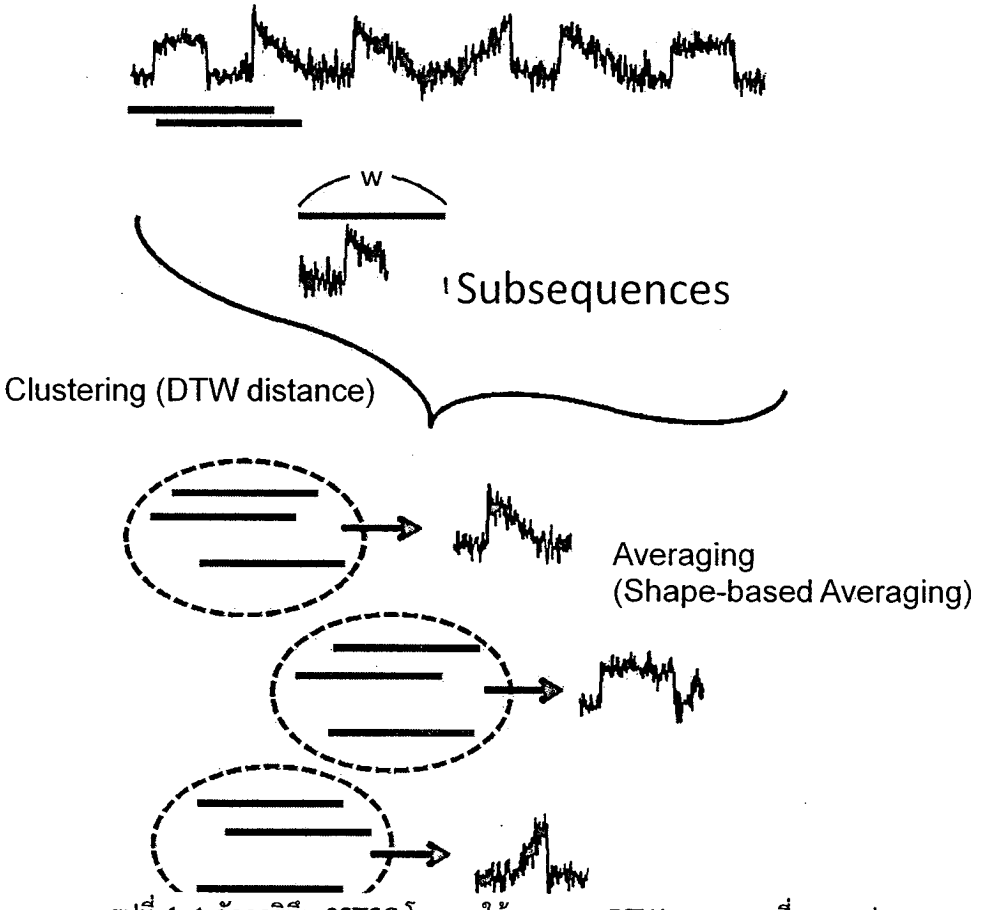
บทที่ 4
การจัดกลุ่มลำดับย่อยของข้อมูลอนุกรมเวลาแบบกระแส
Subsequence Clustering for Time Series Data Stream

ก่อนที่จะกล่าวถึงรายละเอียดของการจัดกลุ่มลำดับย่อยของข้อมูลอนุกรมเวลาแบบกระแส จะอธิบายถึงการจัดกลุ่มลำดับย่อยของข้อมูลอนุกรมเวลาตามรูปสำหรับข้อมูลอนุกรมเวลาแบบทั่วไปก่อน ซึ่งในที่นี้เรียกอัลกอริทึมนี้ว่า 2STSC (Shape-based Subsequence Time Series Clustering) จากนั้นจะมีการต่อยอดขึ้นไปเป็นอัลกอริทึม 3STSC (Shape-based Streaming Subsequence Time Series Clustering) ซึ่งเป็นอัลกอริทึมที่สามารถรองรับข้อมูลแบบกระแสได้ โดยที่ยังคงประสิทธิผลของการจัดกลุ่มลำดับย่อยได้เป็นอย่างดี

4.1 Shape-based Subsequence Time Series Clustering (2STSC)

งานวิจัยนี้ ได้เสนออัลกอริทึมการจัดกลุ่มลำดับย่อยของข้อมูลอนุกรมเวลาตามรูป (2STSC) ที่เป็นการจัดกลุ่มลำดับย่อยที่มีความหมาย โดยใช้ระยะทางไดนามิกไทม์วอร์ปปิง (DTW) และการเฉลี่ยรูปร่าง ในการวัดความเหมือนที่ถูกต้องระหว่างลำดับย่อย และในการหาค่าเฉลี่ยสมาชิกภายในกลุ่มข้อมูลเดียวกันเพื่อหาตัวแทนกลุ่ม อัลกอริทึมการเฉลี่ยรูปร่างที่นำเสนอนี้มีสองแนวทางคือ การเฉลี่ยแบบ Cubic-Spline Dynamic Time Warping (CDTW) และ การเฉลี่ยแบบ Iterative Cubic-Spline Dynamic Time Warping (ICDTW) ซึ่งทั้งสองฟังก์ชันนี้ ได้มีการนำฟังก์ชัน cubic spline interpolation มาใช้เพื่อการ re-sample ค่าในแนวแกน x ของอนุกรมที่ได้รับการเฉลี่ย หากแต่ฟังก์ชัน ICDTW นั้นมีความถูกต้องมากกว่า โดยผลลัพธ์การเฉลี่ยที่ได้จะอยู่กึ่งกลางระหว่างอนุกรมเริ่มต้นทั้งสองอนุกรม

เพื่อเป็นการแก้ปัญหา trivial-match ของลำดับย่อย ซึ่งเป็นลำดับย่อยที่ติด ๆ กันในช่วงเวลาสั้น ๆ อัลกอริทึม 2STSC จึงได้มีการผสมผสานการวัดระยะแบบไดนามิกไทม์วอร์ปปิง เข้ากับการเฉลี่ยตามรูปสำหรับการจัดกลุ่มแบบมี k ลำดับชั้น กล่าวคือ 2STSC ได้รับอนุกรมเวลาที่มีขนาดยาว $S = \langle s1, s2, \ldots sn \rangle$ เป็นอินพุต จากนั้นอนุกรมเวลานี้จะได้รับการสกัดออกมาเป็นเซต $S = \{S1, S2, \ldots, Si, Sn-w+1\}$ โดยใช้หน้าต่างขนาดความยาว w โดยที่ $= \langle si, si+1, \ldots, si+w-1 \rangle$ และ $1 \leq i \leq n-w+1$ ลำดับย่อยของอนุกรมเวลานี้จะได้รับการปรับขนาดแบบ z-normalization และจัดกลุ่มโดยใช้อัลกอริทึม k-hierarchical clustering โดยใช้การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง และฟังก์ชันการเฉลี่ยตามรูป และในลำดับสุดท้าย อัลกอริทึม 2STSC จะให้ค่าเซต $C = \{C1, C2, \ldots, Ck\}$ สำหรับกลุ่มข้อมูล k กลุ่ม โดยที่แต่ละกลุ่มข้อมูล $C = (M, R)$ ประกอบไปด้วยเซต $M = \{Si \mid Si \in S\}$ ของสมกชิกกลุ่มต่าง ๆ และตัวแทนของกลุ่ม $R = \langle r1, r2, \ldots, rw \rangle$ จากการจัดกลุ่มแบบ k-hierarchical นอกจากข้อมูลอินพุตแล้ว 2STSC ต้องการพารามิเตอร์ 2 ตัว ได้แก่จำนวนของกลุ่มข้อมูล (k) และความยาวของหน้าต่าง sliding window (w) อัลกอริทึม 2STSC โดยรวมแสดงได้ดังรูปที่ 4.1



รูปที่ 4. 1 อัลกอริทึม 2STSC โดยการใช้ระยะทาง DTW และการเฉลี่ยตามรูป

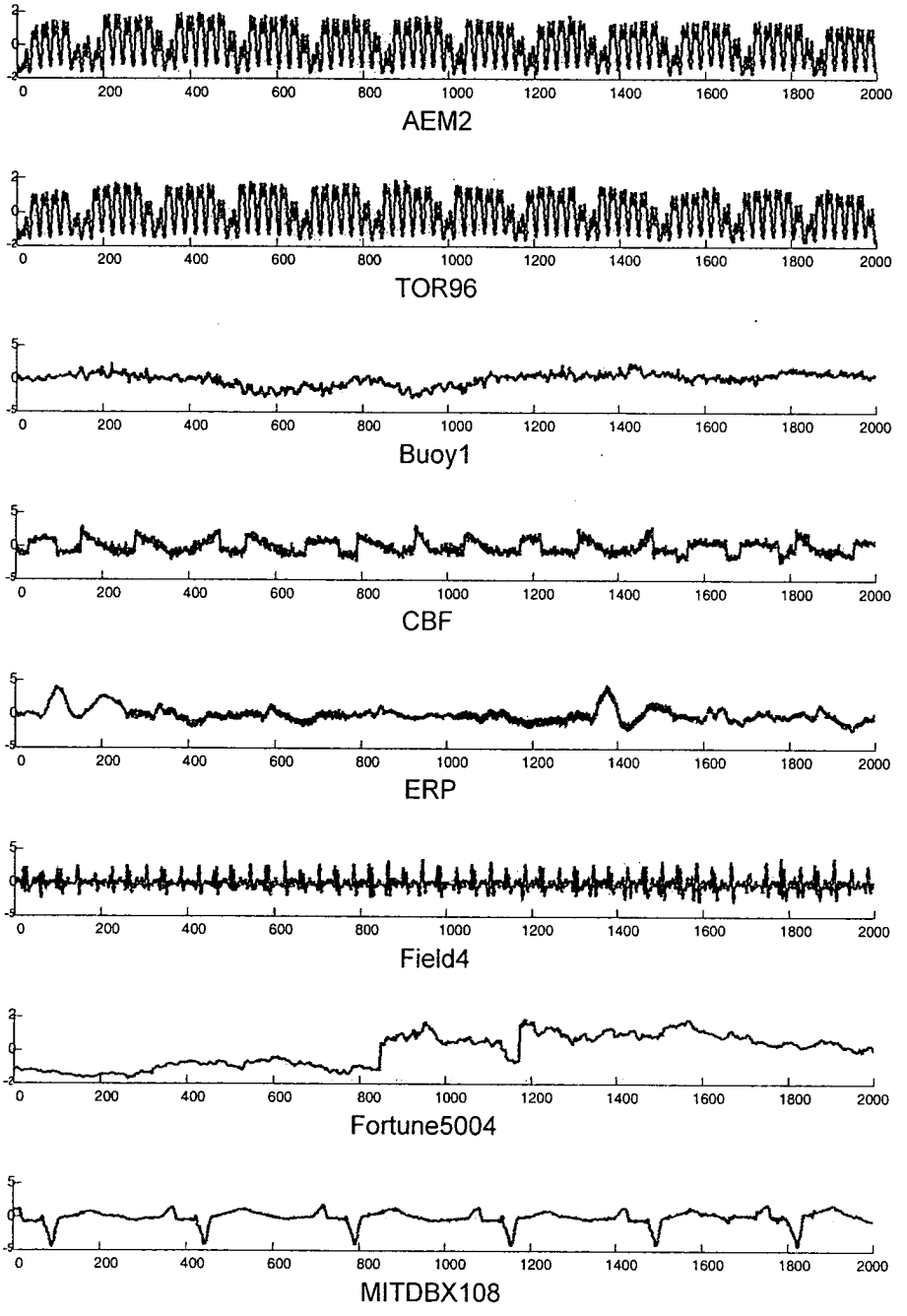
อัลกอริทึม K-hierarchical clustering ที่ใช้ใน 2STSC นั้นถือว่าเป็นการจัดกลุ่มแบบ bottom-up agglomerative ซึ่งเป็นการจัดกลุ่มที่รวมกลุ่มย่อยโดด ๆ เข้าเป็นกลุ่มใหญ่ซ้ำ ๆ การจัดกลุ่มแบบ K-hierarchical นี้จำเป็นต้องใช้ฟังก์ชันการวัดระยะทางระหว่างกลุ่มสองกลุ่ม ซึ่งในงานวิจัยนี้ได้มีการทดลองใช้ฟังก์ชันวัดระยะทางทั้งแบบ complete linkage และ average linkage ซึ่งเป็นฟังก์ชันที่ใช้ค่าระยะทางสูงสุด และค่าเฉลี่ย ของคู่ลำดับย่อยระหว่างสมาชิกในกลุ่มข้อมูลทั้งสองกลุ่มตามลำดับ ส่วนเหตุผลที่ไม่ใช้ฟังก์ชันวัดระยะทางระหว่างกลุ่มแบบ single linkage เนื่องจากฟังก์ชัน single linkage ไม่สามารถใช้กับลำดับย่อยที่เป็น trivial-match ได้ เพราะลำดับย่อยบางลำดับจะไม่มีโอกาสได้อยู่ในกลุ่มใด ๆ เลย หากลำดับย่อยเหล่านี้มีค่าระยะทางเพื่อนบ้านใกล้สุด (nearest neighbor) ที่สูงที่สุด แม้ว่าค่าระยะทางเฉลี่ยของลำดับย่อยนั้นน้อยกว่าลำดับย่อยอื่น ๆ single linkage ก็จะจับกลุ่มเฉพาะลำดับย่อยที่มีค่าระยะทางเพื่อนบ้านใกล้สุดที่น้อยกว่าเท่านั้น อัลกอริทึม 2STSC แสดงได้ดังนี้

```
FUNCTION [C] = 2STSC [S, k, w]

1.   S = EXTRACTSUBSEQUENCES(S, w)
2.   SNorm = NORMALIZESUBSEQUENCES(S)
3.   C = CLUSTERING(SNorm, k) // with DTW distance and Shape-
     based Averaging
4.   Return C
```

การทดลองเบื้องต้น

งานวิจัยนี้ได้ทำการทดสอบประสิทธิผลของอัลกอริทึม 2STSC โดยเทียบกับอัลกอริทึม STSC เดิมในแง่ของความมีความหมายของผลลัพธ์ ซึ่งอัลกอริทึม STSC ที่ใช้ในการทดลองนี้ใช้การจัดกลุ่มแบบเคมีนส์ และ k-hierarchical โดยใช้ระยะทางแบบยุคลิด และการเฉลี่ยแอมพลิจูด ส่วนอัลกอริทึม 2STSC ใช้การจัดกลุ่มแบบ k-hierarchical โดยใช้ระยะทางไดนามิกไทม์วอร์ปปิง และการเฉลี่ยตามรูป (ฟังก์ชัน CDTW และ ICDTW) ข้อมูลที่ใช้ในการทดลองมี 8 ชุดข้อมูลจาก Time Series Data Mining Archive (TSDMA) (Keogh and Folias, 2011) ซึ่งแต่ละชุดข้อมูลมีความยาว 2000 จุดข้อมูล และข้อมูลทุกชุดได้ผ่านการ normalize ดังแสดงในรูปที่ 4.2



รูปที่ 4. 2 ชุดข้อมูลที่ใช้ในการเปรียบเทียบด้านความหมายระหว่าง STSC และ 2STSC

การทดลองนี้ ได้เปรียบเทียบอัลกอริทึม 2STSC ที่เสนอ กับอัลกอริทึม STSC เดิมในแง่ของความหมาย อย่างไรก็ตามมาตรวัดความมีความหมาย KLMM ในงานของ (Lin et al., 2003; Keogh and Lin, 2005) ไม่สามารถนำมาใช้ได้ เนื่องจากยังมีความผิดพลาดอยู่มาก กล่าวคือ สมมติฐานหลักของ KLMM นั้นก็คือผลการจัดกลุ่มจากข้อมูลอินพุทเดียวกันควรต้องมีความเหมือนกัน และหากใช้ข้อมูลอินพุทต่างกันก็ควรจะมีผลการจัดกลุ่มที่มีความแตกต่างกัน ดังนั้น KLMM จึงมีการเปรียบเทียบระยะทางระหว่างผลการจัดกลุ่มจากอินพุทเดียวกัน และระหว่างผลการจัดกลุ่มจากอินพุทที่ต่างกันเท่านั้น แต่ไม่สามารถเปรียบเทียบความเหมือนหรือความต่างระหว่างสองอินพุทใด ๆ ยกตัวอย่างเช่น ถ้ามีข้อมูลอนุกรมเวลาสองอนุกรมที่คล้ายกัน ผลการจัดกลุ่มจากสองอินพุทนี้ก็ควรจะมีค่าความเหมือนกันที่สูง แต่ KLMM กลับเห็นว่าผลการจัดกลุ่มเหล่านี้ไร้ความหมาย แม้ว่าอัลกอริทึมนั้นจะให้ผลลัพธ์ที่มีความหมายจริง ๆ แล้วก็ตาม ส่วนเหตุผลที่สองก็คือมาตรวัดความคล้ายแบบ KLMM นั้นไม่สามารถบอกความเหมือนของคลื่นไซน์ที่มีเฟสหรือความถี่ที่ต่างกันได้ เนื่องจาก KLMM ใช้การวัดระยะทางแบบยุคลิดในการหาระยะทางระหว่างตัวแทนกลุ่มข้อมูล ดังนั้นผลการจัดกลุ่ม โดย KLMM ของข้อมูลไซน์นี้ จะบ่งชี้ว่ามีความแตกต่างกัน แม้ว่าในความเป็นจริงแล้วข้อมูลทั้งหมดเป็นคลื่นไซน์ที่มีความคล้ายกัน

