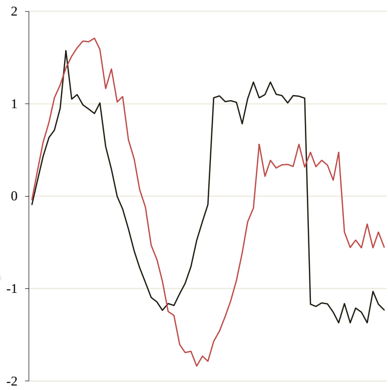


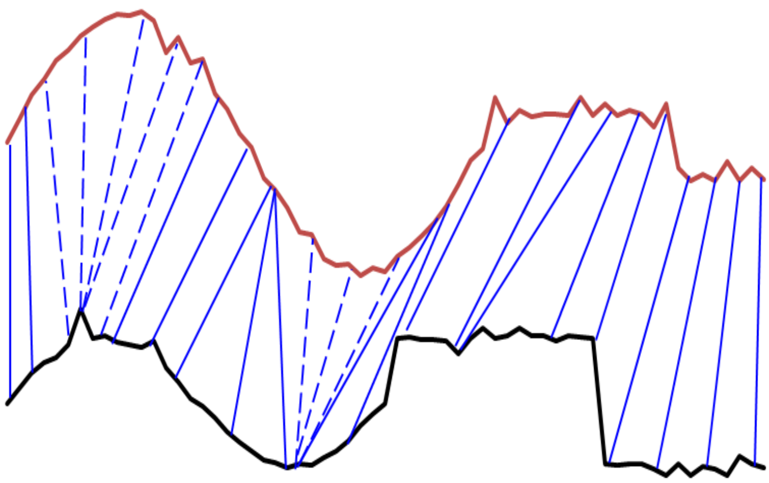
(c) Noisy
Z-normalized→



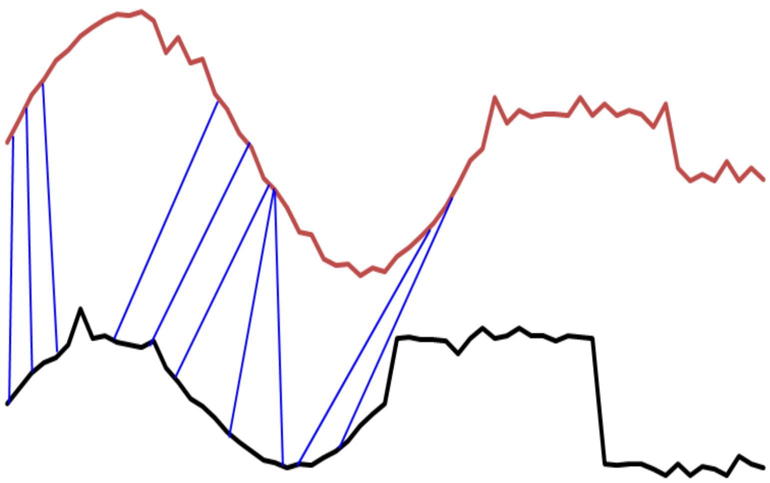


(d) Noisy/Scale/
Warped
Z-normalized→

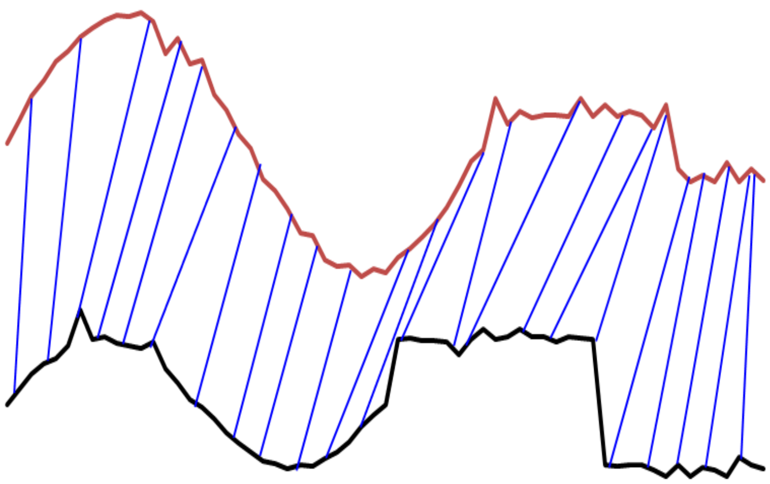




(e) DTW



(f) LCSS



(g) LCSF

Fig. 1-1 Incorrect scaling caused by Z-normalization in sequences containing (a) amplitude scaling, (b) time scaling (warping), (c) noise, and (d) combination of three types of variability. (e) Due to incorrect scaling, high magnitude points of the top sine wave graph cannot be matched correctly using DTW; however, it cannot skip points, and therefore matches the rest of square wave part although the distances are high. (f) Though LCSS can skip data points, the square waves of the two sequences are too different for a relatively small ε parameter to match, and cannot be correctly aligned. (g) The proposed LCSF can correctly align the sequences.

Other than the problem of undesirable normalization results, typical DTW distance functions that consider the raw value of each individual data point can also cause misalignment and an increase in misclassification rate. In an attempt to alleviate the problem, Derivative Dynamic Time warping (DDTW) (E. J. Keogh & Pazzani, 2001), which is performing DTW on derivatives of the data, was introduced to reduce misalignment of the classic DTW. However, DDTW still cannot align these multiple-scale subsequences correctly because the slopes of different-scale sequences are still different; for example, the derivative of a function $f(x)$ is merely half of the derivative of $2f(x)$. Adaptive Feature Based Dynamic Time Warping (Xie & Wiltgen, 2010) later introduced local and global features of each data point and used the distance between the features instead of the original distance, resulting in better alignment and better classification. However, this feature distance still cannot handle multiple-scale subsequences very well.

Finally, this research proposes a Local Scaling Dynamic Time Warping (LSDTW), which uses a new distance function especially for handling **multiple-subsequence-with-multiple-scale** time series. The LSDTW is the DTW-based extension of the proposed Local Feature Normalization (LFN) and Longest Common Local Scaling Feature (LCSF) (Chonbodeechalermroong & Ratanamahatana, 2018). Longest Common Local Scaling Feature (LCSF) was proposed based on the idea of Longest Common Subsequence (LCSS), which allows non-linear alignment with an ability to skip data points. This novel Local Scaling Dynamic Time Warping (LSDTW), on the other hand, is based on DTW with much fewer parameters and with much smaller time complexity than Longest Common Local Scaling Feature (LCSF). In particular, the proposed Local Scaling Dynamic Time Warping (LSDTW) still has about the same time complexity as the original DTW.

1.1 Objectives

This research focuses on inventing new algorithms: normalization, similarity measurement and distance measurement that could handle noisy time series and time series containing multiple subsequences with multiple scales on both axes.

- To propose a novel normalization technique that is robust to noisy/warped time series data.
- To propose a novel time series similarity measurement that can handle matching multiple-subsequence-with-multiple-scale time series.

1.2 Scopes of work

- This work evaluates LFN on small-size UCR datasets that are observed as noisy comparing to Z-normalized data with DTW, DTW with global constraint and LCSS.
- This work evaluates LCSF on large-size UCR datasets, synthetic datasets and on the wrist strengthening rehabilitation exercises classification dataset.
- This work evaluates LSDTW on small-size and large-size UCR datasets, synthetic dataset and the wrist strengthening rehabilitation exercises classification dataset.

1.3 Publications

Journal

- Chonbodeechalermroong, A.,& Hewett, R., Towards Visualizing Big Data with Large-Scale Edge Constraint Graph Drawing, Big Data Research. 10 (2017) 21–32. doi:10.1016/j.bdr.2017.10.001.

Conferences

- Chonbodeechalermroong, A.,& Ratanamahatana, C.A., Robust Scale-Invariant Normalization and Similarity Measurement for Time Series Data, in: A. Sieminski, A. Kozierkiewicz, M. Nunez, Q.T. Ha (Eds.), Modern Approaches for Intelligent Information and Database Systems, Springer International Publishing, Cham, 2018: pp. 149–160. doi:10.1007/978-3-319-76081-0_13.
- Chonbodeechalermroong, A.,& Chalidabhongse, T. H., Dynamic contour matching for hand gesture recognition from monocular image, in: 2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2015: pp. 47–51. doi:10.1109/JCSSE.2015.7219768.

CHAPTER 2
BACKGROUND AND RELATED WORKS

Regardless of extended usage of DTW and LCSS in the past few decades, many researchers may not realize that DTW (Sakoe & Chiba, 1978; Vlachos et al., 2003) and LCSS (Das et al., 1997) only work at its best when the data have similar amplitude scaling. Normalization has played a big role in trying to resolve this scaling issue (Rakthanmanon et al., 2012). However, as shown in **Fig. 1-1**, the most commonly used Z-normalization technique for time series data can still cause DTW and LCSS to produce inaccurate results on data with such variability.

Nowadays, time series have been used in many applications such as gene expression (Bar-Joseph, Gerber, Gifford, Jaakkola, & Simon, 2002), body movement recognition from video (Gavrila & Davis, 1995), prosthesis control and rehabilitation (Crouch & Huang, 2016; Yun et al., 2017) and hand gesture recognition (X. Chen et al., 2007). Most of the applications obtain data from sensors such as electromyography (EMG) or accelerometers, which are noisy and contain various types of scaling, and sometimes a single sensed time series sequence can have many subsequences with a variety of scales.

To match these varied-scale subsequences time series using the traditional DTW or LCSS, the training set needs to contain all possible combinations of scaling of all subsequences, which may not be possible to obtain. To the best of my knowledge, no current normalization technique is specially designed for such variability in the data. The proposed Local Feature Normalization (LFN) and Longest Common Local Scaling Feature (LCSF) can effectively solve this problem with only a limited amount of training data. The background of DTW and some of their variations and also LCSS will be briefly explained in this section.

2.1 Dynamic Time Warping (DTW)

Given two Time Series A and B with length n and m, $A = \{a_1, a_2, a_3, \ldots, a_n\}$, $B = \{b_1, b_2, b_3, \ldots, b_m\}$ where a_i and b_j are real numbers, DTW distance of A and B measuring the distance between the two time series is:

$$\mathrm{DTW}(A, B) = f(n, m) = \begin{cases} 0 & \text{if } n=0 \text{ and } m=0 \\ \infty & \text{if } n=0 \text{ and } m\neq 0 \text{ or } n\neq 0 \text{ and } m=0 \\ d(a_n, b_m) + \min \begin{cases} f(n, m) \\ f(n, m\text{-}1) \\ f(n\text{-}1, m) \end{cases} & \text{otherwise} \end{cases} \tag{1}$$

where $d(a_i, b_j)$ is the distance between a_i and b_j defined as $d(a_i, b_j) = (a_i - b_j)^2$. DTW has many variations such as a global constraint (Sakoe & Chiba, 1978) that constrains the maximum warping distance such that a_i will never match with b_j if $|i - j| > l$; l is a constraint window size.

To speed up time series searching in large datasets, lower bounding techniques, LB_Kim (Kim, Park, & Chu, 2001) and LB_Keogh (E. Keogh & Ratanamahatana, 2005) are proposed to help estimate the lower bound distance for pruning some non-worthy data instances.

In addition to lower bounding, another approach to speed up dynamic time warping calculation can be done through "early abandoning" (Rakthanmanon et al., 2012). As DTW is typically applied in K-nearest neighbor classification, the smallest distance so far, ƫ, needs to be maintained to finally obtain a nearest sequence to the given query. However, to calculate the cumulative distance $f(i, j)$, i from 1 to n and j from 1 to m for each i, it is compulsory that $f(i, j) \geq f(i, j\text{-}1)$. Therefore, the DTW calculation can safely be abandoned when $\forall j$ $f(i,j) >$ ţ since the cumulative distance from this point further will never improve (lower) the best distance so far (ţ), and hence, this candidate will never be the query's nearest neighbor. This early abandoning technique is implemented in this research's experiments as well to achieve some speedup.

UCR-DTW (Rakthanmanon et al., 2012) combines many lower bounding approaches and early abandoning techniques to significantly speed up DTW. However, UCR-DTW also relies on Z-normalization.

Due to the ability to non-linearly align time series together, Dynamic Time Warping has been popular in many pattern recognition applications, such as signature recognition (Faundez-Zanuy, 2007), speech recognition (Godin & Lockwood, 1989), shape matching (Marzal & Palazón, 2005), ECG pattern recognition (Huang & Kinsner, 2002), Electronic Health Records (EHRs) similarity measurement (Huang & Kinsner, 2002), among many others.

2.2 Derivative Dynamic Time Warping (DDTW)

Derivative Dynamic Time Warping (DDTW) (E. J. Keogh & Pazzani, 2001) has been proposed to help improve the DTW alignment between a time series pair by reducing spurious alignment and "singularities" problem (E. J. Keogh & Pazzani, 2001). DDTW converts the original time series $\{a_1, a_2, a_{3,}\ldots,a_n\}$, into its derivative $\{a'_1, a'_2, a'_{3,}\ldots,a'_{n-2}\}$, using equation (2).

$$a'_j = ((a_i - a_{i-1}) + (a_{i+1} - a_{i-1})/2)/2;\ j = i-1 \quad (2)$$

Note that the derivative form is the average slope of each data point. So, the first and the last data points will be omitted as they do not have the definition of their slopes. Regardless of its ability to improve the alignment, DDTW still has a slight problem in that it is sensitive to noise, as little noise can significantly change the slope of the graph because each derivative data point considers only three points. Therefore, smoothing could become very important.

2.3 Adaptive Feature Based Dynamic Time Warping

To improve robustness to noise of DDTW, Adaptive Featured Based Dynamic Time Warping (AFBDTW) (Xie & Wiltgen, 2010) has been proposed. It is quite similar to DDTW, but more points are used in the averaging process, representing the global feature of the data. Each data point a_i has a local feature $f_{local}(a_i)$ and a global feature $f_{global}(a_i)$. The local feature $f_{local}(a_i) = (a_i - a_{i-1}, a_i - a_{i+1})$, which is similar to the slope in DDTW. The global feature $f_{global}(a_i) = (a_i - (a_1 + a_2 + \ldots + a_{i-1})/(i-1), a_i - (a_{i+1} + a_{i+2} + \ldots + a_n)/(n-i))$; the global feature of a data point is the difference between that point and the average of all other points before and after that point. The distance function of any two data points a_i and b_j is as follows:

$$distance(a_i, b_j) = w1 * dist(f_{local}(a_i), f_{local}(b_j)) + w2 * dist(f_{global}(a_i), f_{global}(b_j)) \qquad (3)$$

where *dist* is the distance function, proposed in two alternatives: Manhattan distance (AFBDTW1) and Euclidian distance (AFBDTW2); $w1$ and $w2$ are the weights proposed in (Xie & Wiltgen, 2010).

However, this AFBDTW still could not handle multiple-subsequence-with-multiple-scale time series data because the local features (slopes) of any two subsequences in different scales will be different, and the global features (average) of two subsequences in different scales would also be different.

2.4 Longest Common Subsequence Similarity Measure (LCSS)

Given two Time Series A and B with length n and m, $A = \{a_1, a_2, a_3, \ldots, a_n\}$, $B = \{b_1, b_2, b_3, \ldots, b_m\}$ where a_i and b_j are real numbers, LCSS score of A and B measuring the similarity between the two time series is:

$$LCSS(A, B) = f(n, m) = \begin{cases} 0 & \text{if } n=0 \text{ or } m=0 \\ \max \begin{cases} s(a_n, b_m) + f(n-1, m-1) \\ f(n, m-1) \\ f(n-1, m) \end{cases} & \text{otherwise} \end{cases} \quad (4)$$

where $s(a_i, b_j)$ is the similarity between a_i and b_j defined as $s(a_i, b_j) = 1$ if $|a_i - b_j| < \varepsilon$ and $|i - j| \leq l$, and $s(a_i, b_j) = 0$ otherwise; ε is a given small arbitrary value; l is a constraint window size. This discrete similarity has a drawback that if ε is too small, many points are considered as noise. If ε is too large, too many points can match with the others such that LCSS may produce just one-to-one matching (no warping/skipping).

LCSS' main advantage is its ability to skip noisy data points while DTW has to match every single data point. However, DTW has fewer parameters, making it more popular and practical than LCSS.

CHAPTER 3
PROPOSED ALGORITHM

This chapter explains the proposed Local Scaling Feature (LSF), which is then utilized in the proposed Local Feature Normalization (LFN) and Longest Common Local Scaling Feature (LCSF) similarity measure. Then, the Local Scaling Dynamic Time Warping (LSDTW) Distance measure, the DTW-based extension of Longest Common Local Scaling Feature (LCSF), will be explained.

3.1 Local Scaling Feature (LSF) and Local Feature Normalization (LFN)

The proposed Local Scaling Feature (LSF) is based on LCSS concept, as this work holds the assumption that the data is particularly noisy and LCSS can skip some noisy data points. However, Local Scaling Feature (LSF) still works well on non-noisy data.

Given two time series A and B with length n and m, $A = \{a_1, a_2, a_3,\ldots,a_n\}$, $B = \{b_1, b_2, b_3,\ldots, b_m\}$ where a_i and b_j are real numbers, one could normalize the time series A with some value a_i and normalize B with some value b_j. If the two time series are similar, but having different scaling in the Y-axis, then there must exist $a_i \approx b_j$ if A and B are in the same scaling. Hence, if one normalizes A by a_i and B by b_j, one will get the correct scale of A and B called $A' = A / a_i$ and $B' = B / b_j$.The illustration, the proposed normalization mechanism, is shown in **Fig. 3-1**.

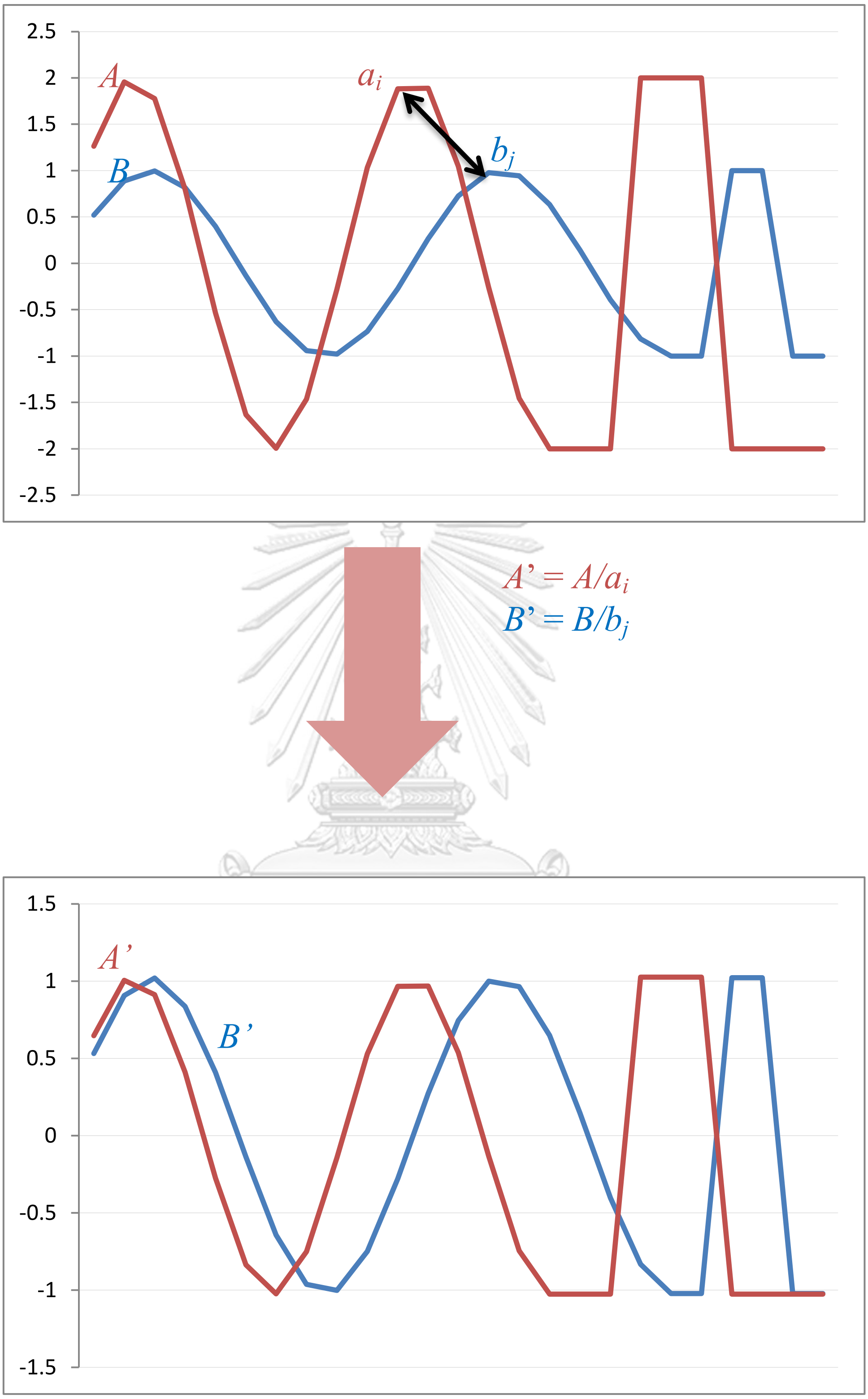


Fig. 3-1 LFN's mechanism is finding the proper candidates a_i and b_j, and then uses this candidates to normalize (divide) the original time series.