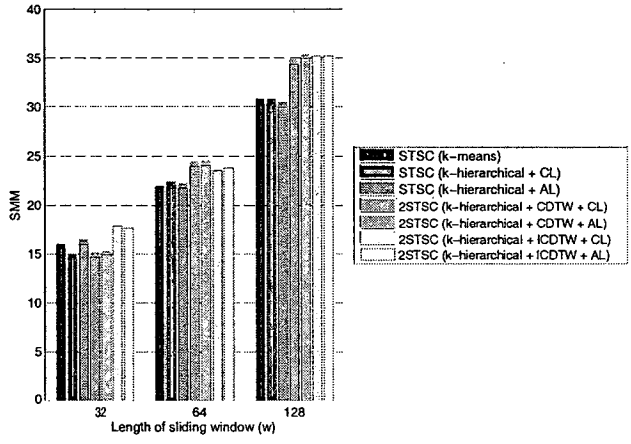
ในงานวิจัยนี้ จึงได้มีการนำเสนอมาตรวัดความมีความหมายของผลลัพธ์การจัดกลุ่มอันใหม่เรียกว่า Shape-based Meaningfulness Measurement (SMM) โดยมีไอเดียหลักว่า ผลลัพธ์จากการจัดกลุ่มจะมีความหมายก็ต่อเมื่อ ผลลัพธ์นั้นสามารถเป็นตัวแทนลำดับย่อยของข้อมูลอนุกรมเวลาได้จริง หรืออีกนัยหนึ่ง หากข้อมูลอินพุทไม่ใช่คลื่นไซน์ ตัวแทนกลุ่มก็ต้องไม่เป็นคลื่นไซน์ และหากข้อมูลอินพุทเป็นคลื่นไซน์ ตัวแทนกลุ่มก็ต้องเป็นคลื่นไซน์ไปด้วย มิฉะนั้นจะถือว่าผลลัพธ์นั้นไร้ความหมาย ซึ่งต่างจากมาตรวัดแบบ KLMM กล่าวคือ SMM คำนวณหาความมีความหมายระหว่างข้อมูลอินพุทและผลลัพธ์จากการจัดกลุ่ม ส่วน KLMM นั้นจะคำนวณหาความมีความหมายระหว่างชุดข้อมูลที่ต่างกันสองชุดเท่านั้น กำหนดให้ข้อมูลอินพุท S = <s1, s2, ..., sn> และเซตของผลลัพธ์ C = {C1, C2, ..., Ck} ของกลุ่มข้อมูลจำนวน k กลุ่ม จากนั้นเซต S = {S1, S2, ..., Si, ... Sn-w+1} ของลำดับย่อย จะถูกสกัดออกมาจากข้อมูลอินพุท S โดยใช้ sliding window ที่ขนาดความยาว w จุดข้อมูล โดยแต่ละกลุ่มข้อมูล C = (M, R) ประกอบไปด้วยเซต M = {Si | Si ∈ S} ของสมาชิกของกลุ่มข้อมูล และตัวแทนกลุ่ม R = <r1, r2, ..., rw> และเซต R = {R1, R2, ..., Rk} ของตัวแทนกลุ่มคือตัวแทนของกลุ่มข้อมูลทุก ๆ กลุ่ม ซึ่งสามารถกล่าวได้ว่า SMM เป็นมาตรวัดที่หาผลรวมของระยะทางที่น้อยที่สุดระหว่างแต่ละลำดับย่อยและตัวแทนกลุ่ม ส่วนค่าความมีความหมายสามารถคำนวณได้จากสมการดังนี้

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

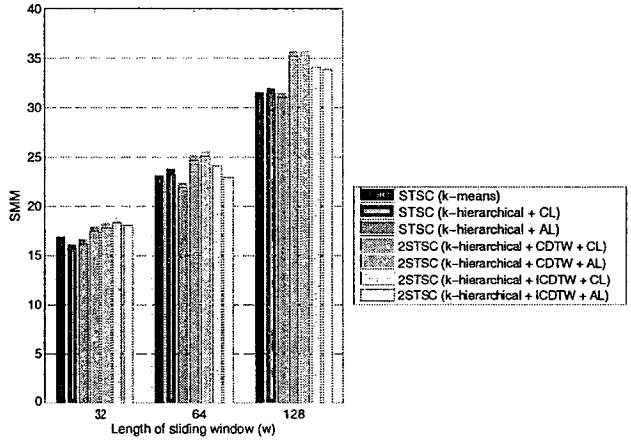
โดย Distance (Si; Rj) เป็นระยะทางไดนามิกไทม์วอร์ปปิงระหว่างสองอนุกรม Si และ Rj

ค่าของมาตรวัด SMM อยู่ระหว่าง 0 ถึง ∞ และเป็นค่าสัมพัทธ์ จึงต้องใช้ในการเปรียบเทียบกันระหว่างสองอัลกอรึทึม ที่ใช้พารามิเตอร์เซตเดียวกัน จึงจะบอกได้ว่าอัลกอริทึมสำหรับการจัดกลุ่มใดให้ผลลัพธ์ของการจัดกลุ่มที่มีความหมายมากกว่ากัน สำหรับการทดลองในส่วนนี้ พารามิเตอร์ที่ใช้มี

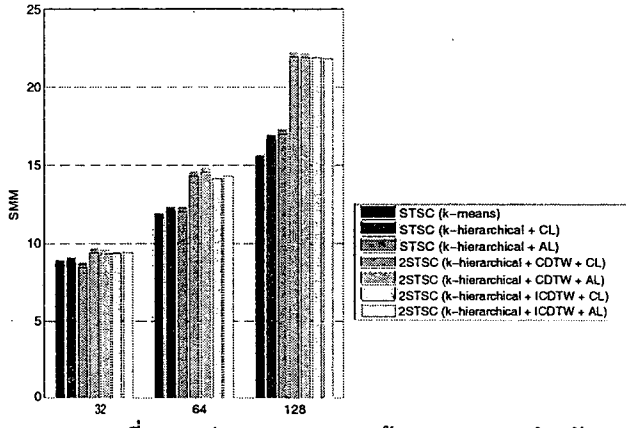
สองตัวได้แก่ ขนาดความยาว w ของ sliding window และจำนวนกลุ่มข้อมูล (k) ซึ่งจะมีการเปลี่ยนค่าพารามิเตอร์สองค่านี้ที่หลากหลาย เพื่อที่จะทดสอบผลการจัดกลุ่มลำดับย่อยของข้อมูลอนุกรมเวลา ผลการทดลองสำหรับมาตรวัด SMM ของทั้ง 8 ชุดข้อมูล แสดงในรูปที่ 4.3 – 4.10 สำหรับจำนวนกลุ่มข้อมูลมีค่าเป็น 3 และค่าขนาดความยาวของ sliding window เป็น 32, 64, และ 128 รูปที่ 4.11 – 4.18 แสดงผลของชุดข้อมูลทั้ง 8 ชุดเมื่อค่าความยาวของ sliding window เป็น 64 ส่วนจำนวนกลุ่มข้อมูลเป็น 3, 5, และ 7 รูปที่ 4.19-4.23 แสดงตัวแทนกลุ่มจากอัลกอริทึม 2STSC ที่ใช้ฟังก์ชัน CDTW ส่วนรูปที่ 4.24-4.28 แสดงตัวแทนกลุ่มจากอัลกอริทึม 2STSC ที่ใช้ฟังก์ชัน ICDTW โดยที่จำนวนกลุ่มข้อมูล และค่าความยาวของ sliding window เป็น (3, 64), (3, 128), (5, 128), (7, 128), และ (3, 256) ตามลำดับ



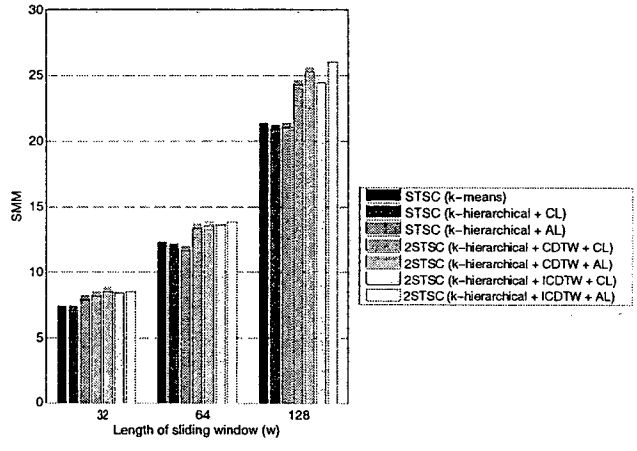
รูปที่ 4. 3 ค่า SMM ของชุดข้อมูล AEM2 สำหรับ k=3, w={32, 64, 128}



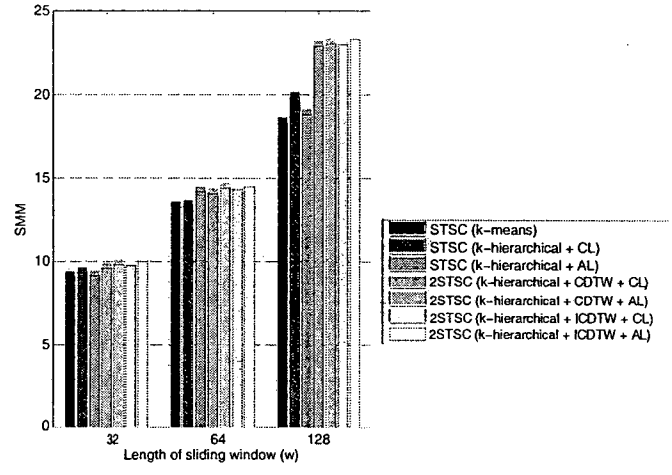
รูปที่ 4. 4 ค่า SMM ของชุดข้อมูล TOR96 สำหรับ k=3, w={32, 64, 128}



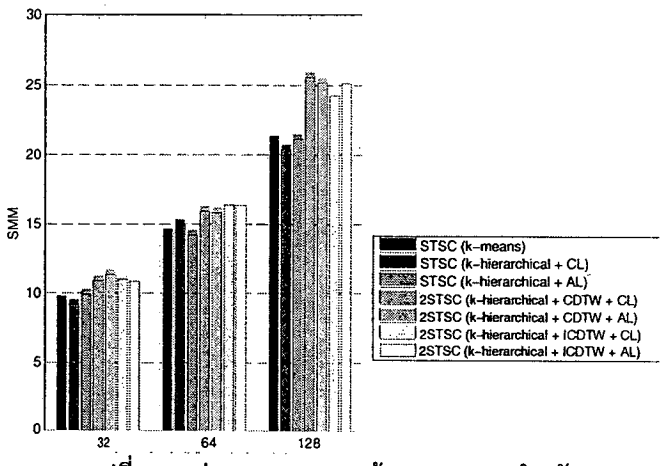
รูปที่ 4. 5 ค่า SMM ของชุดข้อมูล Bouy1 สำหรับ k=3, w={32, 64, 128}



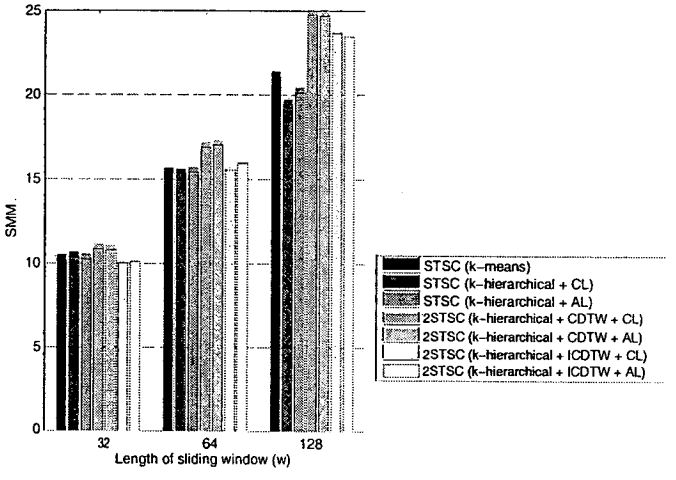
รูปที่ 4. 6 ค่า SMM ของชุดข้อมูล CBF สำหรับ k=3, w={32, 64, 128}



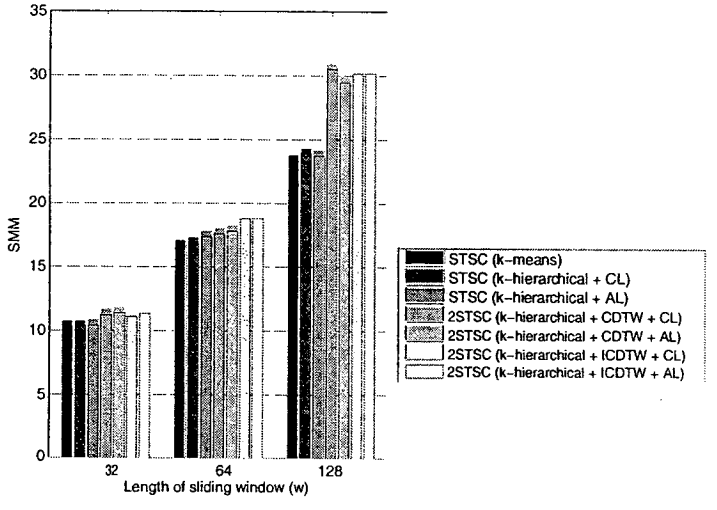
รูปที่ 4. 7 ค่า SMM ของชุดข้อมูล ERP สำหรับ k=3, w={32, 64, 128}



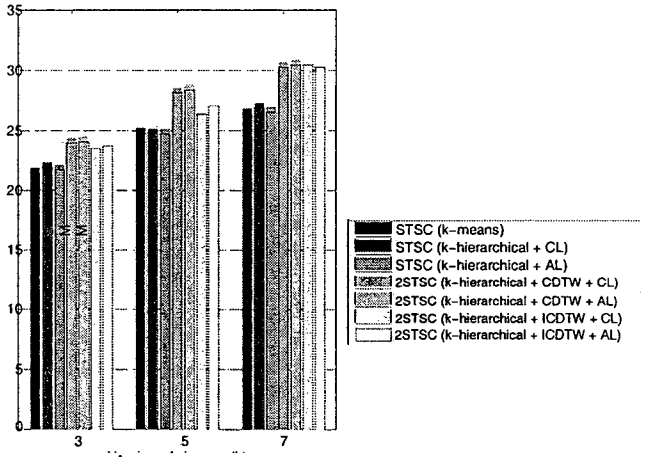
รูปที่ 4. 8 ค่า SMM ของชุดข้อมูล Field4 สำหรับ k=3, w={32, 64, 128}



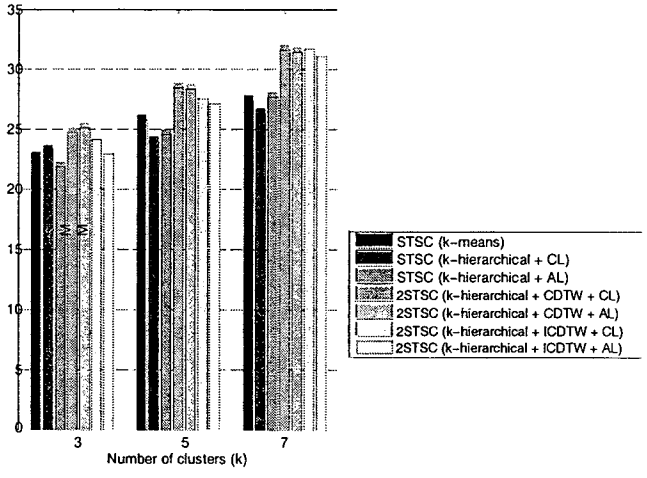
รูปที่ 4. 9 ค่า SMM ของชุดข้อมูล Fortune5004 สำหรับ k=3, w={32, 64, 128}