In order to find the correct *A'* and *B'*, one might compare all possible combinations of a_i and b_j to normalize and select the best LCSS/DTW score among all matched normalized data; however, the time complexity is excessively high: $O(n^2m^2)$, *nm* for trying all possible *A'* and *B'*, and another *nm* for doing LCSS/DTW on each of *A'* and *B'*. However, this huge amount of combination can be reduced by finding some potential candidates called **Local Scaling Features** (LSFs).

If a_i and b_j are potential candidates, then some neighboring points of a_i should match with some neighboring points of b_i. LCSS is then used to match the left subsequences of length *w*: $LA = \{a_{i-w},\ldots, a_{i-1}\}$ with $LB = \{b_{j-w},\ldots, b_{j-1}\}$ as the Left Score (*LS*), and the right subsequences of length *w*: $RA = \{a_{i+1},\ldots, a_{i+w}\}$ with $RB = \{b_{j+1},\ldots, b_{j+w}\}$ as the Right Score (*RS*), then the potential score $PS = LS + RS$; *PS* measures the local similarity. Then the best *c* candidates (*c* highest *PS* scores) are selected as the LSFs. Because the data are assumed to be noisy, one should not use Z-normalization in calculation of *LS* and *RS*. Instead, this local LCSS uses this newly proposed special similarity function that requires no normalization.

To match the left subsequences *LA* and *LB* that is assumed to be in different scales and a_i and b_j are assumed to be matched, there must be indices *x* and *y* that LA_x/LB_y is equal to a_i/b_j where $LA_x \in LA$ and $LB_y \in LB$ as well as matching *RA* and *RB*, as illustrated in **Fig. 3-2.**

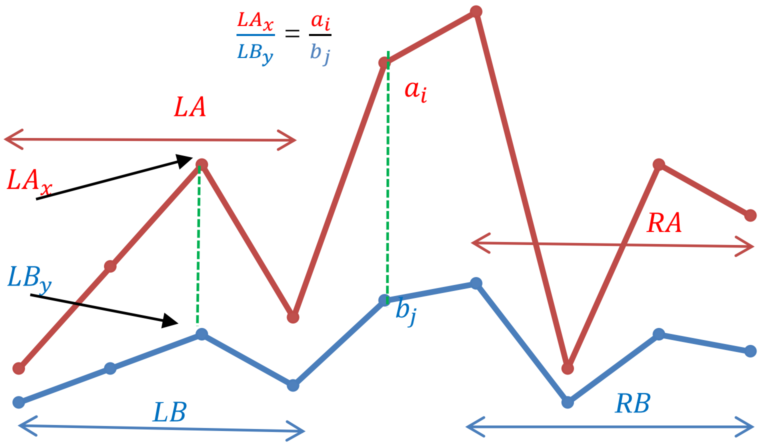


Fig. 3-2 An illustration of potential score calculation of a_i and b_j; the left/right subsequences will be scored using LCSS such that two data points are matched if their scale is equal to a_i/b_j.

However, another important problem is when a_i and b_j are much larger than LA_x and LB_y. There is a fact that if LA_x and LB_y are aligned, then $(a_i - LA_x)/(b_j - LB_y)$ is also equal to a_i/b_j. For example, given $a_i = 10$, $b_j = 20$, $LA_x = 0.2$, $LB_y = 0.1$, one could see that $a_i/b_j = 0.5$ while $LA_x/LB_y = 2$, but $(a_i - LA_x)/(b_j - LB_y) = 0.49$. This too little value of LA_x and LB_y comparing to a_i and b_j can be seen as noise. Therefore, this method selects the closest value to a_i/b_j among LA_x/LB_y and $(a_i - LA_x)/(b_j - LB_y)$. According to this principle, the proposed similarity function given the candidate a_i and b_j is as follows:

$$s2(X, Y) = 1 - \frac{min\left\{\begin{array}{l}\left|\frac{X}{Y} - \frac{a_i}{b_j}\right| \\ \left|\frac{a_i - X}{b_j - Y} - \frac{a_i}{b_j}\right|\end{array}\right.}{\varepsilon'\left|\frac{a_i}{b_j}\right|} \tag{5}$$

where $X \in LA$ (or LB); $Y \in RA$ (or RB); ε' is a parameter similar to ε in the original similarity function.

The best c candidates are used to normalize the time series; then LCSS is applied to these c normalized series. In this LCSS part, to distinguish finer dissimilarity that helps solve the one-to-one matching problem of the original LCSS with large ε, this method prefers the continuous similarity score instead of the old binary score. Therefore, equation (4) in section 2.4 will be replaced by another similarity function, $s3(a_i, b_j) = 1 - |a_i - b_j|/\varepsilon$. The idea behind this function is that LCSS will not match a_i to b_j when the difference is larger than ε because $s3(a_i, b_j)$ is less than 0; however, if the difference is smaller than ε, the score will be a continuous value from 0 to 1. Similar idea also applies to $s2$ in Local Scaling Feature (LSF). To allow maximum flexibility of matching, window constraint is not applied.

The sequence with the best LCSS score is chosen to be the output of the **Local Feature Normalization** (**LFN**). Note that the flipping candidates ($a_i/b_j < 0$) need not be calculated to save some time and improve overall accuracy as it rarely happens to be in the same class.

The time complexity of finding the potential candidates is $O(w^2mn)$; w^2 is the complexity of using LCSS to calculate each candidate's *PS*; $O(mn)$ is for finding all possible candidates. Applying LCSS on the normalized series c times uses $O(cmn)$. Therefore, the overall complexity is $O(w^2mn + cmn)$. Practically, w and c are relatively small comparing to n and m such that the potential complexity could only be $O(nm)$

3.2 Longest Common Local Scaling Feature (LCSF)

As shown in **Fig. 1-1**(a), some time series may contain several subsequences, each of which may also have different scaling. Local Scaling Feature (LSF) could be utilized to handle multiple scaling in subsequences.

To Find a Local Scaling Feature (LSF), LCSS is performed on both sides of the candidate subsequence; each side has a length of w so that the maximum possible value for PS is $2w$. Given a threshold t, a candidate whose PS score is larger than or equal to $2wt$ is called the potential candidate; t is the ratio of the maximum possible value of PS; $t \in (0, 1]$.

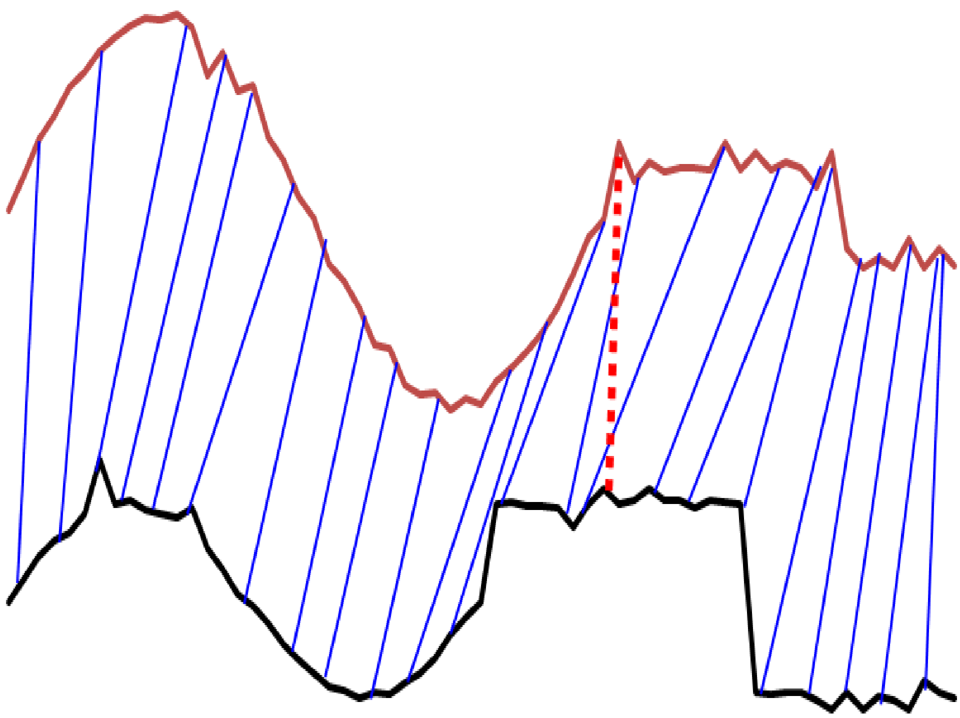


Fig. 3-3 The crossing of high *PS* score LSF; the alignment lines shown the high *PS* score LSF, however, the dash line is crossing some other lines.

After finding the best c potential candidates with $PS \geq 2wt$, each candidate consists of the indices i, j (denoting the position in A and B) and the PS score: (i, j, PS); however, there can be two or more candidates, which are crossing, for example, (i, j, $PS1$) and (u, v, $PS2$): $i > u$ and $j < v$. This crossing could occur because either each potential candidate is discovered locally, or the threshold t is too small. An example of crossing is illustrated in **Fig. 3-3**.

In order to find a proper longest LSF matching, this method sorts the potential candidates in an ascending order of i, together with a descending order of j, then find the longest increasing subsequence based on j on the sorted potential candidates. The output of the longest increasing subsequence is the LCSF score. For example, given the potential candidates (1, 2, 0.5), (1, 3, 0.6), (1, 4, 0.9), (2, 3, 0.6) and (3, 4, 0.6), after sorting, the result will be (1, 4, 0.9), (1, 3, 0.6), (1, 2, 0.5), (2, 3, 0.6) and (3, 4, 0.6). If one looks only on the list of j, one will see {4, 3, 2, 3, 4}. Then the proper longest increasing subsequences on this set will be {2, 3, 4}, where $|\{2, 3, 4\}| = 3$ is the output of the Longest Common Local Scaling Feature (LCSF) similarity score.

Discovering all potential candidates, the $O(w^2mn)$ time complexity is needed, as described in the previous section. Sorting c candidates needs $O(c\log(c))$, and finding the longest increasing subsequences using the efficient algorithm (Fredman, 1975) requires $O(c\log(c))$ time complexity. Hence, the overall complexity is $O(w^2mn + clog(c))$. Longest Common Local Scaling Feature (LCSF) has four parameters: w, ε', t and c to tune. However, to make sure that no potential candidates are missed, the parameter c can be ignored and accept all potential candidates ($PS \geq 2wt$). The pseudocode of Local Scaling Feature (LSF), Local Feature Normalization (LFN) and Longest Common Local Scaling Feature (LCSF) are as follows.

```
LSF(A,B,w, t, c)
pcs ={} #set of potential candidates
for i in |A|:
  for j in |B|:
    LS  = lcss'(A[i-w,i-1], B[j-w,j-1], A[i], B[i])
    RS = lcss'(A[i+1,i+w], B[j+1,j+w] , A[i], B[i])
    PS = LS + RS
    if PS >= 2wt:
      if  |pcs| < c:
        pcs.add((i,j,ps))
      else if PS  > pcs.getMinPS():
        pcs.removeMin()
        pcs.add((i,j,ps))
return pcs
#lcss' is LCSS using s2 as a similarity function, given A[i] and B[i]
```

```
LFN(A, B, w)
pcs = LSF(A, B, w, 0, c)
max= 0
for f in pcs:
  A' = A/A[f.i]
  B' = B/B[f.j]
  l = lcss''(A', B')
  if l > max:
    max = l
    bestA',bestB' = A',B'
return max,bestA',bestB'
# lcss'' is LCSS using s3 as a similarity function
```

```
LCSF(A, B, w, t, c)
pcs = LSF(A, B, w, t, c) #c is set to ∞ in this resrach experiment
sort(pcs) in ascending order of i and in descending order of j on equal i
return longest_increasing_subsequence(pcs) based on j
```

3.3 Local Scaling Dynamic Time Warping (LSDTW)

A Local Scaling Dynamic Time Warping (LSDTW) Distance measure is proposed here to resolve the problem of multiple scales in multiple subsequences with relatively low time complexity compared with the state-of-the-art approach. If a time series sequence has multiple subsequences, where each subsequence does have different scales, normalizing the whole time series sequence will not guarantee correct scaling of the result. However, if one normalizes those subsequences locally, most likely one will have correct scaling results. However, segmentation of time series into proper subsequences is still problematic and becoming a challenging task.

Instead of performing time series segmentation, this paper proposes an easy but powerful approach using only values of the data points from the time series pair. More specifically, the distance between a data point a_i in A and another data point b_j in B will no longer be only the difference in their original values. Instead, the distance will cover the distance between the normalized local subsequences around a_i and the normalized local subsequences around b_j.

Given the parameter w, all of the local subsequences $\{a_{i-w},\ldots a_{i+w}\}$ and $\{b_{j-w},\ldots,b_{j+w}\}$—$w$ points before and after a_i and b_j, respectively, will be normalized using 0-1 normalization. Thus, every normalized sequence will have a minimum value of 0 and a maximum value of 1. If every point in the subsequence is identical (min = max), every point will be set to 0.5 by default. This local normalization does not use Z-normalization here due to the problems mentioned in **Fig. 1-1**. It is important to note that this method only normalizes a copy of the subsequence, not the original one. For a data point a_i, there are i-1 points to its left and n-i points to its right, and there will not be enough points on the left or the right side when $i <= w$ or $i > n - w$. In other words, the largest number of possible data points on the left would be $wl = min(w, i\text{-}1, j\text{-}1)$, and the largest number of possible data points on the right would be $wr = min(w, n\text{-}i, m\text{-}j)$. Therefore, the subsequences $\{a_{i-wl},\ldots a_{i+wr}\}$ and $\{b_{j-wl},\ldots,b_{j+wr}\}$ with the length of $wl + 1 + wr$ will be used instead.

Next, this method defines $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as a 0-1 normalized $\{a_{i-wl},\ldots a_{i+wr}\}$ sequences and a 0-1 normalized $\{b_{j-wl},\ldots,b_{j+wr}\}$ sequence, respectively. The dissimilarity between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is robust to scale variance since $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are already locally normalized such that their squared Euclidean distance, $(\boldsymbol{\alpha} - \boldsymbol{\beta})^2$, can provide much better DTW alignment for **multiple-subsequence-with-multiple-scale time series data**. However, even though the distance $(\boldsymbol{\alpha} - \boldsymbol{\beta})^2$ can provide scale invariant alignment, my preliminary experiments discovered that it is too flexible. Therefore, this thesis proposes a function that involves original data points into its calculation (with relatively lower priority), while maintaining its scale-robust alignment feature of the local normalized distance. However, since wl and wr may vary, the squared distance has to be normalized (divided) by $wl + 1 + wr$. Hence, my proposed distance function is summarized as follows (6).

$$
\begin{aligned}
& wl = min(w, i\text{-}1, j\text{-}1) \\
& wr = min(w, n\text{-}i, m\text{-}j) \\
& \boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_{wl+wr+1}\} = 0\_1\_normalize(\{a_{i-wl}, \ldots, a_{i+wr}\}) \\
& \boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots, \beta_{wl+wr+1}\} = 0\_1\_normalize(\{b_{j-wl}, \ldots, b_{j+wr}\}) \qquad (6) \\
& \alpha_{wl+1} = a_i \\
& \beta_{wl+1} = b_j \\
& LSdist(a_i, b_j) = (\boldsymbol{\alpha} - \boldsymbol{\beta})^2/(wl + 1 + wr)
\end{aligned}
$$

The impact of a single original point to other points in a local subsequence implicitly equals to the original DTW distance with just the weight of $1/(wl + 1 + wr)$, while the rest of the local subsequence implicitly has the weight of $(wl + wr)/(wl + 1 + wr)$. This research calls a DTW distance measure that uses my proposed *LSdist* distance function (6) a Local Scaling Dynamic Time Warping (**LSDTW)**. Note that when $w = 0$, LSDTW simply becomes the original DTW.

The time complexity of the proposed Local Scaling Dynamic Time Warping (LSDTW) is $O(wmn)$—$O(mn)$ comes from the DTW's complexity, and $O(w)$ is from my proposed distance function between two data points. If w is relatively small or is a (small) constant (which usually are the cases), the complexity will eventually be $O(mn)$, the same as the original DTW's complexity.

To calculate the dissimilarity between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, one may use other distance measures such as original DTW or LCSS. However, when w is relatively small such that the alignment will not warp that much, using a squared distance or Euclidian distance could give excellent results with only very small overheads comparing to DTW or LCSS. In addition, if the leftmost point of one time series sequence is matched with the rightmost point of another time series sequence, *wl* and *wr* will be smaller than *w*, then this method will simply set the distance to infinity to bypass unnecessary computation as this scenario will be less likely to happen in reality.

The alignment results from matching **multiple-subsequence-with-multiple-scale** time series data, comparing among the classic DTW, DDTW, Feature Based Dynamic Time Warping (FBDTW), the proposed Longest Common Local Scaling Feature (LCSF) and the proposed Local Scaling Dynamic Time Warping (LSDTW) are shown in **Fig. 3-4 - Fig. 3-8**

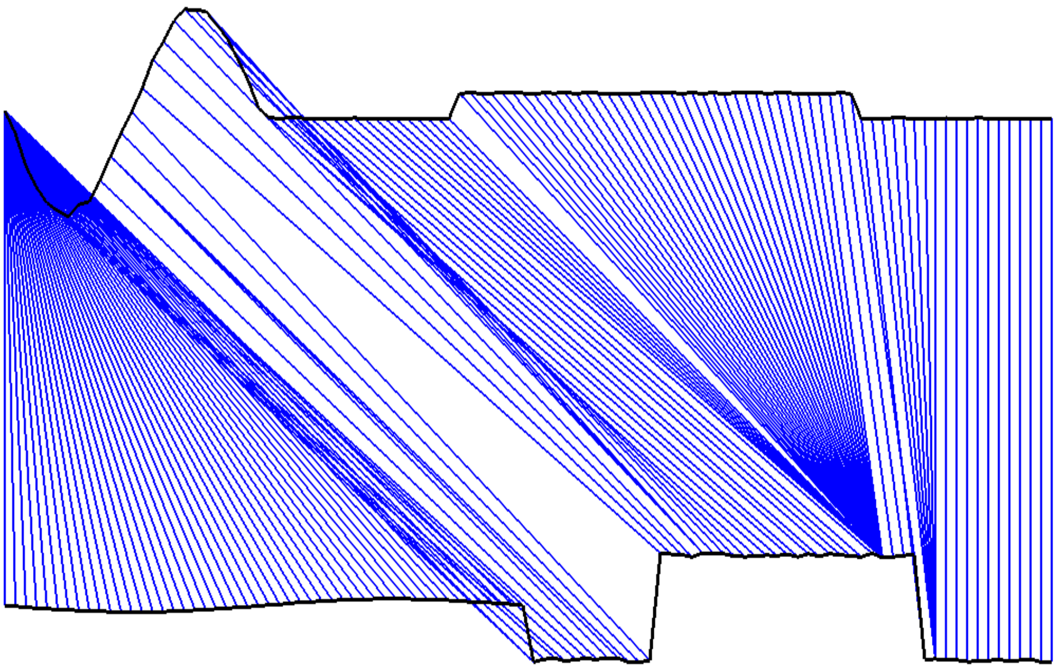


Fig. 3-4 Alignment from DTW of two Z-normalized time series consisting of two subsequences: a sine wave and a square wave, which have extremely different scaling on both X and Y axes. Evidently, it produces incorrect alignments.

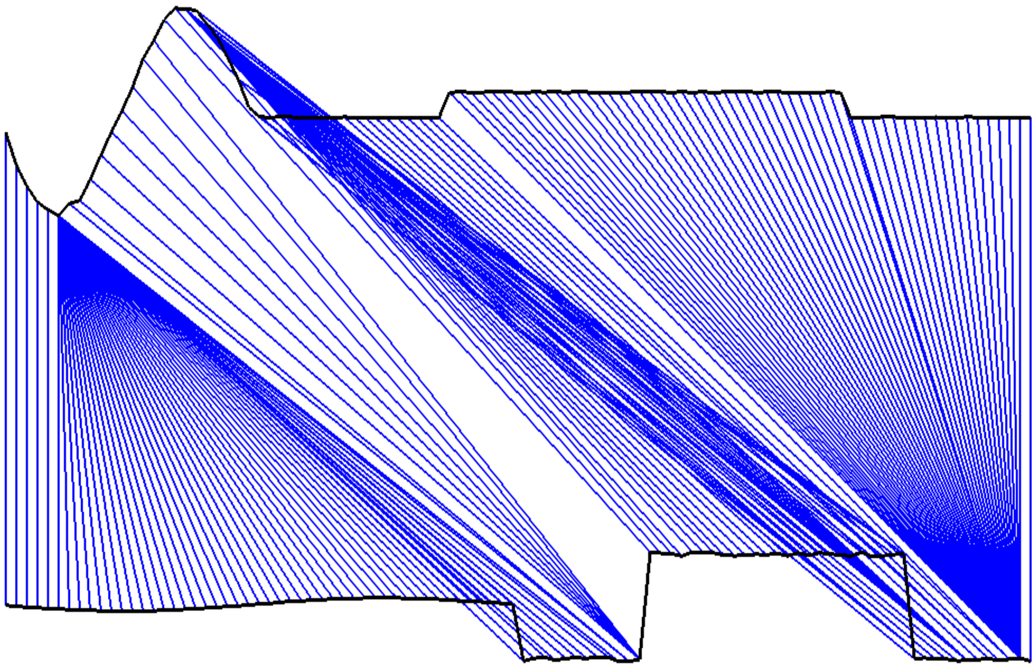


Fig. 3-5 Alignment from DDTW of two Z-normalized time series consisting of two subsequences: a sine wave and a square wave, which have extremely different scaling on both X and Y axes. Evidently, it produces incorrect alignments.