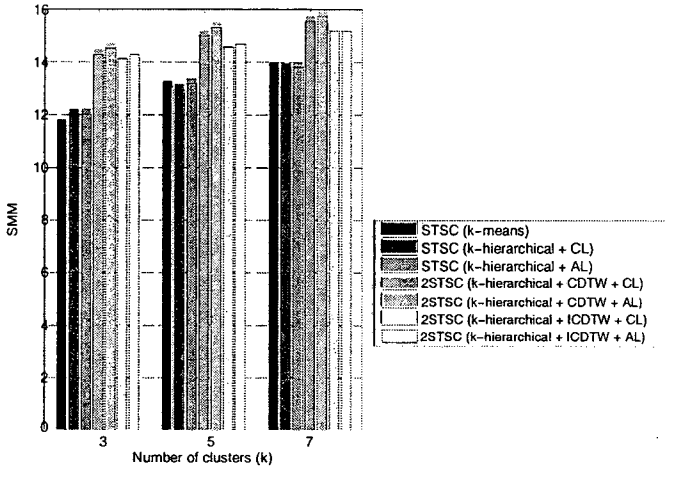
รูปที่ 4. 10 ค่า SMM ของชุดข้อมูล MITDBX108 สำหรับ k=3, w={32, 64, 128}



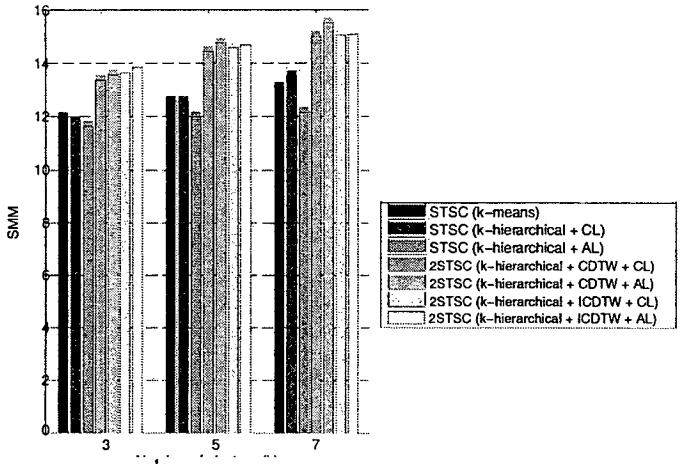
รูปที่ 4. 11 ค่า SMM ของชุดข้อมูล AEM2 สำหรับ k={3, 5, 7}, w=64



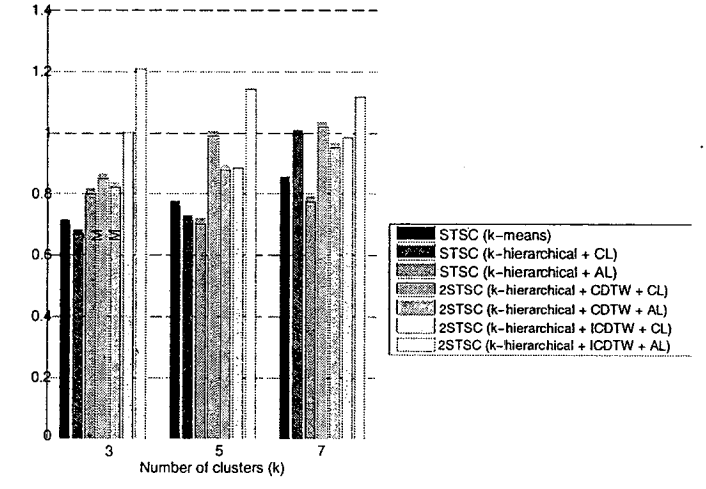
รูปที่ 4. 12 ค่า SMM ของชุดข้อมูล TOR96 สำหรับ k={3, 5, 7}, w=64



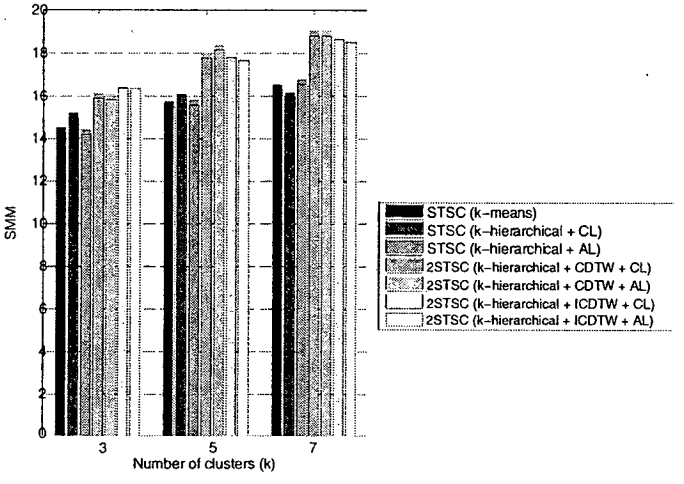
รูปที่ 4. 13 ค่า SMM ของชุดข้อมูล Bouy1 สำหรับ k={3, 5, 7}, w=64



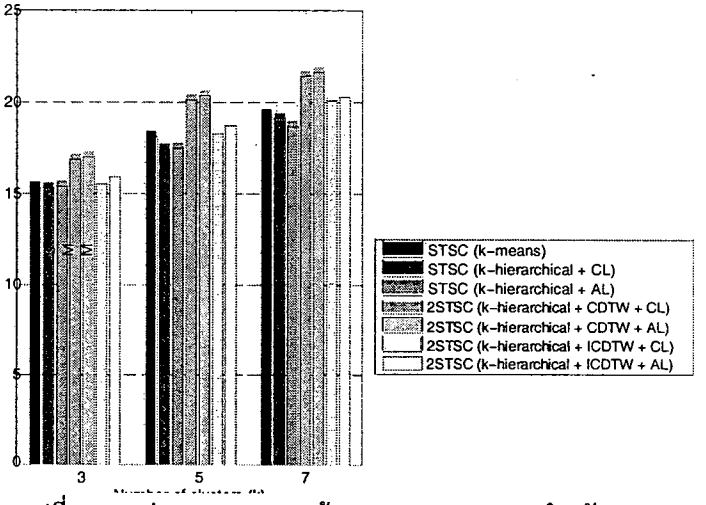
รูปที่ 4. 14 ค่า SMM ของชุดข้อมูล CBF สำหรับ k={3, 5, 7}, w=64



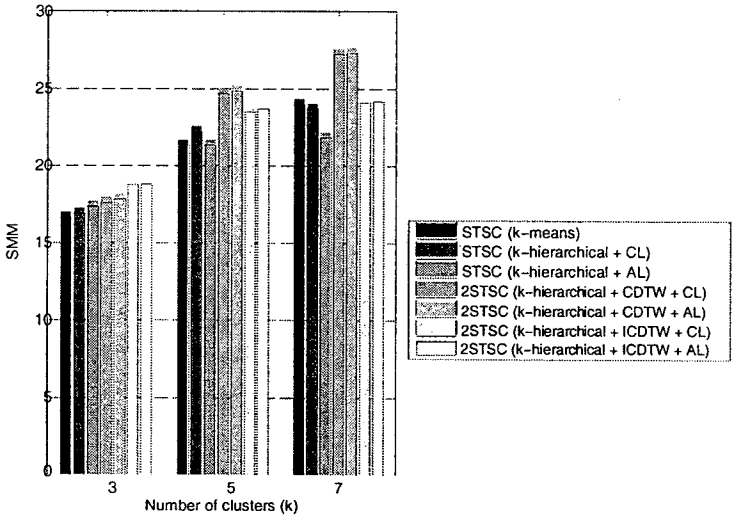
รูปที่ 4. 15 ค่า SMM ของชุดข้อมูล ERP สำหรับ k={3, 5, 7}, w=64



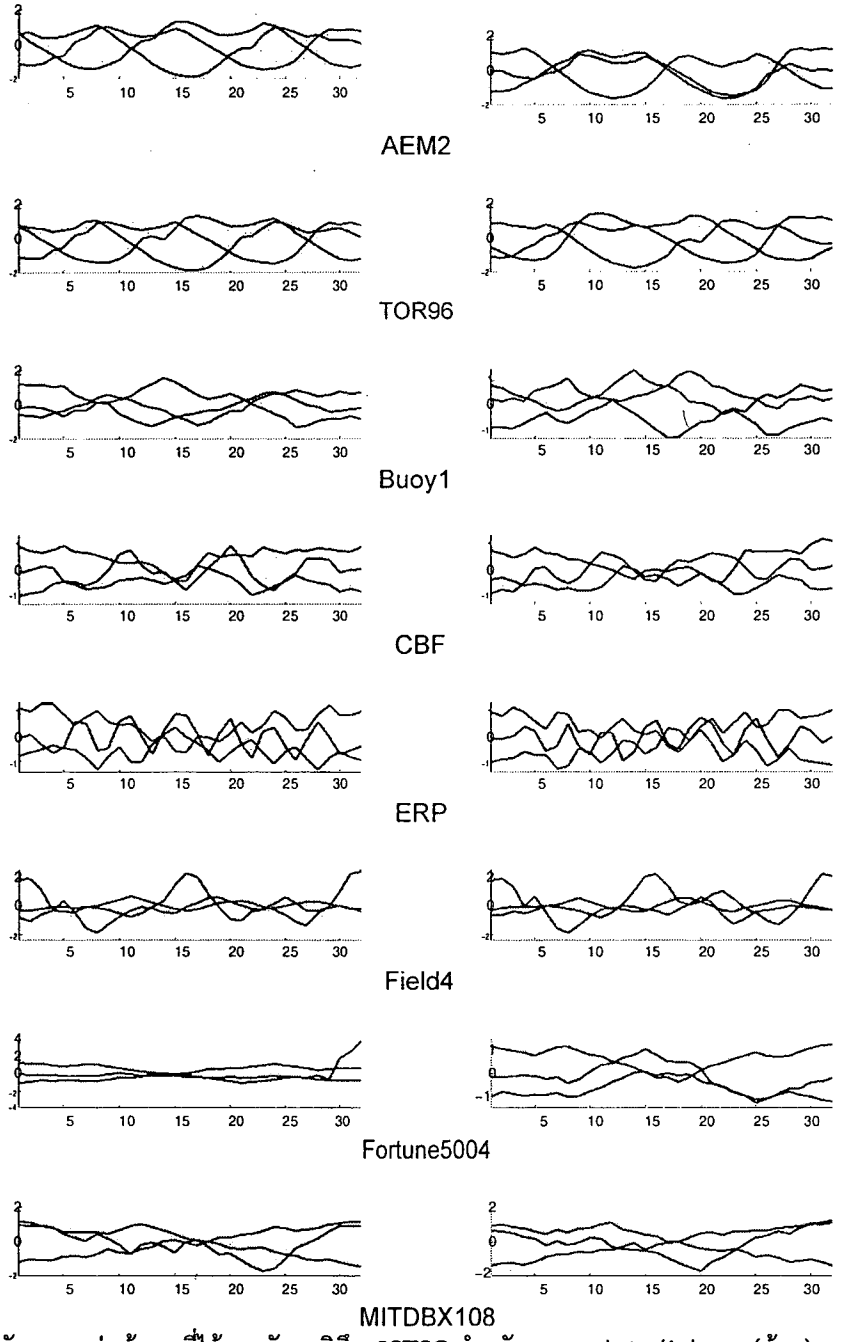
รูปที่ 4. 16 ค่า SMM ของชุดข้อมูล Field4 สำหรับ k={3, 5, 7}, w=64



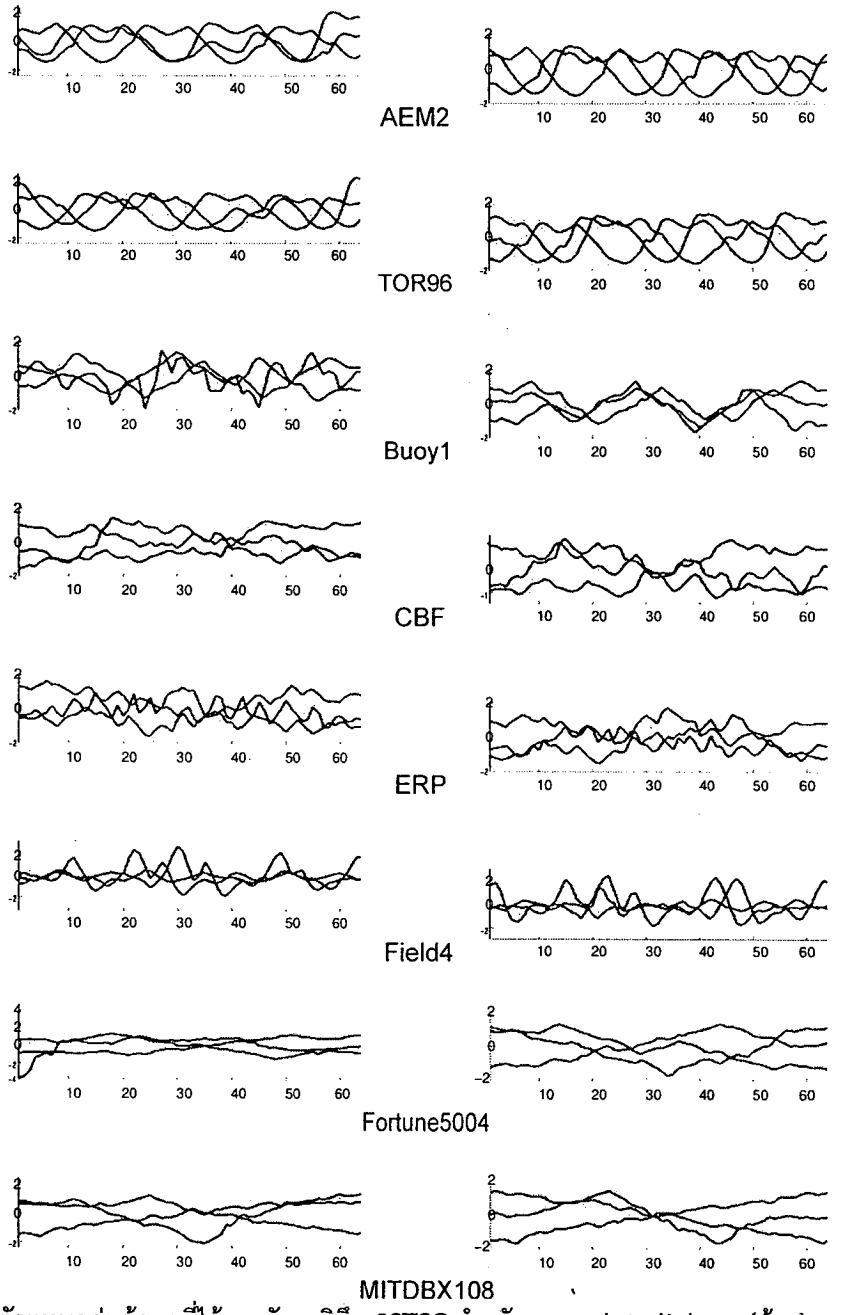
รูปที่ 4. 17 ค่า SMM ของชุดข้อมูล Fortune5004 สำหรับ k={3, 5, 7}, w=64