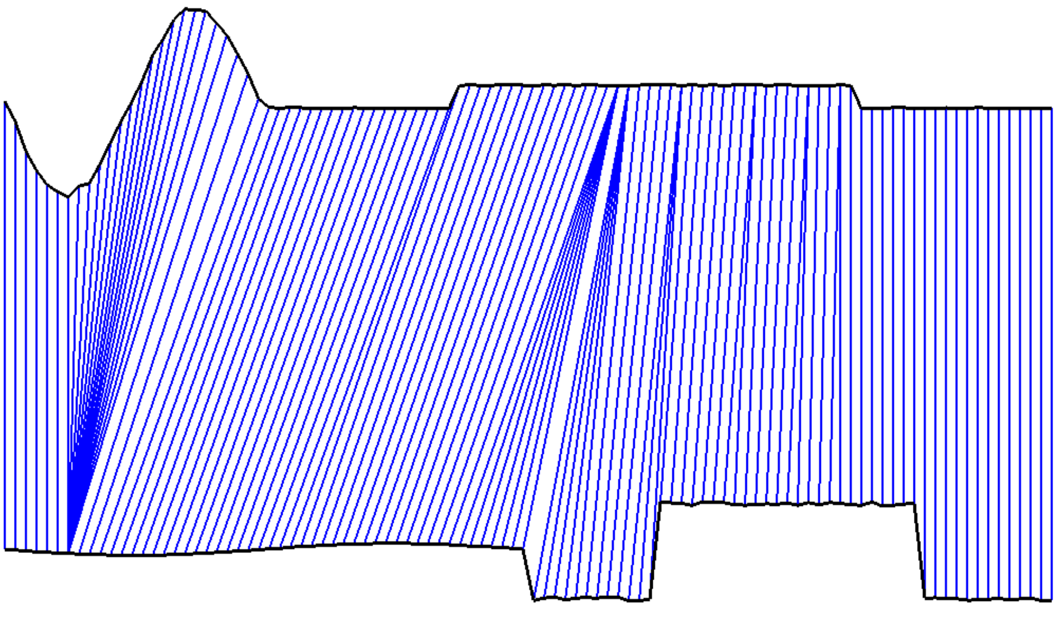


Fig. 3-6 Alignment from FBDTW of two Z-normalized time series consisting of two subsequences: a sine wave and a square wave, which have extremely different scaling on both X and Y axes. Evidently, it produces incorrect alignments.

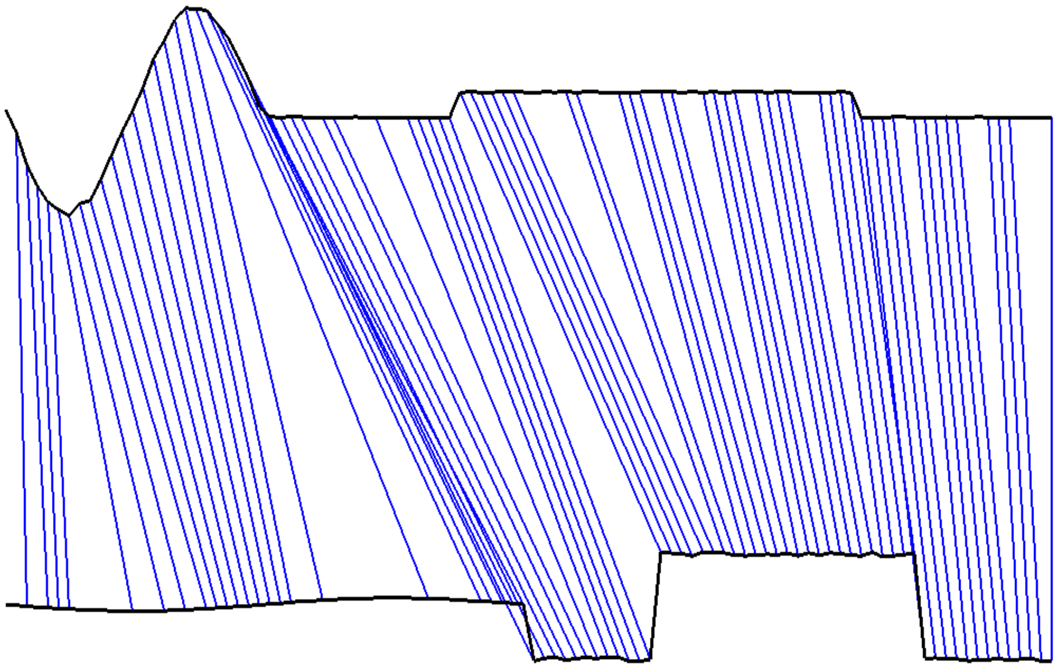


Fig. 3-7 The proposed LCSF can align quite correctly but LCSF is based on LCSS that does not allow duplicate matching such that some data points the lower time series' sine wave, which are longer than the upper time series' sine wave, are skipped.

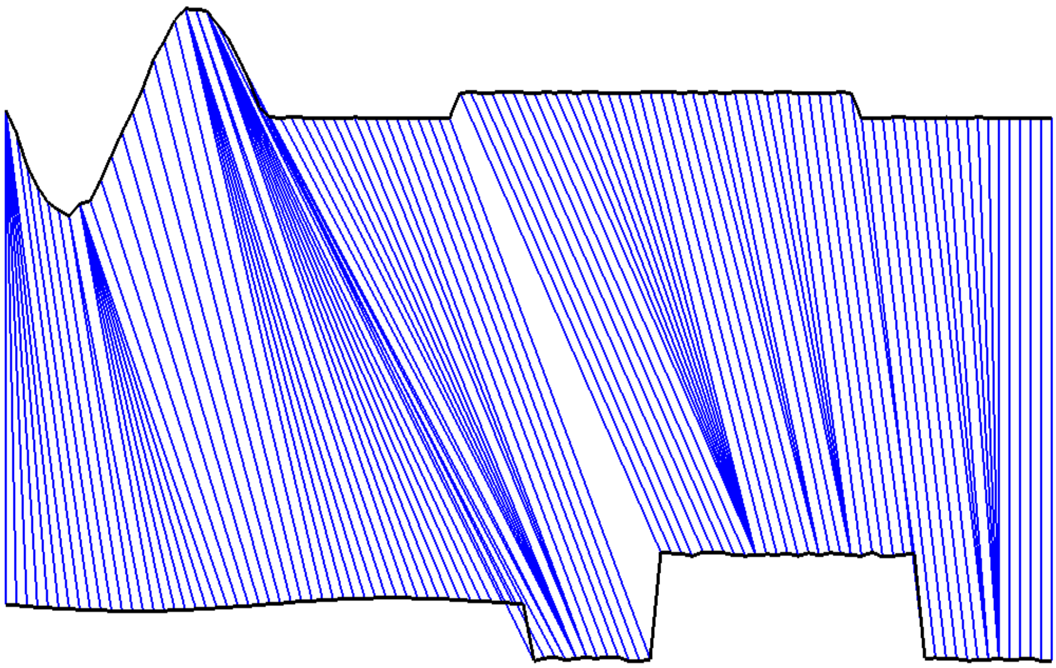


Fig. 3-8 The proposed LSDTW can correctly align these two time series pair, matching the sine wave from one sequence to another sine wave of another sequence, and matching the square wave from one sequence to another square wave of another sequence.

CHAPTER 4
EXPERIMENTS

UCR's public datasets (Y. Chen et al., 2015), synthetic datasets and Real World Applications data were used to evaluate Local Feature Normalization (LFN), Longest Common Local Scaling Feature (LCSF) and Local Scaling Dynamic Time Warping (LSDTW).

4.1 UCR datasets

Local Feature Normalization (LFN) has more parameters and higher time complexity than Local Scaling Dynamic Time Warping (LSDTW). Therefore the Local Feature Normalization (LFN) will be evaluated in the smaller size dataset: fewer instances, short time series length. While Local Scaling Dynamic Time Warping (LSDTW) will additionally be evaluated on larger datasets: more instances, longer length time series.

4.1.1 Small-size UCR datasets

The performance of Local Feature Normalization (LFN), Longest Common Local Scaling Feature (LCSF) and Local Scaling Dynamic Time Warping (LSDTW) were evaluated using 1-nearest neighbor (1-NN) classification (Peterson, 2009) on 21 UCR's public datasets (Y. Chen et al., 2015).

Note that every dataset on this archive is already Z-normalized, labeled, and split into training and test sets. The datasets are selected based on their high-noise and relatively small size criteria.

To find an optimal set of parameters w, ε' and ε in LFN, the experiments use grid search on a given training set. Leave-one-out cross-validation is used to calculate the accuracy for each parameter setup. The highest-accuracy setup is used to evaluate the test set.

To avoid a small floating point inaccuracy problem, this experiment defines integers $e' = 1/\varepsilon'$ and $e = 1/\varepsilon$, then the experiments iterate the multiplier $e(e')$ instead of the divider $\varepsilon(\varepsilon')$. Based on my empirical results, e and e' are chosen from a set {1, 2, 3, 6, 10}, w from a set [4, 10], and $c = 3$. The higher c value gives more opportunity for better accuracy, but it consumes more time; however, $c = 3$ empirically appears to be the smallest c value that does not sacrifice much accuracy.

In **Table 4-1**, the experiment compares classification accuracy of the proposed Local Feature Normalization (LFN), Longest Common Local Scaling Feature (LCSF) and Local Scaling Dynamic Time Warping (LSDTW) with well-known similarity measures: 1) Euclidean Distance, 2) DTW with the best global constraint window reported on the UCR repository, 3) DTW with no global constraint, 4) the original LCSS, and 5) the LCSS using $s3$ similarity function with the same Local Feature Normalization (LFN)'s ε parameter that shows the comparison between Z-normalization and my normalization.