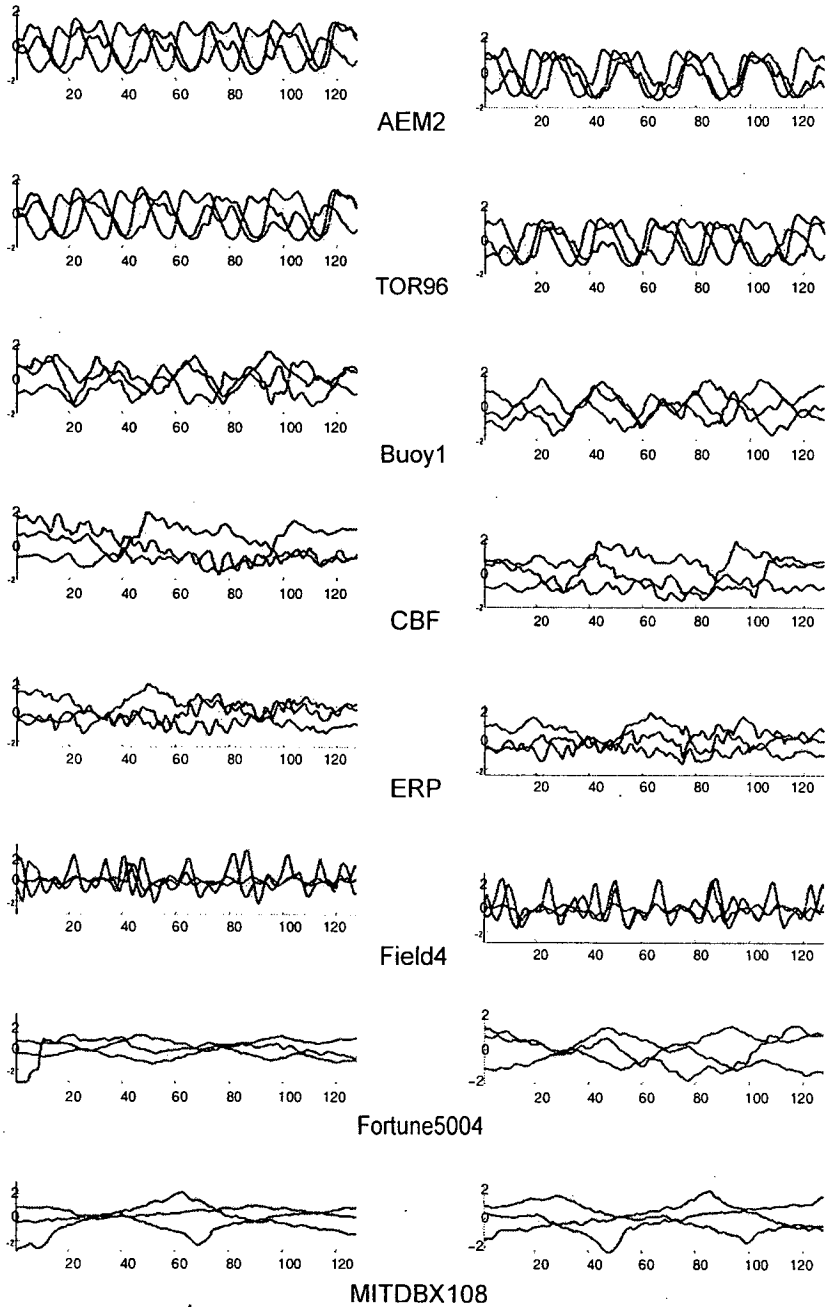
รูปที่ 4. 18 ค่า SMM ของชุดข้อมูล MITDBX108 สำหรับ k={3, 5, 7}, w=64



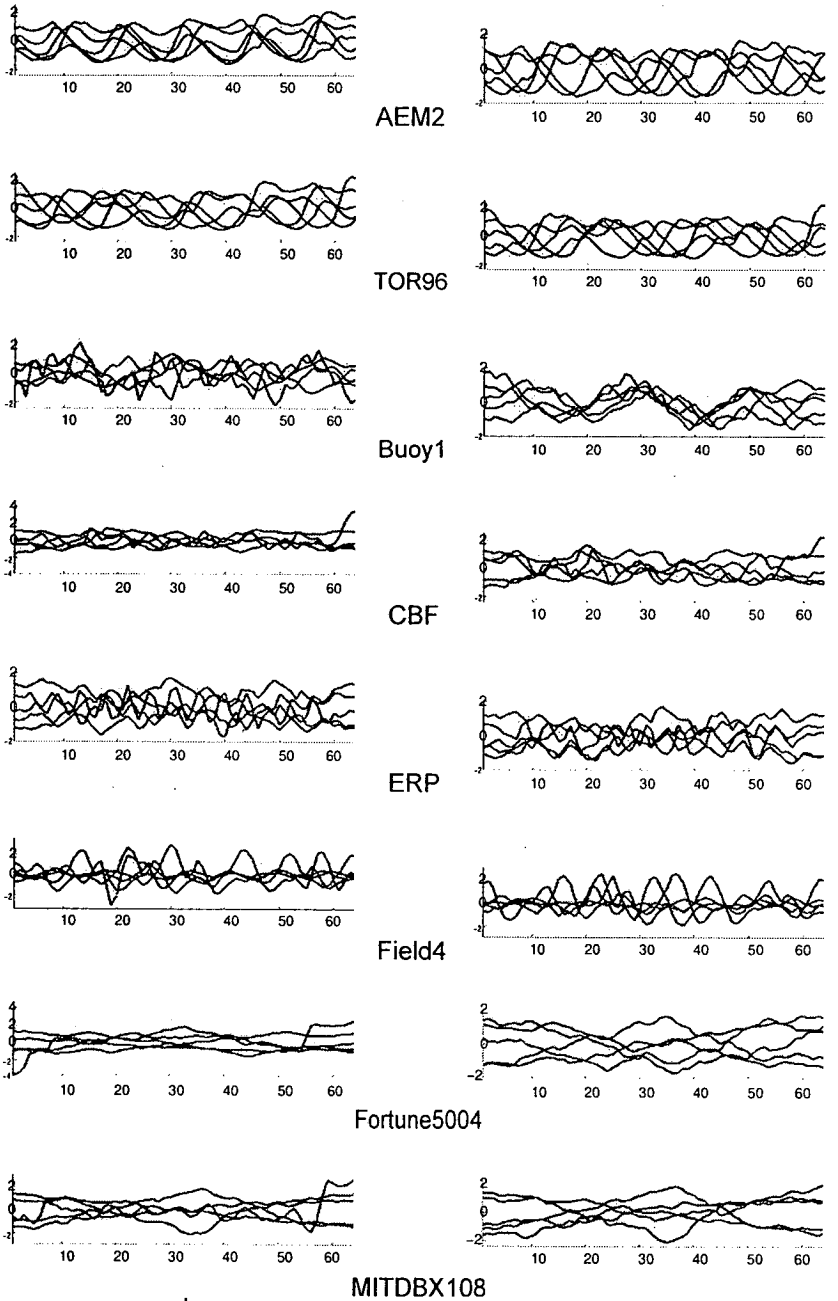
รูปที่ 4. 19 : ตัวแทนกลุ่มข้อมูลที่ได้จากอัลกอริทึม 2STSC สำหรับ complete linkage (ซ้าย) และ average linkage (ขวา) โดยใช้ฟังก์ชัน CDTW เมื่อ k = 3 และ w = 32



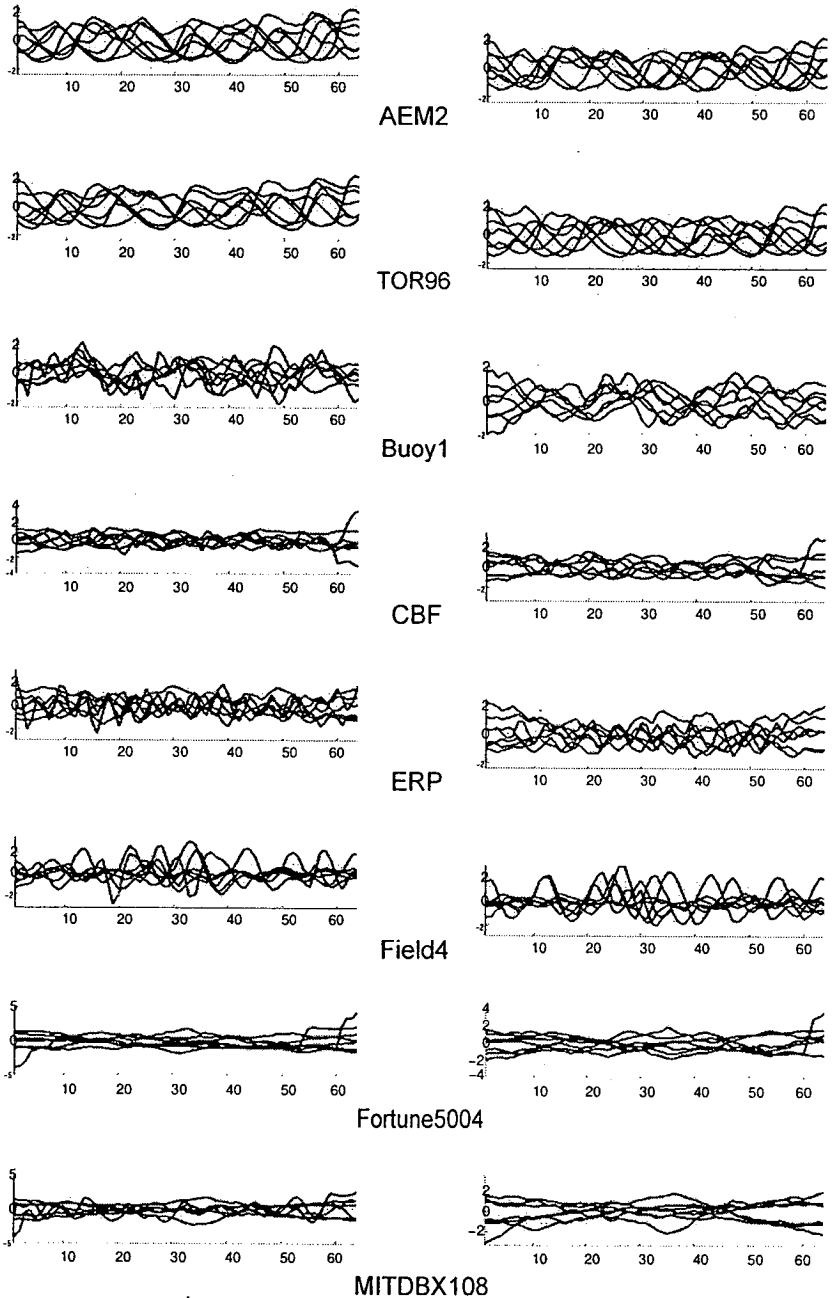
รูปที่ 4. 20 : ตัวแทนกลุ่มข้อมูลที่ได้จากอัลกอริทึม 2STSC สำหรับ complete linkage (ซ้าย) และ average linkage (ขวา) โดยใช้ฟังก์ชัน CDTW เมื่อ k = 3 และ w = 64



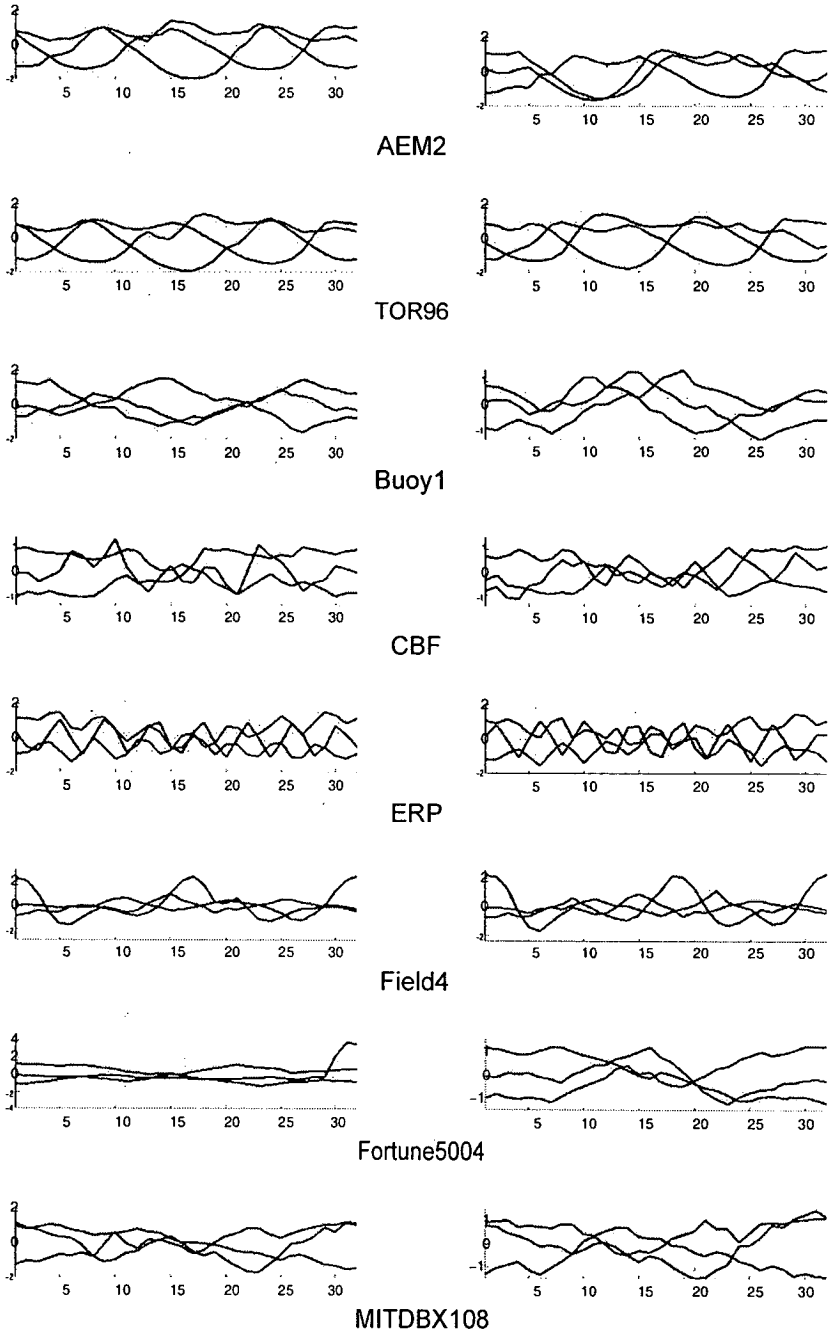
รูปที่ 4. 21 : ตัวแทนกลุ่มข้อมูลที่ได้จากอัลกอริทึม 2STSC สำหรับ complete linkage (ซ้าย) และ average linkage (ขวา) โดยใช้ฟังก์ชัน CDTW เมื่อ k = 3 และ w = 128



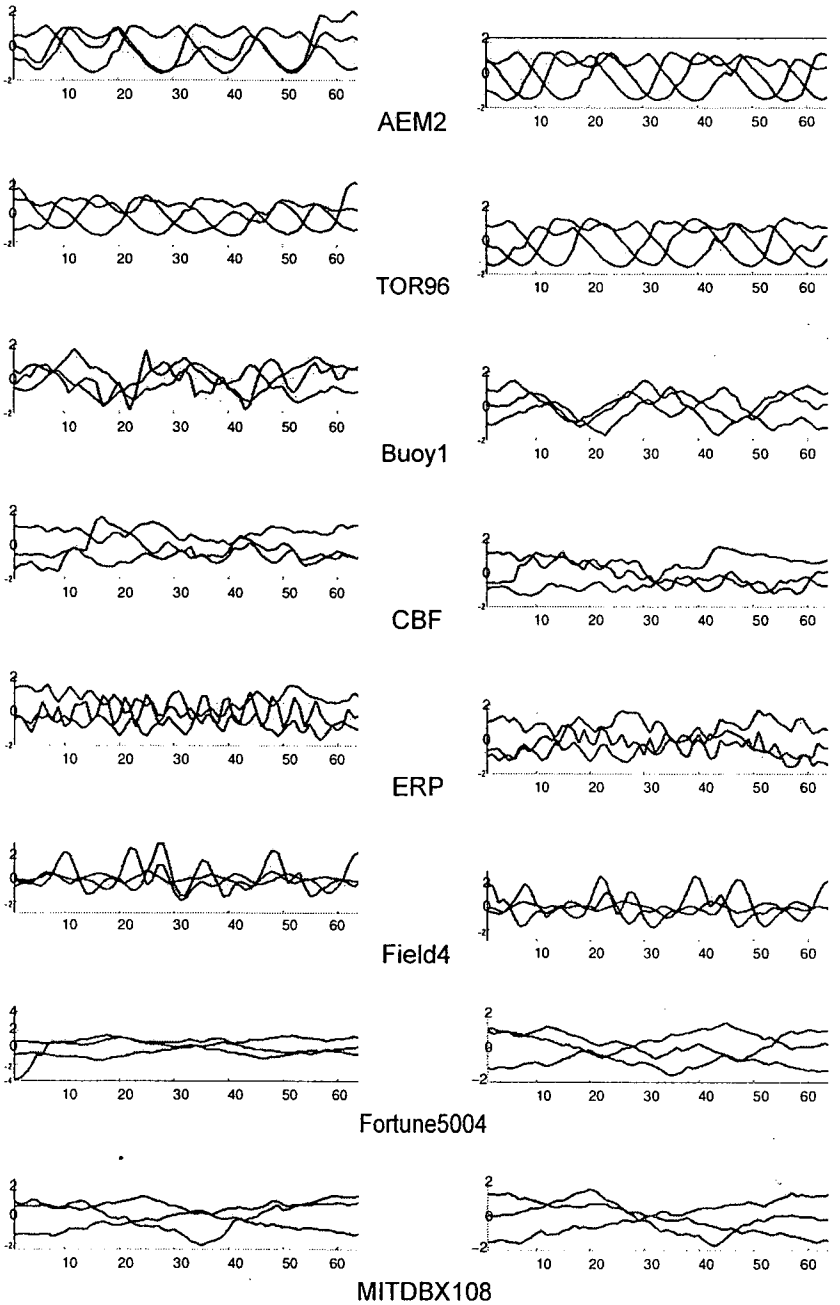
รูปที่ 4. 22 : ตัวแทนกลุ่มข้อมูลที่ได้จากอัลกอริทึม 2STSC สำหรับ complete linkage (ซ้าย) และ average linkage (ขวา) โดยใช้ฟังก์ชัน ICDTW เมื่อ k = 5 และ w = 64



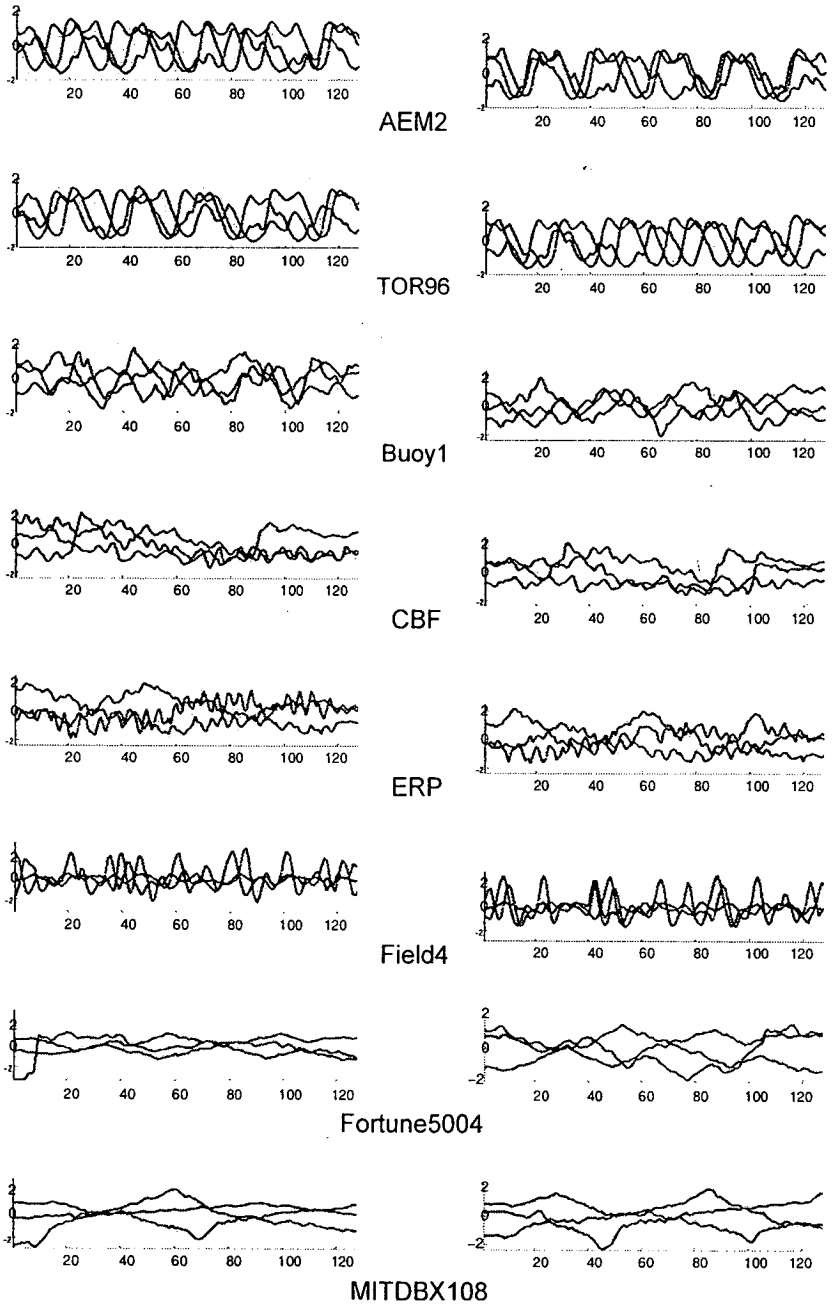
รูปที่ 4. 23 : ตัวแทนกลุ่มข้อมูลที่ได้จากอัลกอริทึม 2STSC สำหรับ complete linkage (ซ้าย) และ average linkage (ขวา) โดยใช้ฟังก์ชัน ICDTW เมื่อ k = 7 และ w = 64



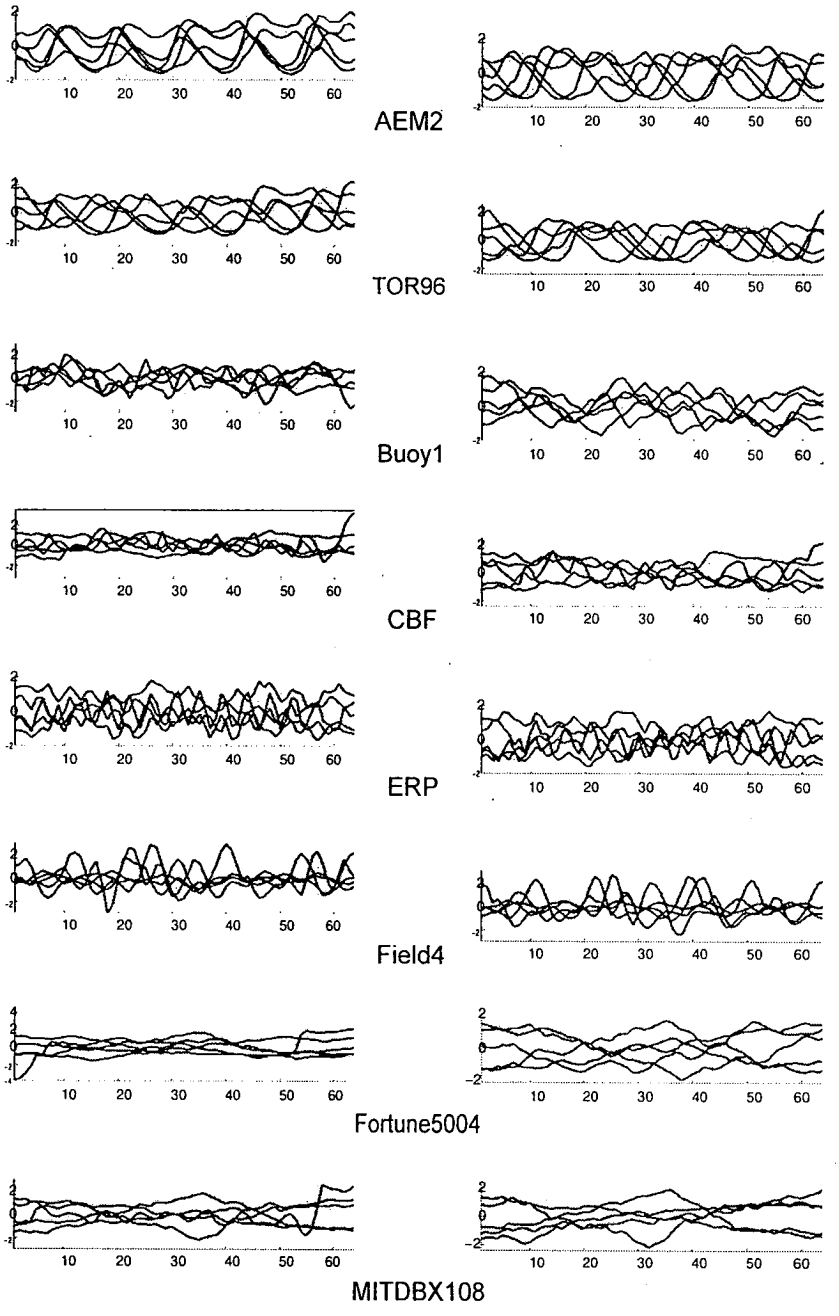
รูปที่ 4. 24 : ตัวแทนกลุ่มข้อมูลที่ได้จากอัลกอริทึม 2STSC สำหรับ complete linkage (ซ้าย) และ average linkage (ขวา) โดยใช้ฟังก์ชัน ICDTW เมื่อ k = 3 และ w = 32



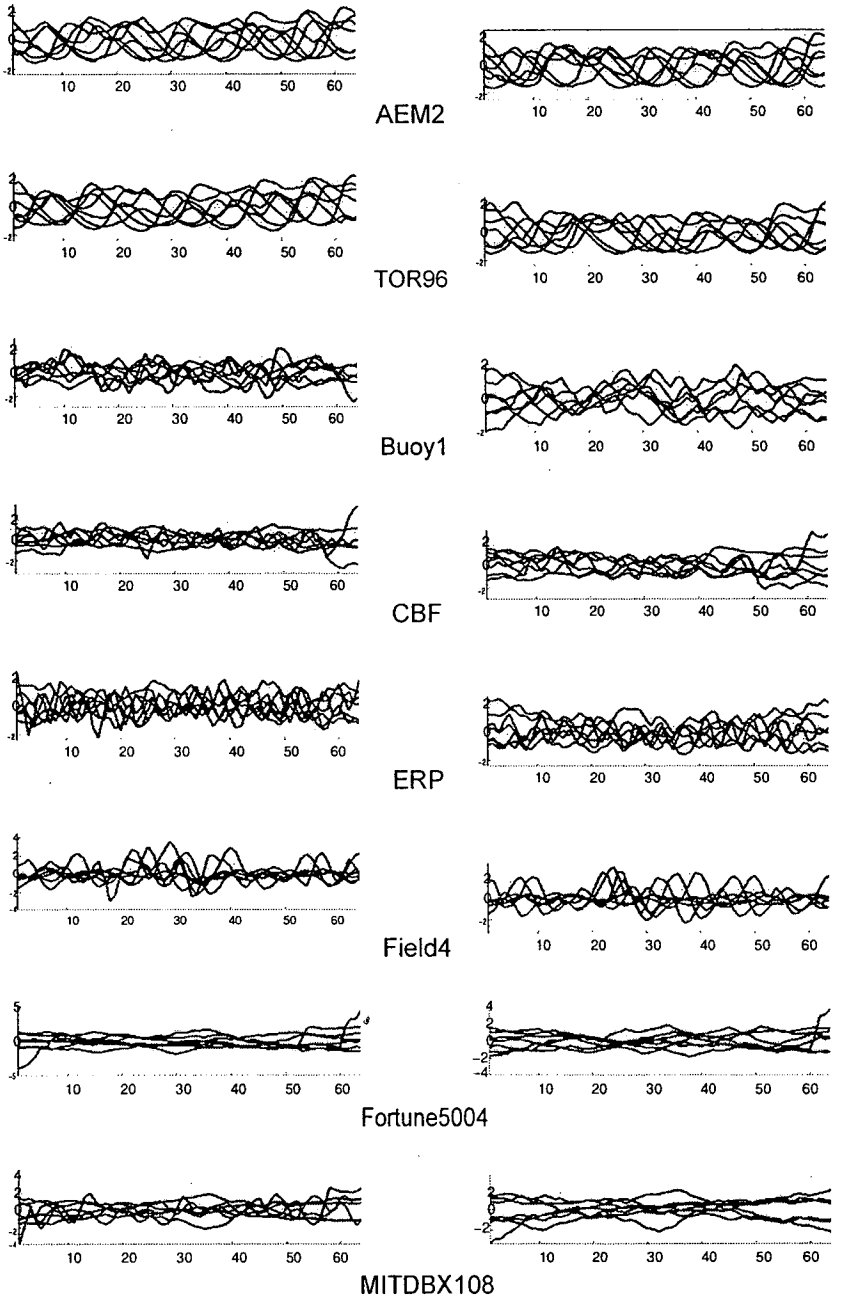
รูปที่ 4. 25 : ตัวแทนกลุ่มข้อมูลที่ได้จากอัลกอริทึม 2STSC สำหรับ complete linkage (ซ้าย) และ average linkage (ขวา) โดยใช้ฟังก์ชัน ICDTW เมื่อ k = 3 และ w = 64



รูปที่ 4. 26 : ตัวแทนกลุ่มข้อมูลที่ได้จากอัลกอริทึม 2STSC สำหรับ complete linkage (ซ้าย) และ average linkage (ขวา) โดยใช้ฟังก์ชัน ICDTW เมื่อ k = 3 และ w = 128



รูปที่ 4. 27 : ตัวแทนกลุ่มข้อมูลที่ได้จากอัลกอริทึม 2STSC สำหรับ complete linkage (ซ้าย) และ average linkage (ขวา) โดยใช้ฟังก์ชัน ICDTW เมื่อ k = 5 และ w = 64



รูปที่ 4. 28 : ตัวแทนกลุ่มข้อมูลที่ได้จากอัลกอริทึม 2STSC สำหรับ complete linkage (ซ้าย) และ average linkage (ขวา) โดยใช้ฟังก์ชัน ICDTW เมื่อ k = 7 และ w = 64

บทสรุป

การวิจัยนี้เสนอการวัดระยะแบบไดนามิกไทม์วอร์ปปิง และการเฉลี่ยตามรูป สำหรับอัลกอริทึม Shape-based Subsequence Time Series Clustering (2STSC) แทนอัลกอริทึม Subsequence Time Series Clustering (STSC) เดิม ซึ่งใช้การวัดระยะแบบยุคลิดและการเฉลี่ยแอมพลิจูด และแทนที่จะละทิ้งลำดับย่อยที่เป็น trivial-match ออกเหมือนกับงานวิจัยอื่นๆ ที่ผ่านมา อัลกอริทึม 2STSC เลือกใช้มาตรวัดระยะทางไดนามิกไทม์วอร์ปปิง เพื่อที่จะหาความคล้ายกันระหว่างเซตของลำดับย่อยที่ติดๆกัน และฟังก์ชันเฉลี่ยตามรูป เพื่อที่จะสร้างตัวแทนกลุ่มที่เหมาะสม นอกจากนั้น

ผลลัพธ์ของการจัดกลุ่มที่ได้จาก 2STSC ยังมีความหมายตามมาตรวัด Shape-based Meaningfulness Measurement (SMM) ซึ่งเป็นการวัดว่าผลการจัดกลุ่มนั้นเป็นตัวแทนข้อมูลอนุกรมเวลาที่เป็นอินพุทได้ดีเพียงใด และยังเป็นอัลกอริทึมที่ไม่ต้องการพารามิเตอร์ใด ๆ ซึ่งต่างจากอัลกอริทึมอื่น ๆ ที่เสนอมาก่อนหน้า และยังสามารถต่อยอด 2STSC ให้รองรับข้อมูลแบบกระแสได้อีกด้วย โดยการทดลองในส่วนนี้ได้แสดงให้เห็นแล้วว่าตัวแทนกลุ่มที่ได้ มีคุณลักษณะคล้ายกับข้อมูลอินพุท ในขณะที่อัลกอริทึม STSC เดิมจะให้ผลตัวแทนกลุ่มเป็นคลื่นไซน์ ที่ไม่ถูกต้อง