To tune for the w parameter of Local Scaling Dynamic Time Warping (LSDTW), grid search was used in the training set to find the w parameter; w was selected from {1, 1%, 2%,…, 10% of the time series length}; $w > 0$. Leave-one-out cross validation and 1-NN classifier were used to calculate the accuracy for each w. The w value with a highest accuracy in the training set was then used to evaluate the test set. A boldface number indicates the highest accuracy of each dataset.

The LCSF's parameters are trained as follows: t ∈ {0.5, 0.6, 0.7, 0.8, 0.9}, w ∈ [4, 10], e' ∈ [1, 10].

Table 4-1 Classification accuracy on small size UCR datasets (* observed as noisy dataset)

Name	**Euclidean Distance**	**DTW (best global constraint)**	**DTW**	**LCSS**	**LCSS(*s*3)**	**LFN**	**LCSF**	**LSDTW(*w*)**
Beetle Fly*	0.750	0.700	0.700	0.700	0.700	0.850	0.700	**0.950(15)**
Bird Chicken*	0.550	0.700	0.750	0.750	**0.850**	**0.850**	**0.850**	**0.850(10)**
Distal Phalanx Outline Age Group*	0.782	0.772	0.792	0.760	0.758	**0.812**	0.763	0.780(4)
Distal Phalanx Outline Correct*	0.752	0.768	0.768	0.747	0.760	**0.777**	0.733	0.765(7)
Distal Phalanx TW*	0.727	0.728	0.710	0.727	0.727	**0.732**	0.725	**0.732(7)**
ECG	0.880	0.880	0.770	0.880	0.880	0.890	0.820	**0.900(8)**
ECG Five Days	0.797	0.797	0.768	0.770	0.770	0.801	0.782	**0.969(4)**
Herring*	0.516	0.531	0.531	0.578	0.485	0.594	0.563	**0.641(20)**
Italy Power Demand	0.955	0.955	0.950	0.830	0.845	**0.961**	0.893	0.948(2)

Medical Images	0.684	0.747	0.737	0.641	0.707	0.703	0.674	**0.767(2)**
Middle Phalanx Outline Age Group*	0.740	0.747	0.750	0.750	0.725	0.750	0.728	**0.755(2)**
Middle Phalanx Outline Correct	0.753	0.682	0.648	0.648	0.605	0.742	0.685	**0.755(5)**
Middle Phalanx TW*	0.561	0.581	0.584	0.576	0.584	0.591	0.589	**0.599(8)**
Mote Strain	0.879	0.866	0.835	0.897	0.903	0.882	**0.947**	0.892(4)
Proximal Phalanx Outline Age Group	0.785	0.785	0.805	0.746	0.777	0.805	0.761	**0.820(6)**
Proximal Phalanx Outline Correct	0.808	0.790	0.784	0.770	0.756	0.801	0.794	**0.856(5)**
Proximal Phalanx TW*	0.708	0.737	0.737	0.735	0.727	**0.745**	0.690	0.735(5)
Sony AIBO Robot Surface*	0.695	0.695	0.725	0.642	0.607	**0.819**	0.740	0.802(3)
Sony AIBO Robot SurfaceII	0.859	0.859	0.831	0.791	0.842	0.842	0.861	**0.861(1)**

Two Lead ECG	0.747	0.868	0.904	0.943	0.900	**0.955**	0.940	0.940(5)
Wine*	0.611	0.611	0.574	0.500	0.500	0.741	0.667	**0.815(1)**

In these datasets, Longest Common Local Scaling Feature (LCSF) preformed not quite well comparing to the LFN and LSDTW because LCSF was not created for general time series, but for the **multiple-subsequence-with-multiple-scale** time series, which were evaluated in next sections. However, LSDTW was created to cope both general and multiple-subsequence-with-multiple-scale time series such that the LSDTW performed very well among these datasets.

For the proposed Local Feature Normalization (LFN) comparing to the other popular algorithms (excluding LSDTW), Euclidean Distance, DTW (best global constraint), DTW and LCSS the proposed LFN produces the highest accuracy in 16 out of 21 datasets among other methods. Local Feature Normalization (LFN) exclusively outperforms others on noisy and low-accuracy (DTW accuracy $\leq$ 0.7) datasets, e.g., MiddlePhalanxTW, Wine, Herring, BeetleFly, and BirdChicken. For high-accuracy datasets, my LFN algorithm performs better or only slightly less accurately as this research observes that these datasets are already in the correct scale. Local Feature Normalization (LFN) outperforms LCSS (original and s3) on 19/21 datasets, which shows significant improvement of my normalization. If c or w is too small, obtained LSFs might not cover true important features and that some errors may occur. Local Feature Normalization (LFN) impressively outperforms DTW with global constraints and LCSF in 19/21 datasets.

Comparing the proposed Local Scaling Dynamic Time Warping (LSDTW), to the other popular algorithms (excluding LFN): Euclidean Distance, DTW (best global constraint), DTW and LCSS, the proposed Local Scaling Dynamic Time Warping (LSDTW) produces the highest accuracy in 14 out of 21 datasets.

Comparing only the Local Feature Normalization (LFN) and the Local Scaling Dynamic Time Warping (LSDTW), LSDTW has the highest accuracy in 15 out of 21 datasets. If every method is taken into the account, LSDTW produces the highest accuracy in 14 out of 21 datasets.

4.1.2 Larger-size UCR datasets

These datasets were additionally used to evaluate the classification accuracy of the proposed Local Scaling Dynamic Time Warping (LSDTW) on general time series. LFN has higher time complexity and more parameters than LSDTW, but LFN cannot perform better than LSDTW such that LFN will not be more evaluated. Similar to the previous experiments, 1-nearest neighbor (1-NN) classifier was used on 20 UCR public datasets, comparing the results with DTW, DDTW, AFBDTW1, AFBDTW2 and LCSF. Every dataset on this archive has also already been Z-normalized, labeled, and split into training and test sets.

Grid search was used in the training set to find the *w* parameter; *w* was selected from {1, 1%, 2%,…, 10% of the time series length}; $w > 0$. Leave-one-out cross validation and 1-NN classifier were used to calculate the accuracy for each *w*. The *w* value with a highest accuracy in the training set was then used to evaluate the test set. **Table 4-2** compares the classification accuracies of the proposed Local Scaling Dynamic Time Warping (LSDTW) with other rival methods, i.e., DTW, DDTW, AFBDTW1, and AFBDTW2, along with their improvement over DTW. A boldface number indicates the highest accuracy of each dataset, and a red number indicates a decline in the accuracy comparing with DTW.

Longest Common Local Scaling Feature (LCSF) was also evaluated here, even though Longest Common Local Scaling Feature (LCSF) has such a large time complexity that was almost impossible to complete the experiments on all datasets within realistic amount of time.

Table 4-2 Classification accuracy and the improvement over DTW

Dataset Name	**DTW**	**DDTW**	**AFBDTW1**	**Improvement**	**AFBDTW2**	**Improvement**	**LCSF**	**LSDTW(w)**	**Improvement**
50 words	0.690	0.651	0.787	14.06%	**0.807**	16.96%	0.712	0.798(24)	15.62%
Adiac	0.604	0.591	0.660	9.27%	0.683	13.08%	0.458	**0.685(12)**	13.41%
Beef	0.633	0.767	0.667	5.37%	0.633	0%	0.667	**0.967(9)**	**52.76%**
CBF	0.997	0.577	0.996	-0.1%	0.979	-1.81%	0.813	**0.999(1)**	0.2%
Coffee	**1**	0.929	0.821	-17.9%	0.864	-13.6%	0.964	**1(17)**	0%
ECG200	0.770	0.730	0.880	14.29%	0.880	14.29%	0.820	**0.900(8)**	16.88%
Face All	0.808	0.445	0.811	0.37%	0.802	-0.74%	0.795	**0.812(7)**	0.47%
Face Four	0.830	0.580	0.875	5.42%	0.875	5.42%	**0.955**	0.943(11)	13.64%
Fish	0.823	0.851	0.903	9.72%	**0.949**	15.31%	0.943	0.931(19)	13.17%
Gun Point	0.907	0.907	0.980	8.05%	0.980	8.05%	0.973	**0.987(14)**	8.78%
Lightning2	0.869	0.623	0.885	1.84%	0.885	1.84%	0.803	**0.902(1)**	3.76%

Lightning7	0.726	0.562	0.712	-1.93%	0.699	-3.72%	0.671	**0.808(10)**	11.29%
Olive Oil	0.833	0.833	0.833	0%	0.800	-3.96%	**0.933**	0.833(40)	0%
OSU Leaf	0.591	0.731	0.731	23.69%	0.756	27.92%	0.777	**0.843(9)**	**42.64%**
Swedish Leaf	0.792	0.827	0.891	12.5%	0.886	11.87%	0.842	**0.891(13)**	12.53%
Synthetic Control	**0.993**	0.570	0.977	-1.61%	0.947	-4.63%	0.933	0.970(1)	-2.32%
Trace	**1**	0.960	**1**	0%	**1**	0%	0.980	**1(11)**	0%
Two Patterns	**1**	0.813	**1**	0%	**1**	0%	**1**	**1(1)**	0%
wafer	0.980	0.944	0.993	1.33%	0.994	1.43%	**0.996**	0.993(2)	1.33%
yoga	0.836	**0.958**	0.868	3.83%	0.866	3.59%	0.836	0.866(17)	3.59%

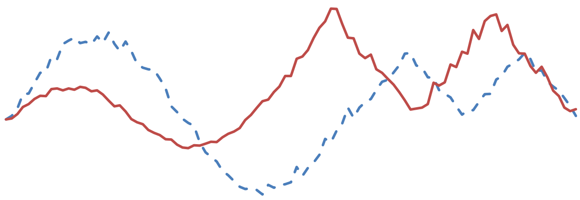
The proposed Local Scaling Dynamic Time Warping (LSDTW) could perform as good as or outperformed DTW in 19 out of 20 datasets, outperformed DDTW in 19 out of 20 datasets, outperformed AFBDTW1 in 18 out of 20 datasets, and outperformed AFBDTW2 in 16 out of 20 datasets. In some datasets, Local Scaling Dynamic Time Warping (LSDTW) drastically outperformed DTW—a 52.76% improvement in Beef dataset and a 46.64% improvement in OSULeaf dataset. In particular, time series sequences within each class of these datasets actually contain subsequences with similar shape but in different scales, exactly for which characteristic the Local Scaling Dynamic Time Warping (LSDTW) was particularly created.