4.2 3STSC: SHAPE-BASED STREAMING SUBSEQUENCE TIME SERIES CLUSTERING

อัลกอริทึมสำหรับการจัดกลุ่มข้อมูลอนุกรมเวลาแบบกระแส สามารถแบ่งได้เป็น 2 ประเภท ได้แก่ การจัดกลุ่มข้อมูลทั้งอนุกรมแบบกระแส (streaming whole clustering) และการจัดกลุ่มข้อมูลลำดับย่อยแบบกระแส (streaming subsequence clustering) สำหรับการจัดกลุ่มข้อมูลทั้งอนุกรมแบบกระแสนั้น ข้อมูลอนุกรมเวลาอันใหม่ทั้งอนุกรมจะถูกใช้ในการปรับผลลัพธ์ในการจัดกลุ่ม (ตัวแทนกลุ่มข้อมูล) ในขณะที่การจัดกลุ่มลำดับย่อยแบบกระแสนั้นมีความยุ่งยากกว่ามาก เมื่อจุดข้อมูลจุดใหม่ได้ถูกต่อเข้าไปกับข้อมูลเดิม ลำดับย่อยจะถูกสกัดออกมาโดยใช้ sliding window และได้รับการ Normalize จากนั้น ตัวแทนของกลุ่มข้อมูลจะถูกปรับโดยลำดับย่อยเหล่านี้แทน หากต้องการวิธีในการแก้ปัญหาสำหรับข้อมูลแบบกระแสที่ง่ายที่สุด ก็น่าจะเป็นวิธีแบบตรงไปตรงมานั่นก็คือการหาผลลัพธ์โดยการคำนวณจากลำดับย่อยก่อนหน้าทั้งหมด สำหรับทุกๆ จุดข้อมูลใหม่ที่เพิ่มเข้ามา ซึ่งจะใช้เวลาในการคำนวณที่นานมาก ๆ ในงานวิจัยนี้ จึงจะมุ่งเน้นไปที่การหาอัลกอริทึมที่มีประสิทธิภาพ สำหรับการจัดกลุ่มข้อมูลลำดับย่อยแบบกระแส

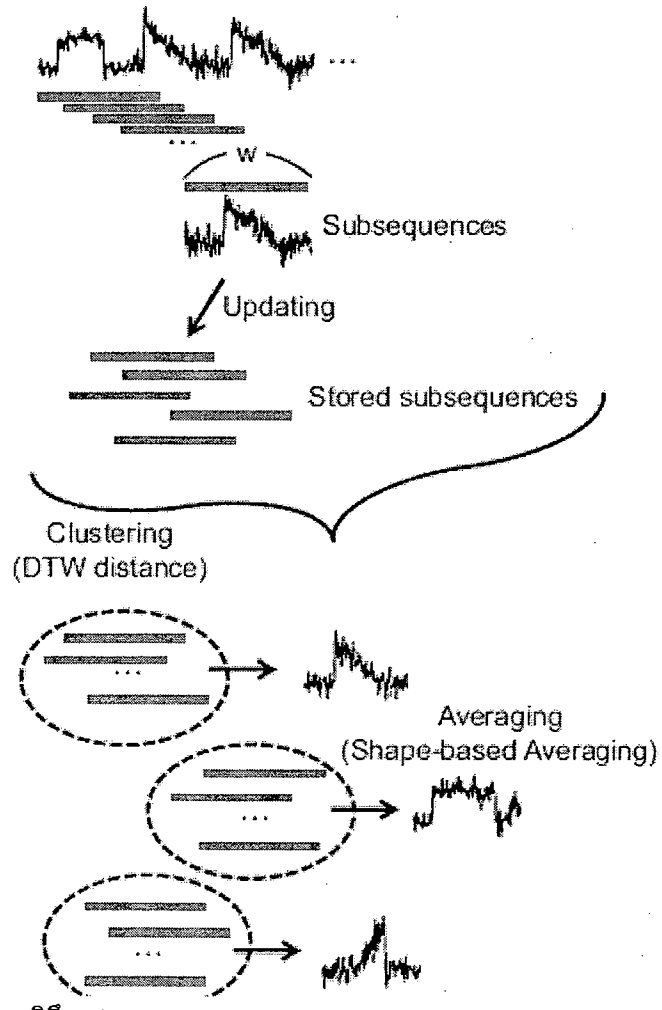
ในงานวิจัยที่ผ่านมา มีนักวิจัยได้พิสูจน์ให้เห็นแล้วว่าการจัดกลุ่มลำดับย่อย STSC นั้นไร้ความหมาย และยังไม่มีงานวิจัยใดที่สามารถแก้ปัญหานี้ได้อย่างถูกต้อง ในงานวิจัยนี้จึงจะทำการต่อยอดอัลกอริทึม 2STSC ที่ได้นำเสนอในหัวข้อที่ผ่านมา ซึ่งได้แสดงให้เห็นแล้วว่าสามารถจัดกลุ่มข้อมูลได้อย่างมีความหมาย โดยการนำการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง และฟังก์ชันการเฉลี่ยตามรูป แทนที่การคำนวณระยะทางแบบยุคลิด และการเฉลี่ยแอมพลิจูด สำหรับอัลกอริทึม STSC ซึ่งหากเป็นการต่อยอด 2STSC ให้ใช้กับข้อมูลแบบกระแสโดยตรง ก็จะไม่มีประสิทธิภาพพอ เนื่องจากจะต้องมีการคำนวณผลลัพธ์การจัดกลุ่มจากลำดับย่อยก่อนหน้าทุกลำดับย่อย และหากข้อมูลมีขนาดยาวมาก ก็จะเสียเวลาในการคำนวณมากขึ้นเป็นลำดับ

งานวิจัยในส่วนนี้ จึงจะนำเสนออัลกอริทึม Streaming Shape-based Subsequence Time Series Clustering (3STSC) ที่สามารถปรับผลลัพธ์การจัดกลุ่มได้อย่างมีประสิทธิภาพ โดยใช้เวลาคงที่ (Constant time) แทนที่จะต้องมีการคำนวณผลลัพธ์การจัดกลุ่มจากลำดับย่อยก่อนหน้าทั้งหมดเหมือนในอัลกอริทึม 2STSCอัลกอริทึม 3STSC นี้ จะทำการคำนวณผลการจัดกลุ่มจากกลุ่มของลำดับย่อยบางส่วนเท่านั้น ซึ่งการปรับลำดับย่อยนั้นจะเป็นไปตามอัลกอริทึม Incremental Shape-based Averaging (ISA) โดยจำนวนของลำดับย่อยที่เก็บไว้ จะไม่เกินจำนวนมากที่สุดที่กำหนดไว้ อัลกอริทึม 3STSC จะทำการจัดกลุ่มลำดับย่อยเหล่านี้ โดยใช้วิธีการจัดกลุ่มแบบ k-hierarchical โดยใช้การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง และการเฉลี่ยตามรูป และทำการคืนผลลัพธ์การจัดกลุ่มจากเซต

ของลำดับย่อยที่มีขนาดเล็กนี้ ซึ่งสามารถคำนวณได้อย่างรวดเร็วมาก เมื่อเทียบกับการนำอัลกอริทึม 2STSC มาใช้โดยตรง

ในส่วนของการทดลอง จะแสดงให้เห็นว่าอัลกอริทึม 3STSC นั้นมีประสิทธิภาพสูงกว่า 2STSC มาก พร้อมทั้งมีการเปรียบเทียบค่าจากมาตรวัด SMM ซึ่งวัดว่าผลลัพธ์การจัดกลุ่มโดยอัลกอริทึม 3STSC นั้นมีความหมายมากน้อยเพียงใด สำหรับค่าจำนวนกลุ่มข้อมูล (k) ขนาดความยาวของ sliding window (w) และจำนวนลำดับย่อยที่มากที่สุดที่จะเก็บไว้ ที่มีค่าต่าง ๆ กัน

Shape-based Streaming Subsequence Time Series Clustering (3STSC) เป็นอัลกอริทึมการจัดกลุ่มลำดับย่อยแบบค่อยๆ เพิ่มขึ้น (incremental) ซึ่งให้ผลลัพธ์เป็นเซตของตัวแทนกลุ่มข้อมูลสำหรับทุกๆ จุดข้อมูลใหม่ที่เพิ่มเข้ามา โดย 3STSC จะทำการต่อท้ายข้อมูลใหม่เข้ากับข้อมูลอนุกรมเวลาเดิม จากนั้นลำดับย่อยใหม่จะถูกสกัดตามขนาดของ sliding window ที่กำหนดไว้ และจะผ่านการ z-normalization โดยจะมีการปรับลำดับย่อยที่เก็บไว้ในเซตได้ไม่เกินจำนวนมากที่สุดที่อนุญาตให้เก็บไว้ หลังจากที่เซตของลำดับย่อยได้ทำการปรับเรียบร้อยแล้ว อัลกอริทึม 3STSC จะทำการจัดกลุ่มข้อมูลโดยใช้ k-hierarchical clustering กับข้อมูลลำดับย่อยที่เก็บไว้ในเซตเหล่านี้ โดยใช้การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง (DTW) และการเฉลี่ยข้อมูลตามรูป ส่วนอัลกอริทึมการปรับลำดับย่อยนั้น เป็นไปในทำนองเดียวกันกับฟังก์ชัน Incremental Shape-based Averaging (ISA) ซึ่งจำนวนของลำดับย่อยที่อนุญาตให้เก็บไว้นี้ สามารถปรับเปลี่ยนได้ตามความต้องการของผู้ใช้ ความเร็วของเครื่อง และขนาดของ Storage ที่ใช้เก็บข้อมูล รูปที่ 4.29 แสดงอัลกอริทึมของ 3STSC



รูปที่ 4. 29 : ภาพรวมของอัลกอริทึม Shape-based Streaming Subsequence Time Series Clustering (3STSC)

กำหนดให้มีข้อมูลใหม่ s_t หนึ่งจุดข้อมูล จำนวนกลุ่มข้อมูล k กลุ่ม ขนาดความยาวของ slideing window w และจำนวนลำดับย่อยที่อนุญาตให้เก็บไว้ อัลกอริทึม 3STSC จะให้ผลลัพธ์เป็นเซต C = {C1, C2, ..., Ck} ของแต่ละ k กลุ่มข้อมูล ซึ่งในอันดับแรก 3STSC จะทำการนำข้อมูล s_t มาต่อท้ายข้อมูลเดิมเป็น S = <s1, s2, ..., s_{t-1}> จากนั้นจะสกัดลำดับย่อยใหม่นี้ เป็น S = <s_{t-w+1}, ..., s_{t-1}, s_t> โดยใช้ sliding window w ที่กำหนด จากนั้นจะทำการ Z-Normalize ลำดับย่อยนี้เป็น Snorm อัลกอริทึม 3STSC จะทำการอัพเดทเซต T = {T1, T2, ..., T_∞} โดยใช้อัลกอริทึม ISA ที่เสนอมาข้างต้น ซึ่งลำดับย่อยในเซต T นี้จะได้รับการจัดกลุ่ม และให้ผลลัพธ์เป็นเซต C = {C1, C2, ..., Ck} โดยใช้อัลกอริทึมจัดกลุ่มแบบ k-hierarchical ร่วมกับการวัดระยะแบบไดนามิกไทม์วอร์ปปิง และการเฉลี่ยตามรูป แต่ละกลุ่มที่ได้รับการจัดกลุ่มและจะประกอบไปด้วยเซต M = {Ti | Ti ∈ T} ของลำดับย่อยที่เก็บไว้และตัวแทนของกลุ่ม R รหัสเทียม (pseudo code) ของอัลกอริทึม 3STSC ได้แสดงดังรูปที่ 4.30

FUNCTION [$\mathbb{C}$] = 3STSC [$\mathbb{T}$, $\mathbb{W}$ s_t, k, w, α]

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

รูปที่ 4. 30 : Pseudo code ของอัลกอริทึม Shape-based Streaming Subsequence Time Series Clustering (3STSC)

อัลกอริทึมการจัดกลุ่มแบบ K-hierarchical ที่ใช้ใน 3STSC นี้ สามารถนำไปใช้ร่วมกับฟังก์ชันคำนวณระยะทางระหว่างกลุ่มได้ทั้งแบบ complete linkage และ average linkage ซึ่งสามารถคำนวณได้ดังสมการดังนี้

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

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

โดย $D_{complete}$ และ $D_{average}$ เป็นฟังก์ชันการคำนวณแบบ complete linkage และ average linkage ตามลำดับ ส่วน C_i และ C_j เป็นกลุ่มข้อมูลใด ๆ M_i และ M_j เป็นสมาชิกของกลุ่มข้อมูล C_i และ C_j ตามลำดับ S และ S' เป็นลำดับย่อยใน M_i และ M_j และ Distance(S, S') ให้ผลลัพธ์เป็นค่าระยะทางระหว่างสองลำดับย่อย S และ S'

ในการอัพเดทลำดับย่อยที่เก็บไว้ อัลกอริทึม 3STSC ได้นำอัลกอริทึมของ Incremental Shape-based Averaging (ISA) มาใช้ โดยที่จำนวนของลำดับย่อยที่เก็บไว้จะไม่เกินจำนวนที่กำหนดไว้ (α) ซึ่งค่าที่น้อยที่สุดที่เป็นไปได้ของ α จะเท่ากับจำนวนของกลุ่มข้อมูล (k) และเมื่อมีลำดับย่อยใหม่ S_{norm} เพิ่มเข้ามา ลำดับย่อยที่ถูกเก็บไว้ที่มีระยะทางใกล้ที่สุด (TBest) จะถูกนำมาเฉลี่ยร่วมกับลำดับย่อย S_{norm} นี้และจะนำค่าเฉลี่ยที่ได้ไปแทนที่ค่า TBest เดิม และค่าน้ำหนักในการเฉลี่ยก็จะถูกเพิ่มขึ้นอีก 1 รหัสเทียม (Pseudo code) ของอัลกอริทึมการอัพเดทนี้ได้แสดงดังรูปที่ 4.31 ทั้งนี้จะเห็นได้ว่าอัลกอริทึม 2STSC นั้นถือว่าเป็นเคสหนึ่งของ 3STSC เมื่อจำนวนมากที่สุดที่ยอมให้เก็บลำดับย่อยไว้ได้ (α) เป็นค่าอนันต์ (∞)

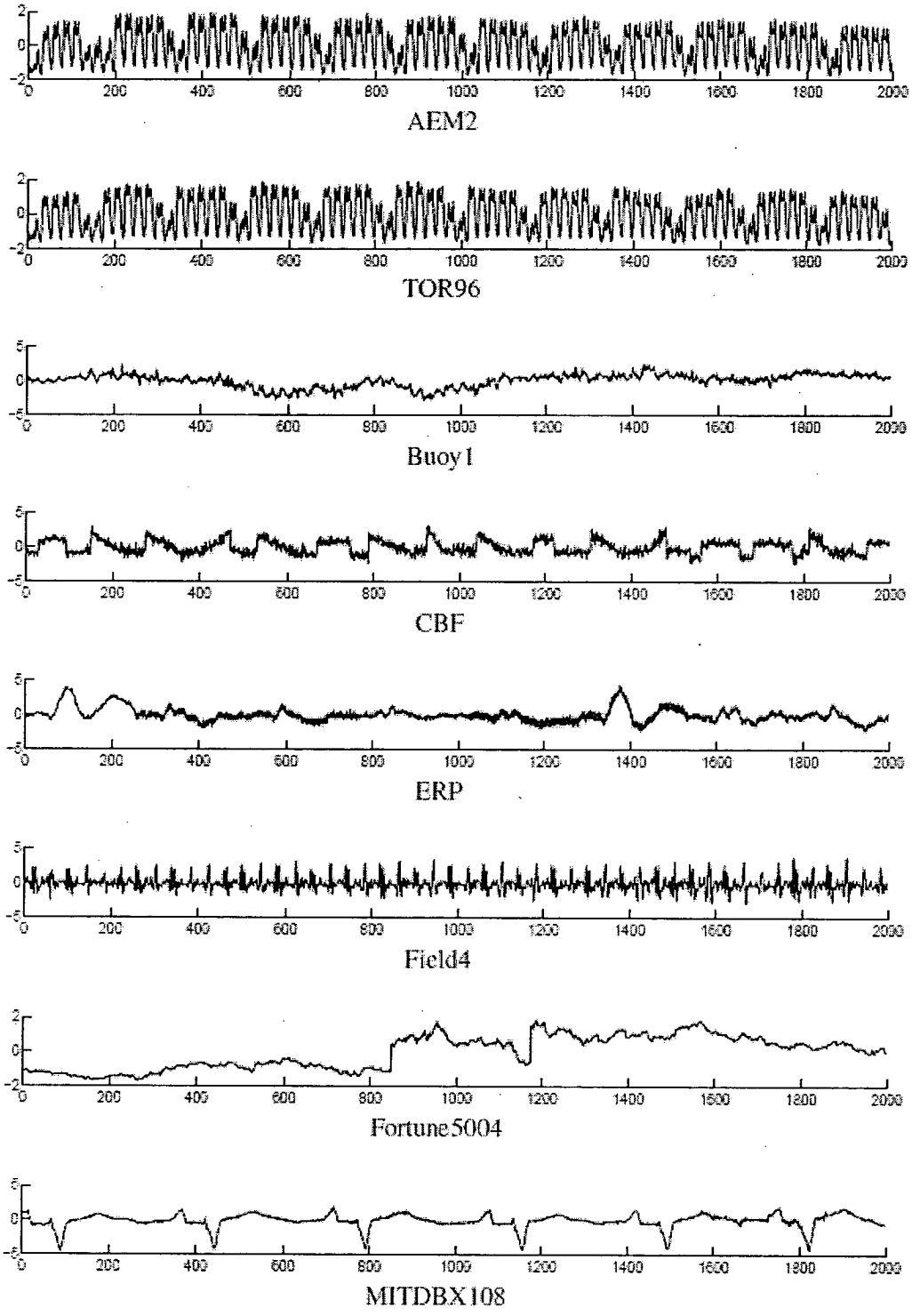
FUNCTION [$\mathbb{T}$, $\mathbb{W}$] = UPDATESTOREDSEQUENCES [$\mathbb{T}$, $\mathbb{W}$, $\mathcal{S}_{norm}$, α]

```
1.  Let n be a number of stored sequences in 𝕋
2.  If (n < α)
3.    Add S_norm in 𝕋
4.    Add w = 1 in 𝕎
5.  Else
6.    dist_Best = INFINITY
7.    For each stored sequence T_i in 𝕋
8.      dist = DTW-DISTANCE(T_i, S)
9.      If (dist < dist_Best)
10.       dist_Best = dist
11.       T_Best = T_i
12.       w_Best = w_i
13.     End if
14.   End for
15.   S_avg = AVERAGINGFUNCTION(T_Best, S_norm, w_Best, 1)
16.   Replace T_Best with S_avg
17.   Replace w_Best with w_Best + 1
18. End If
19. Return [𝕋, 𝕎]
```