Comparing to the Longest Common Local Scaling Feature (LCSF), Local Scaling Dynamic Time Warping (LSDTW) performed as good as or outperformed LCSF in 16 out of 20 datasets. Most importantly, Longest Common Local Scaling Feature (LCSF) performs poorly in many datasets; LCSF outperforms DTW only in 10 datasets. The reason behind this poor performance is that the Longest Common Local Scaling Feature (LCSF) considers only the local scaling distance such that it cannot distinguish, for example, $sin(x)$ shape and $10sin(x)$ shape.

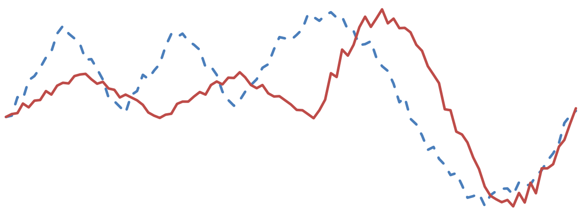
In classification problem, the Longest Common Local Scaling Feature (LCSF) is good only for the **multiple-subsequence-with-multiple-scale** time series but not the others in general. On the other hand, Local Scaling Dynamic Time Warping (LSDTW) is the updated version from Longest Common Local Scaling Feature (LCSF) that counters this problem by taking the original data point into consideration.

4.2 Synthetic Data

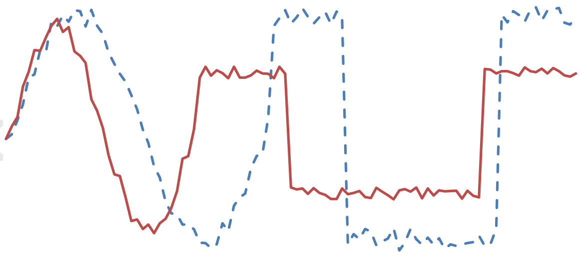
To emphasize the robustness of the proposed Local Scaling Dynamic Time Warping (LSDTW) and Longest Common Local Scaling Feature (LCSF), they both were evaluated on synthetic datasets, which are designed to be **multiple-subsequence-with-multiple-scale** time series. Local Feature Normalization (LFN) was also compared here. The generated datasets have four classes. Each class contains two non-overlapping subsequences, selected from the three patterns: a sine wave, a square wave, or a triangle wave. Each subsequence length varies from 30 to 60 data points with its amplitude varying from 0.1 to 10.0. The generated dataset was perturbed by 3 means: 1) 5-20% noise multiplier (random noise between zero and the amplitude multiplied by the noise multiplier is added to each data point), 2) 5-20% missing data, and 3) 5-20% combination of noise and missing data. The challenge of this dataset is that each time series sequence within each class will contain multiple subsequences of different lengths and different scaling. Each sequence has a total length of 100 data points, and each class contains 25 sequences (100 instances/dataset). Examples of the datasets are shown in **Fig. 4-1**.



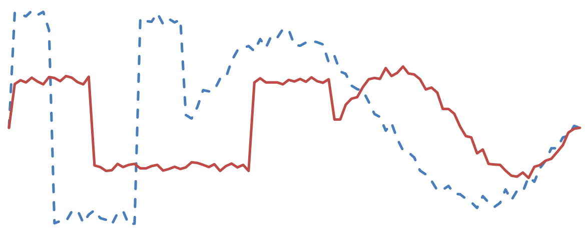
CLASS 1



CLASS 2



CLASS 3



CLASS 4

Fig. 4-1 Examples of the four classes of synthetic data

The experiment tuned the parameter *w* and evaluated the performance using five-fold cross-validation and 1-NN classifier. As shown in **Table 4-3**, the proposed Local Scaling Dynamic Time Warping (LSDTW) significantly outperforms all other methods by giving almost all perfect classification results, demonstrating robustness of the proposed method. LFN obviously perform worse than LCSF and LSDTW, which are created for handling this kind of time series.

Table 4-3 Classification accuracy on synthetic datasets

missing chance%:noise multiplier%	**DTW**	**DDTW**	**AFBDTW1**	**AFBDTW2**	**LFN**	**LCSF**	**LSDTW**
5:00	0.93	0.78	0.98	0.96	0.96	**1.00**	**1.00**
10:00	0.96	0.72	0.94	0.92	0.96	**1.00**	**1.00**
15:00	0.93	0.86	0.98	**1.00**	0.96	0.99	**1.00**
20:00	0.86	0.78	0.90	0.90	0.95	0.98	**1.00**
0:05	0.94	0.90	0.98	0.98	0.98	**1.00**	**1.00**
0:10	0.98	0.86	0.96	0.96	0.95	**1.00**	**1.00**

0:15	0.98	0.78	**1.00**	**1.00**	0.98	0.99	**1.00**
0:20	0.95	0.76	**1.00**	**1.00**	0.96	0.97	**1.00**
5:05	0.92	0.78	0.98	0.98	0.98	**1.00**	0.98
10:10	0.95	0.84	0.92	0.92	0.95	**1.00**	0.98
15:15	0.93	0.68	0.98	0.98	0.93	0.99	**1.00**
20:20	0.86	0.76	0.90	0.88	0.86	0.96	**0.98**

4.3 Real World data and Applications

To explore possibilities in applying the proposed Local Scaling Dynamic Time Warping (LSDTW) and Longest Common Local Scaling Feature (LCSF) to real-world problems, this experiment looked at the problem on wrist strengthening rehabilitation exercises for wrist injuries according to Dr. Steve Lucey, an orthopedic surgeon at Sports Medicine & Joint Replacement of Greensboro (Lucey, 2018). This wrist strengthening exercise has three exercise routines, i.e., wrist flexion, wrist extension, and wrist radial deviation, as shown in **Fig. 4-2**. Wrist flexion starts with palm up and slowly bends the wrist upward, then returns to the starting position. Wrist extension starts with palm down and slowly bends the wrist upward, then returns to the starting position. Wrist radial deviation starts with the wrist in the sideways position with the thumb pointing upward, then bends the wrist upward without moving the forearm before returning to the starting position. As a typical rehabilitation exercise usually contains multiple repetitive moves, the speed and force made by human in each repetition generally vary. This can cause a signal to have multiple subsequences with variability in scales, both in X and Y axes.

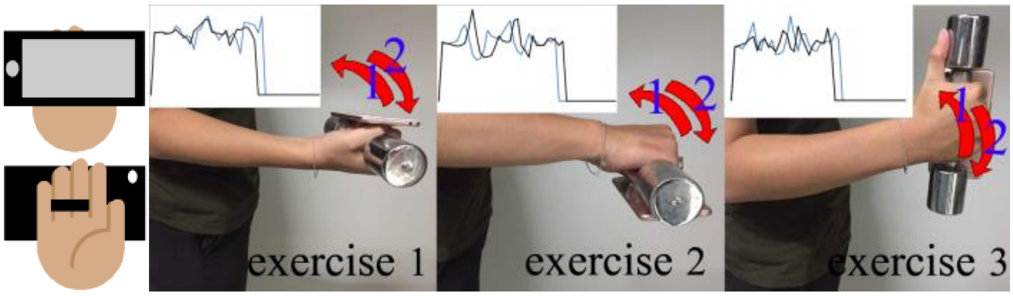


Fig. 4-2 The three wrist strengthening exercises for wrist injuries with sampled time series sequences showing for each class; a mobile phone is tied and taped on middle and ring fingers.

To capture the signals as time series sequences, an accelerometer from a mobile phone was used as a sensor. An Android application was written to collect accelerometer data (magnitude channel) with 20 Hz sampling rate from a smart phone (Android 5.1.1, 2.0 GB RAM). The Android application collected 70 samples for each class (210 samples in total), each with two repetitions of the routine to make the movement more apparent on the sampled signals. The maximum length of the sample is 48 data points.

The experiment was evaluated using 10-fold and 5-fold cross-validation with 1-NN classifier. The *w* parameter was trained using the same way as the experiments on UCR datasets. As shown in **Table 4-4**, the proposed LSDTW has impressively better accuracy especially over DTW and DDTW.

Table 4-4 Accuracy on wrist strengthening rehabilitation exercise classification

Fold	**DTW**	**DDTW**	**AFBDTW2**	**AFBDTW1**	**LFN**	**LCSF**	**LSDTW**
10	0.842	0.857	0.892	0.892	0.857	0.920	**0.971**
5	0.880	0.894	0.921	0.917	0.894	0.937	**0.971**

CHAPTER 5
CONCLUSION

This thesis proposed a Local Feature Normalization (LFN) and a Local Scaling Feature (LSF) to normalize noisy/scaled data, and a Longest Common Local Scaling Feature (LCSF) similarity measure for time series containing multiple subsequences with a variety of scales. The classification results show that the proposed Local Feature Normalization (LFN) is very robust, especially on high-error and noisy datasets, and the proposed Longest Common Local Scaling Feature (LCSF) also outperforms others on both synthetic and real datasets. The execution time of Local Feature Normalization (LFN) and Longest Common Local Scaling Feature (LCSF) are O(*mn*) the same as DTW and LCSS if w and c are fixed.

This thesis also proposed an extension version of Local Feature Normalization (LFN) and Longest Common Local Scaling Feature (LCSF) called Local Scaling Dynamic Time Warping (LSDTW) to solve a major limitation of Dynamic time warping, the inflexible Y-axis matching. This Y-axis inflexibility makes DTW cannot handle **multiple-subsequence-with-multiple-scale** time series, while the proposed Local Scaling Dynamic Time Warping (LSDTW), which bases itself on Dynamic Time Warping with a simple but powerful distance function, gives more flexibility on both X and Y axes.

Local Scaling Dynamic Time Warping (LSDTW) significantly outperforms many existing approaches; DTW, DDTW, AFBDTW, and LCSF, on most datasets, especially on synthetic and real-world application datasets, while practically consuming the same time complexity as the original DTW.

The future researches would be exploring on extending Local Scaling Dynamic Time Warping (LSDTW) to work with global constraints and lower bounding technique to further reduce time complexity and increase accuracy.

REFERENCES

Bar-Joseph, Z., Gerber, G., Gifford, D. K., Jaakkola, T. S., & Simon, I. (2002). A New Approach to Analyzing Gene Expression Time Series Data *Proceedings of the Sixth Annual International Conference on Computational Biology* (pp. 39–48). New York, NY, USA: ACM.

Chen, X., Zhang, X., Zhao, Z., Yang, J., Lantz, V., & Wang, K. (2007). Hand Gesture Recognition Research Based on Surface EMG Sensors and 2D-accelerometers *2007 11th IEEE International Symposium on Wearable Computers* (pp. 11-14).

Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., & Batista, G. (2015). The UCR Time Series Classification Archive http://www.cs.ucr.edu/~eamonn/time_series_data

Chonbodeechalermroong, A., & Ratanamahatana, C. A. (2018). Robust Scale-Invariant Normalization and Similarity Measurement for Time Series Data. In A. Sieminski, A. Kozierkiewicz, M. Nunez, & Q. T. Ha (Eds.), *Modern Approaches for Intelligent Information and Database Systems* (pp. 149-160). Cham: Springer International Publishing.