รูปที่ 4. 31 : Pseudo code ของการอัพเดทข้อมูลที่ถูกเก็บไว้สำหรับอัลกอริทึม Shape-based Streaming Subsequence Time Series Clustering (3STSC)

การทดลองและวัดผล

งานวิจัยนี้ ได้เสนออัลกอริทึม Shape-based Streaming Subsequence Time Series Clustering (3STSC) เพื่อใช้ในการหาตัวแทนของกลุ่มข้อมูลแบบค่อยเป็นค่อยไป ซึ่งในที่นี้จำทำการวัดผลจากสองการทดลอง การทดลองแรกจะเป็นการแสดงค่า speedup ของ 3STSC เมื่อเทียบกับ 2STSC ซึ่ง 3STSC จะเป็นการอัพเดทตัวแทนกลุ่มข้อมูลทุกครั้งที่มีข้อมูลลำดับย่อยใหม่เข้ามาโดยจะใช้เวลาคงที่ ส่วน 2STSC จะทำการคำนวณเซทของตัวแทนกลุ่มใหม่ทุกๆ ครั้งที่มีข้อมูลใหม่เข้ามา จะเห็นได้ว่าผลลัพธ์ของอัลกอริทึม 2STSC และ 3STSC นั้นไม่เหมือนกัน เนื่องจากมีอัลกอริทึมการอัพเดทที่ต่างกัน ดังนั้นจึงได้ทำการทดลองเพิ่มเติมเพื่อให้เห็นความแตกต่างกันในผลลัพธ์การจัดกลุ่มระหว่าง 2STSC และ 3STSC ด้วย และการทดลองสุดท้ายจะเป็นการแสดงให้เห็นว่า หากเรามีความสามารถในการคำนวณ และหน่วยเก็บข้อมูลที่พอเพียงแล้ว ผลลัพธ์จากการจัดกลุ่มของ 3STSC จะมีความใกล้เคียงกับผลลัพธ์จากการจัดกลุ่มแบบ 2STSC ชุดข้อมูลที่ใช้ในการทดลองทั้งแปดชุดข้อมูลนี้ นำมาจากคลังข้อมูลอนุกรมเวลา Time Series Data Mining Archives (TSDMA) (Keogh and Folias, 2011) ของ University of California, Riverside ประเทศสหรัฐอเมริกา โดยแต่ละชุดข้อมูลอนุกรมเวลาประกอบด้วย 2000 จุดข้อมูล ตัวอย่างอนุกรมเวลาของแต่ละชุดข้อมูลได้แสดงไว้ในรูปที่ 4.32

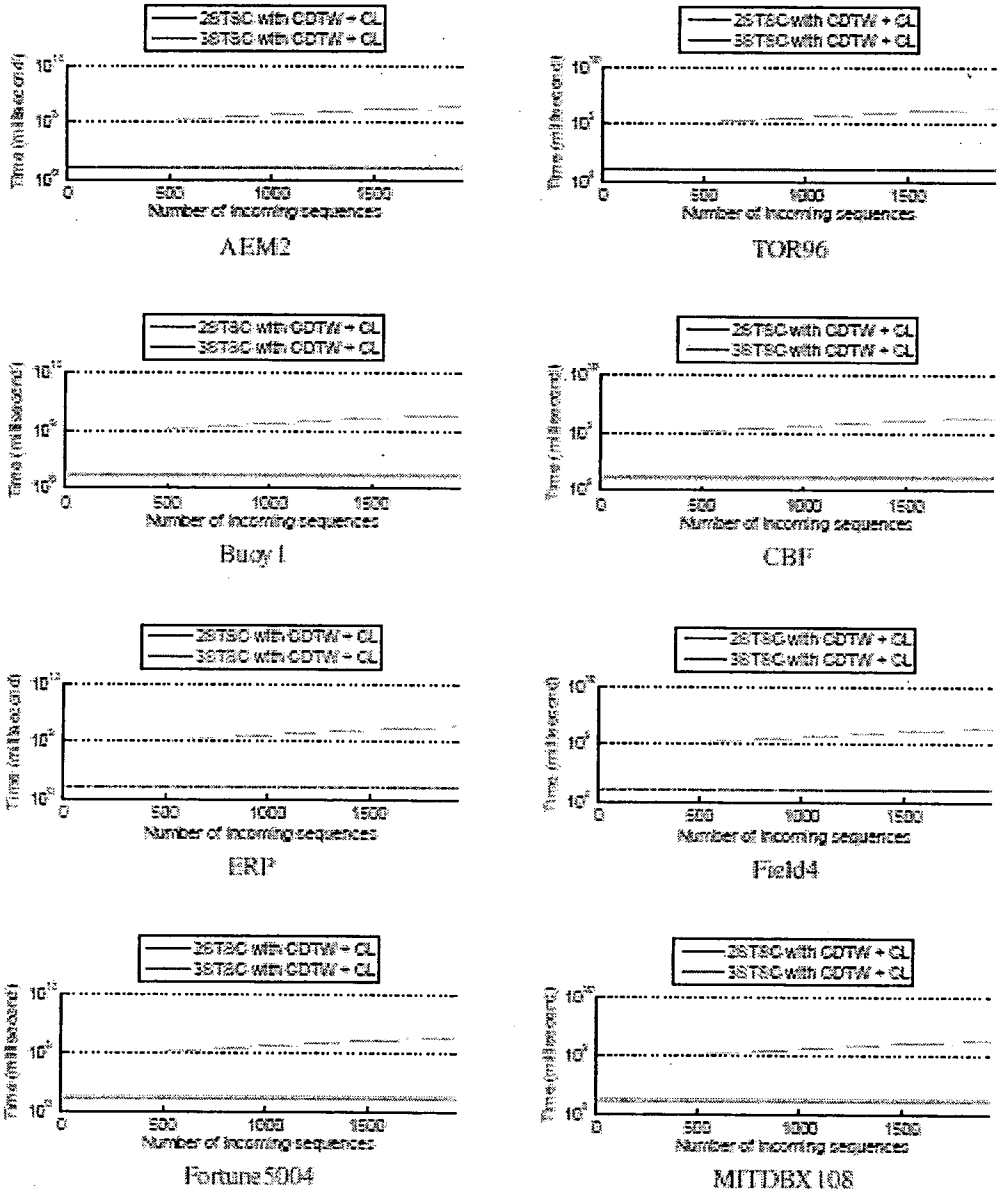


รูปที่ 4. 32 : ตัวอย่างข้อมูลของทั้งแปดชุดข้อมูลที่ใช้ในการทดลอง

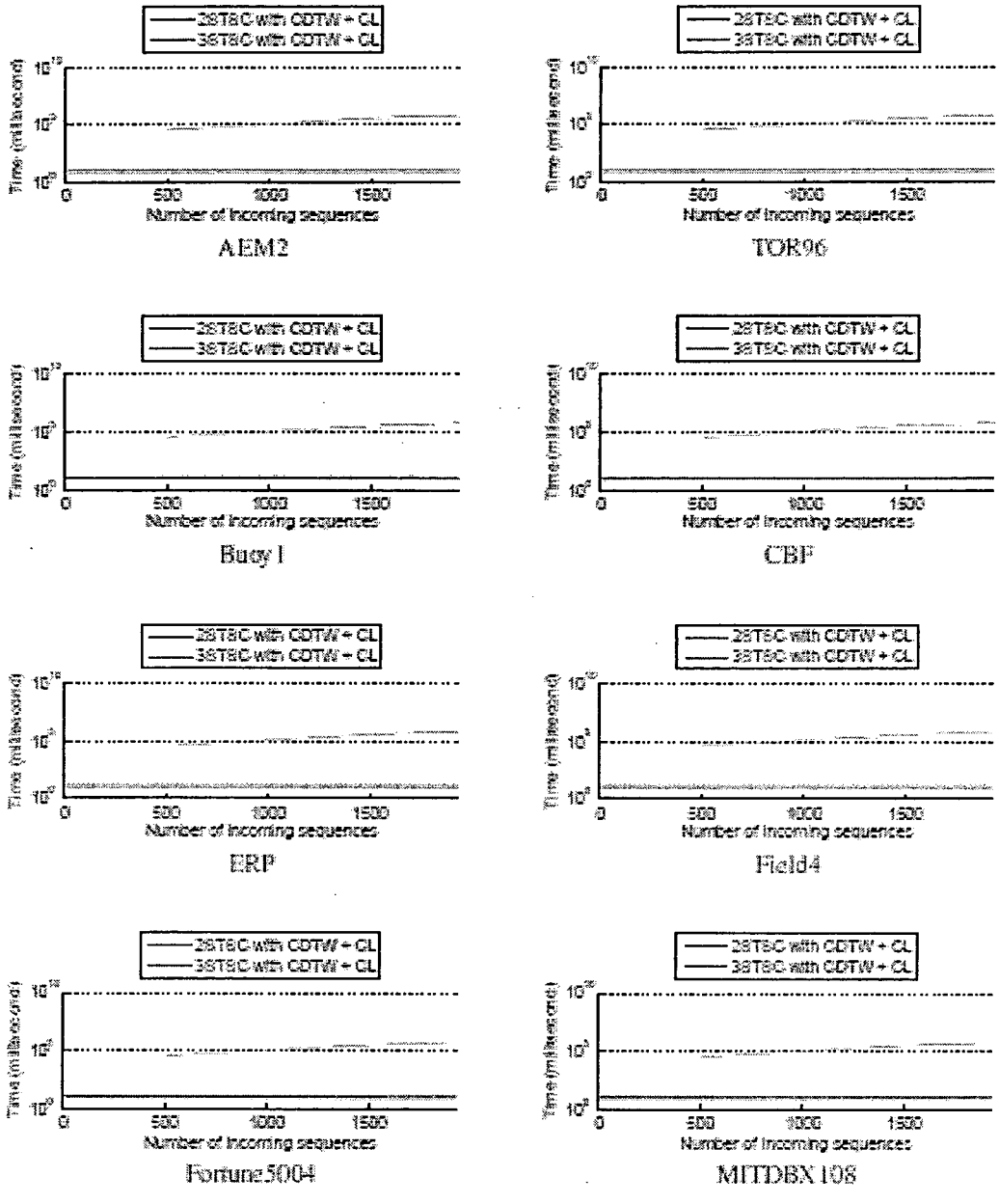
การทดลองที่ 1

การทดลองแรกนี้เป็นการทดลองเพื่อแสดงให้เห็นว่า อัลกอริทึม 3STSC สามารถให้ผลลัพธ์เป็นเซตของกลุ่มข้อมูลได้เร็วกว่าอัลกอริทึมดั้งเดิมที่ใช้ 2STSC โดยที่ทุก ๆ จุดข้อมูลใหม่ที่เพิ่มเข้ามา จะจับเวลาที่ใช้ในการอัพเดทตัวแทนกลุ่มข้อมูลของ 3STSC และอัลกอริทึมดั้งเดิม เพื่อนำมาเปรียบเทียบกัน ซึ่งในการทดลองนี้ได้มีการเปลี่ยนค่าพารามิเตอร์ต่าง ๆ ได้แก่จำนวนของกลุ่มข้อมูล

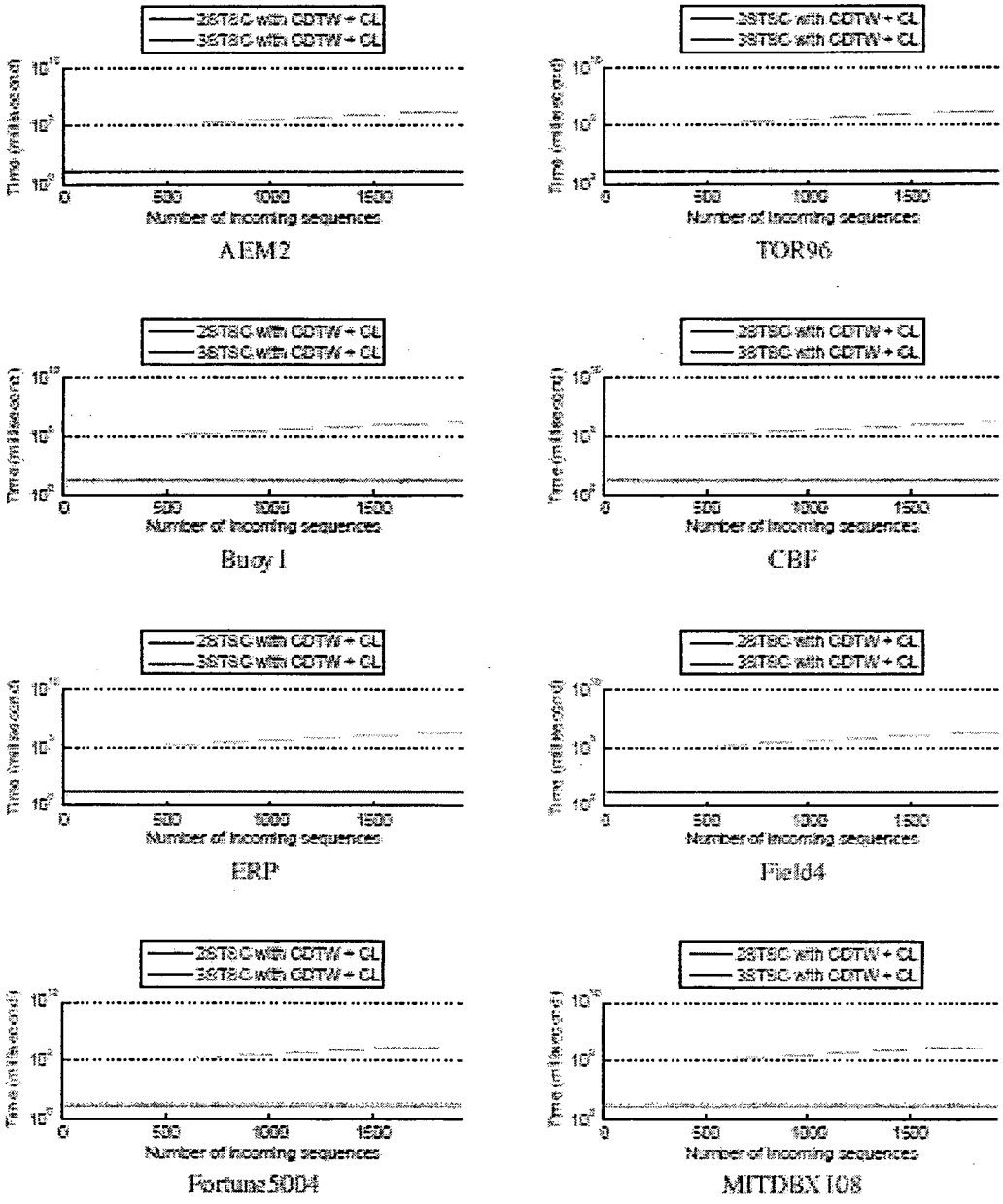
(k) และ ขนาดของ sliding window (w) ส่วนค่าสูงสุดที่อนุญาตให้เก็บลำดับย่อยไว้ (α) กำหนดให้ใช้ค่าเท่ากับจำนวนของกลุ่มข้อมูล นอกจากนี้ได้ทดลองใช้ฟังก์ชันการคำนวณระยะทางระหว่างกลุ่มข้อมูลทั้งสองแบบคือ complete linkage และ average linkage และใช้ฟังก์ชันการเฉลี่ยทั้งสองแบบเช่นกันคือ CDTW และ ICDTW โดยผลการเปรียบเทียบเวลาที่ใช้ในการคำนวณระหว่าง 3STSC และ 2STSC จากการปรับพารามิเตอร์ต่างๆ ได้แสดงในรูปที่ 3.xx – 3.xx