Crouch, D., & Huang, H. (2016). Simple EMG-driven musculoskeletal model enables consistent control performance during path tracing tasks *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 1-4).

Das, G., Gunopulos, D., & Mannila, H. (1997). Finding similar time series *Principles of Data Mining and Knowledge Discovery* (pp. 88-100): Springer, Berlin, Heidelberg.

Faundez-Zanuy, M. (2007). On-line Signature Recognition Based on VQ-DTW *Pattern Recogn.* (Vol. 40, pp. 981–992).

Fredman, M. L. (1975). On Computing the Length of Longest Increasing Subsequences *Discrete Math.* (Vol. 11, pp. 29–35).

Gavrila, D. M., & Davis, L. S. (1995). 3-D model-based tracking of human upper body movement: a multi-view approach *Proceedings of International Symposium on Computer Vision - ISCV* (pp. 253-258).

Godin, C., & Lockwood, P. (1989). DTW schemes for continuous speech recognition: a unified view *Computer Speech & Language* (Vol. 3, pp. 169-198).

Huang, B., & Kinsner, W. (2002). ECG frame classification using dynamic time warping *IEEE CCECE2002. Canadian Conference on Electrical and Computer Engineering. Conference Proceedings (Cat. No.02CH37373)* (Vol. 2, pp. 1105-1110).

Keogh, E., & Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping *Knowledge and Information Systems* (Vol. 7, pp. 358-386).

Keogh, E. J., & Pazzani, M. J. (2001). Derivative Dynamic Time Warping *First SIAM International Conference on Data Mining (SDM'2001)*.

Kim, S.-W., Park, S., & Chu, W. W. (2001). An index-based approach for similarity search supporting time warping in large sequence databases *Proceedings 17th International Conference on Data Engineering* (pp. 607-614).

Lucey, S. (2018). Rehabilitation Exercises for Wrist and Hand Injuries. Retrieved from https://drlucey.com/wp-content/uploads/2018/01/WristHandFingerExercises.pdf

Marzal, A., & Palazón, V. (2005). Dynamic Time Warping of Cyclic Strings for Shape Matching *Pattern Recognition and Image Analysis* (pp. 644-652): Springer, Berlin, Heidelberg.

Peterson, L. E. (2009). K-nearest neighbor *Scholarpedia* (Vol. 4).

Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., & Keogh, E. (2012). Searching and Mining Trillions of Time Series Subsequences Under Dynamic Time Warping *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 262–270). New York, NY, USA: ACM.

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition *IEEE Transactions on Acoustics, Speech, and Signal Processing* (Vol. 26, pp. 43-49).

Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., & Keogh, E. (2003). Indexing Multi-dimensional Time-series with Support for Multiple Distance Measures *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 216–225). New York, NY, USA: ACM.

Xie, Y., & Wiltgen, B. (2010). Adaptive Feature Based Dynamic Time Warping *IJCSNS* (Vol. 10).

Yun, Y., Dancausse, S., Esmatloo, P., Serrato, A., Merring, C. A., Agarwal, P., & Deshpande, A. D. (2017). Maestro: An EMG-driven assistive hand exoskeleton for spinal cord injury patients *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2904-2910).

APPENDIX

Local Scaling Dynamic Time Warping Origin (LSDTWO)

$$wl = min(w, i\text{-}1, j\text{-}1)$$
$$wr = min(w, n\text{-}i, m\text{-}j)$$
$$\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_{wl+wr+1}\} = 0\_1\_normalize(\{a_{i-wl}, \ldots, a_{i+wr}\}) \quad (0)$$
$$\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots, \beta_{wl+wr+1}\} = 0\_1\_normalize(\{b_{j-wl}, \ldots, b_{j+wr}\})$$
$$LSdist\_o(a_i, b_j) = (\boldsymbol{\alpha} - \boldsymbol{\beta})^2/(wl + 1 + wr)$$

At the beginning, the original Local Scaling Dynamic Time Warping, Local Scaling Dynamic Time Warping Origin (LSDTWO), was designed only for handling **multiple-subsequence-with-multiple-scale** time series, such that it was not consider the original data point. The Local Scaling Dynamic Time Warping Origin (LSDTWO) uses the distance function *LSdist_o* in (0) as the DTW's distance function, which is only the distance of the normalized local subsequences. The result is that the Local Scaling Dynamic Time Warping Origin (LSDTWO) could produce such an aesthetic alignment, which could handle multiple-subsequence-with-multiple-scale time series matching very well.

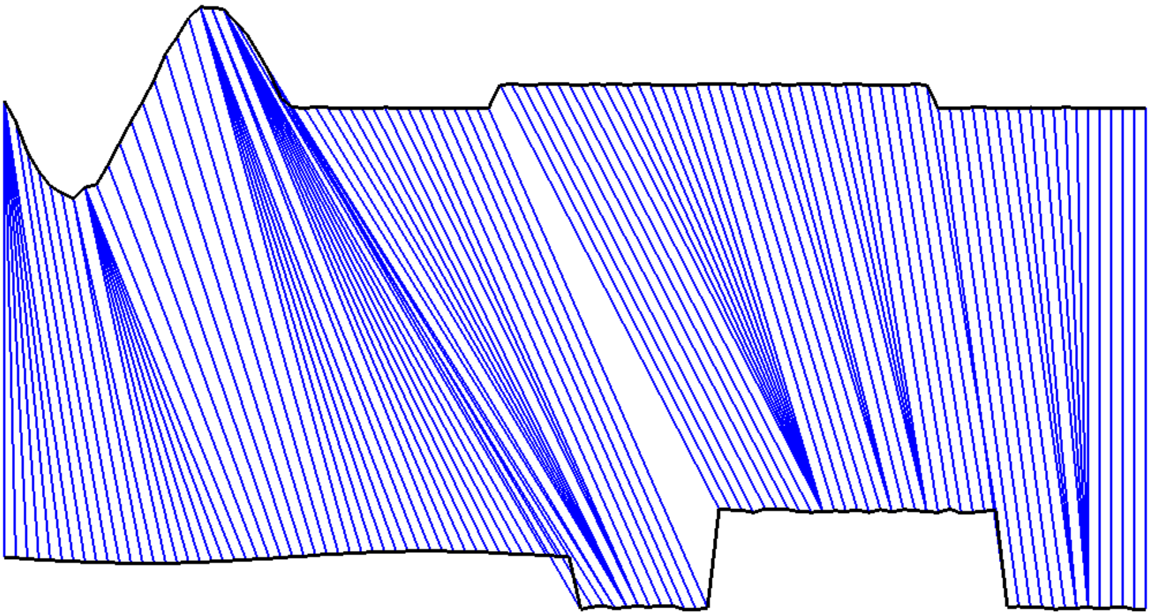
However, when LSDTWO was used in the classification problem, LSDTWO performed very poorly in many datasets as shown in **Table A-1**; Local Scaling Dynamic Time Warping Origin lost Dynamic Time Warping and Adaptive Feature Based Dynamic Time Warping in many datasets marked as underlined.

Table A-1 Preliminary experiments classification

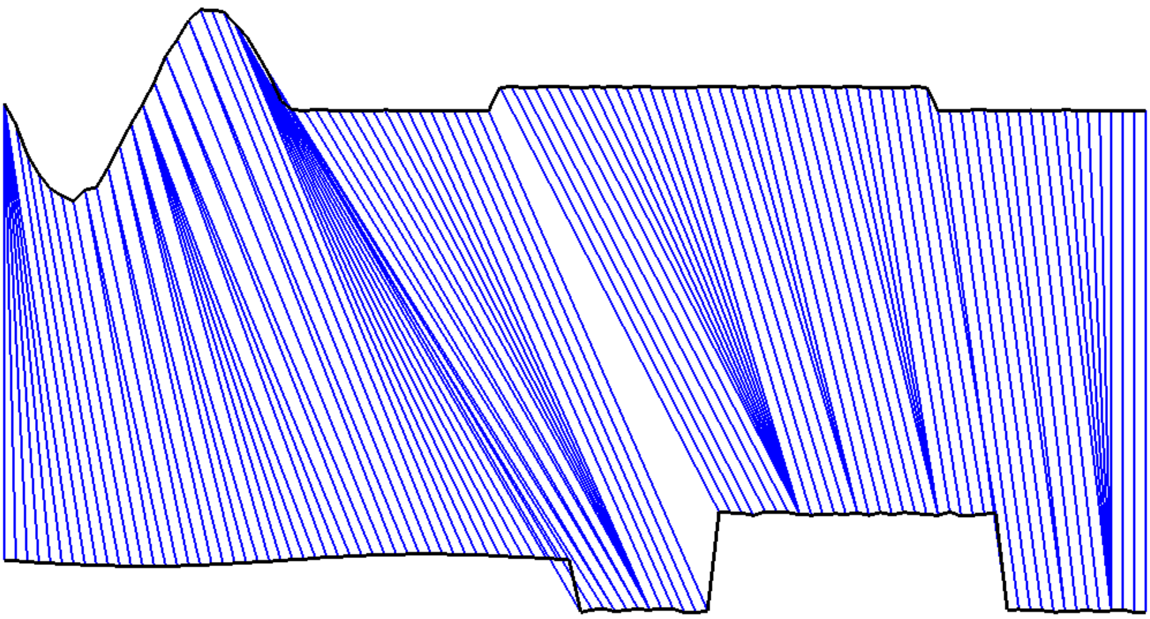
	DTW	**AFBDTW1**	**LSDTW**	**LSDTWO**
50words	0.690	0.787	**0.798**	0.741
Adiac	0.604	0.660	**0.685**	0.657

Beef	0.633	0.667	**0.967**	**0.967**
CBF	0.997	0.996	**0.999**	**0.999**
FaceAll	0.808	0.811	**0.812**	0.808
Lighting2	0.869	0.885	**0.902**	0.656
OliveOil	**0.833**	**0.833**	**0.833**	0.767

After conducting many preliminary experiments, the idea of using the original point was invented and that became the successfully proposed Local Scaling Dynamic Time Warping (LSDTW). Because the weight of the original data point of Local Scaling Dynamic Time Warping (LSDTW) is relatively small, the time series matching alignments from LSDTW and LSDTWO are not much different as illustrated in **Fig. A-1**.



a) LSDTWO



b) LSDTW

Fig. A-1 The comparison between a) LSDTWO and b) LSDTW

VITA

I am currently a Master's student in the Department of Computer Engineering, Chulalongkorn University since 2017. I graduated Bachelor of Engineering in Computer Engineering, 2015, 1st class honor, from Chulalongkorn University.