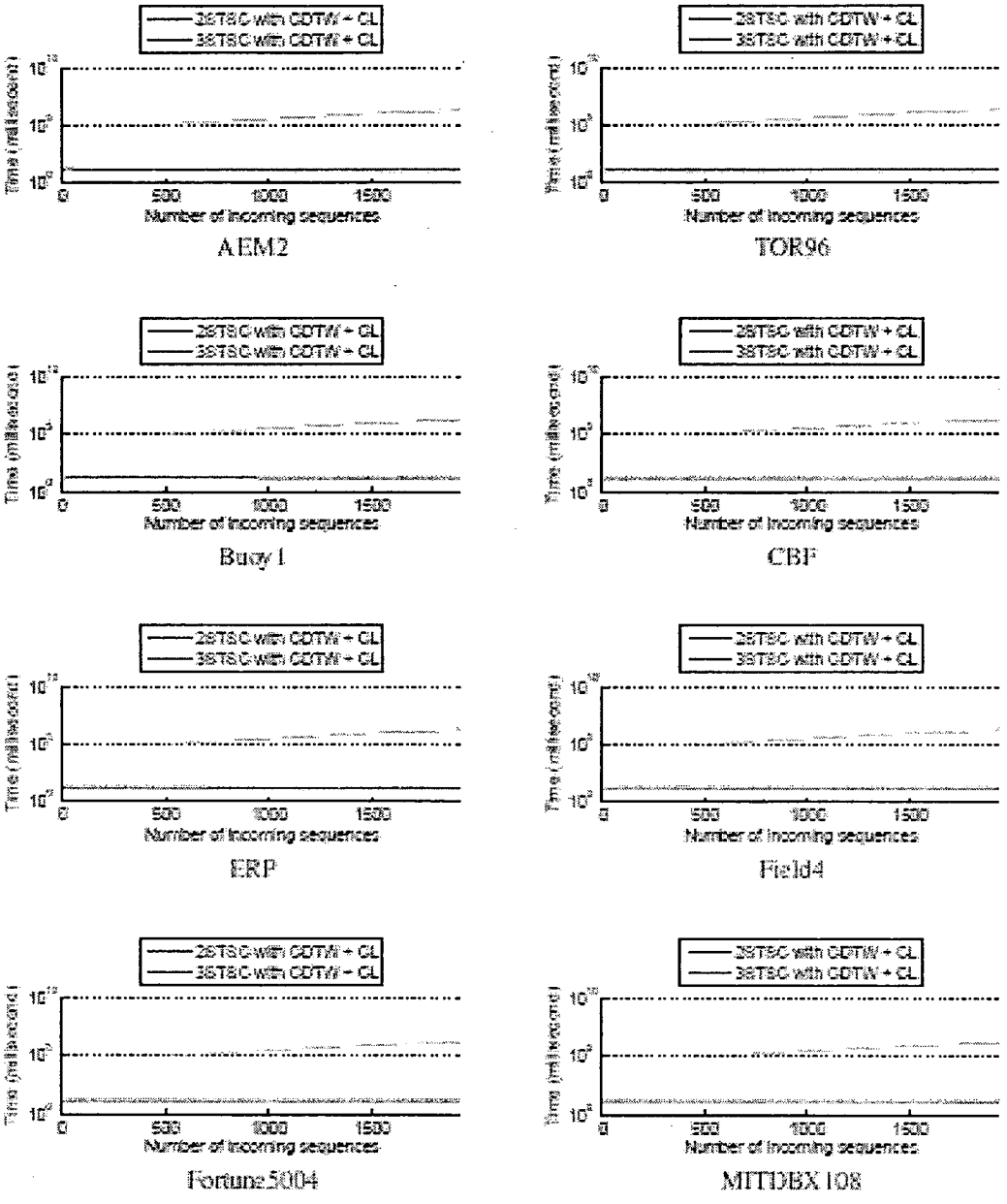
รูปที่ 4. 33 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน CDTW และ complete linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 3 และ w = 64



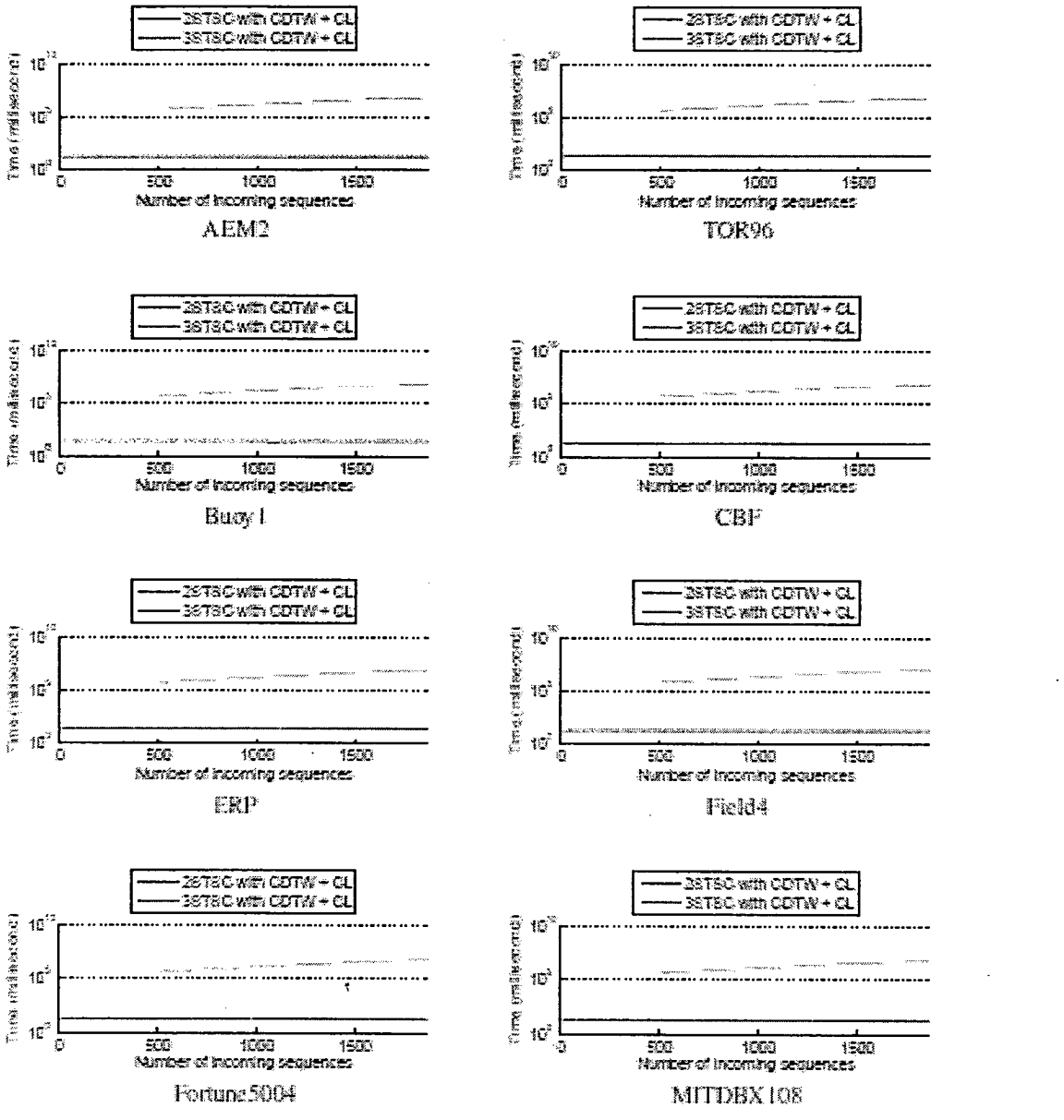
รูปที่ 4. 34 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน CDTW และ complete linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 3 และ w = 32



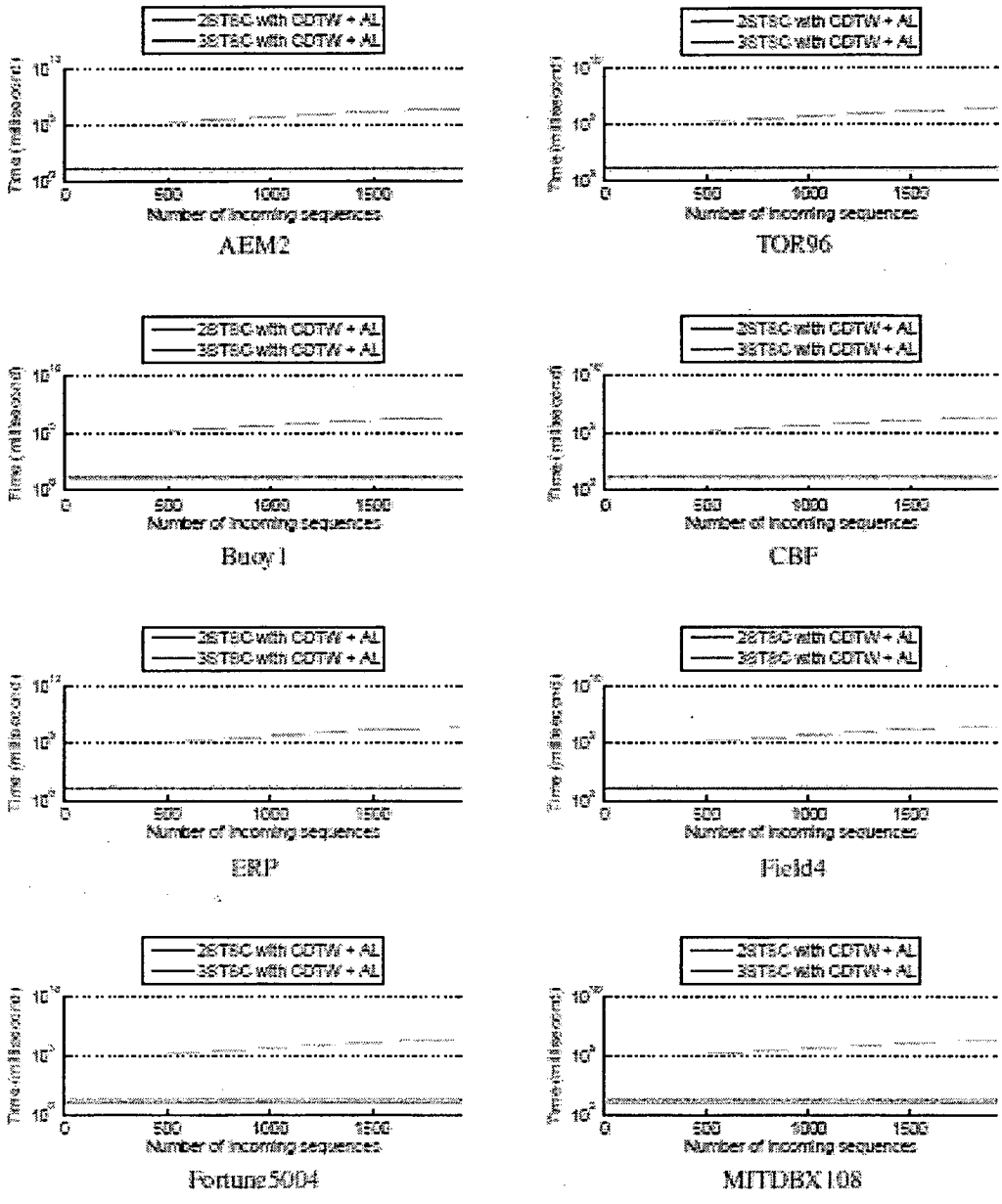
รูปที่ 4. 35 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน CDTW และ complete linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 5 และ w = 64



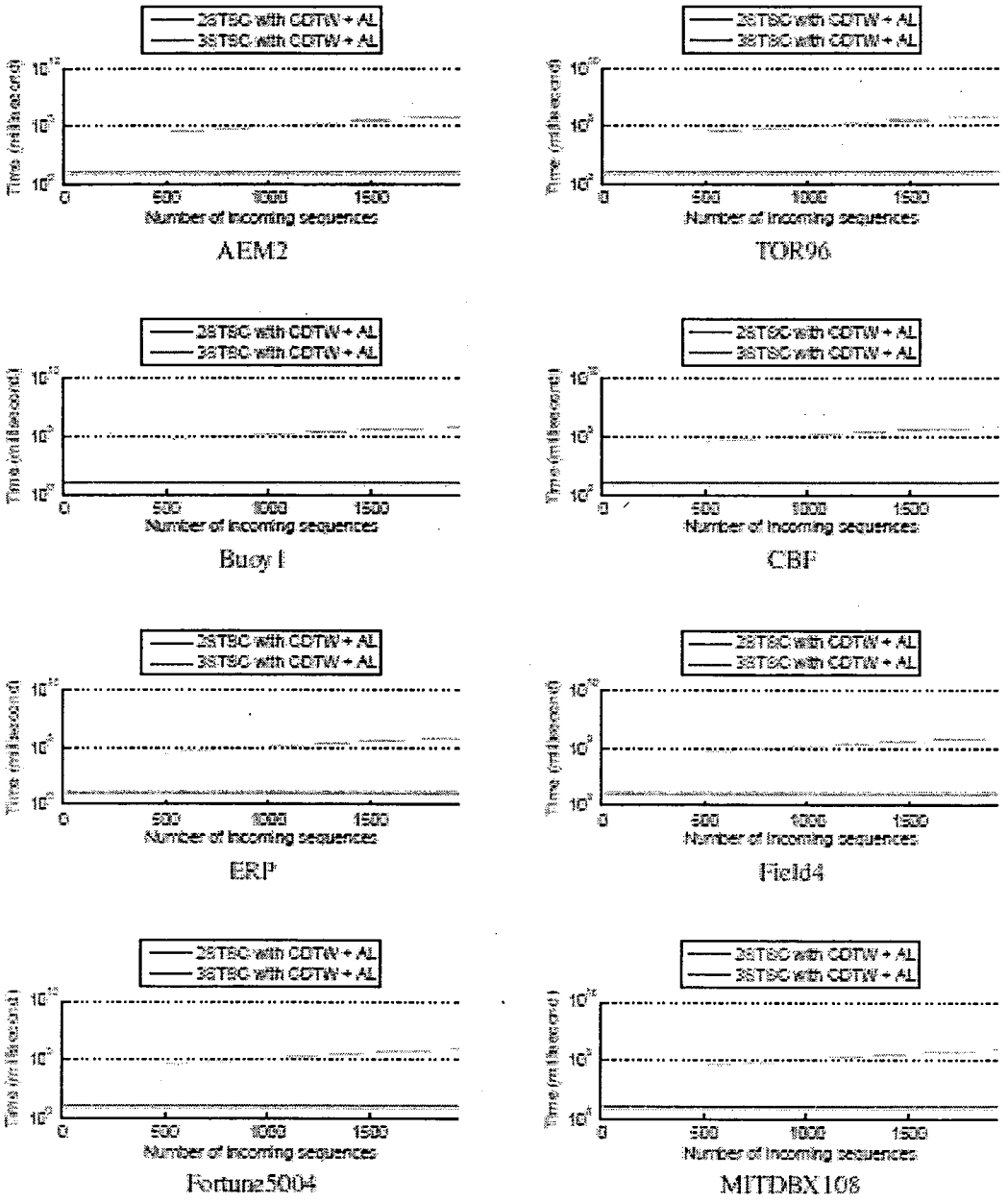
รูปที่ 4. 36 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน CDTW และ complete linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 7 และ w = 64



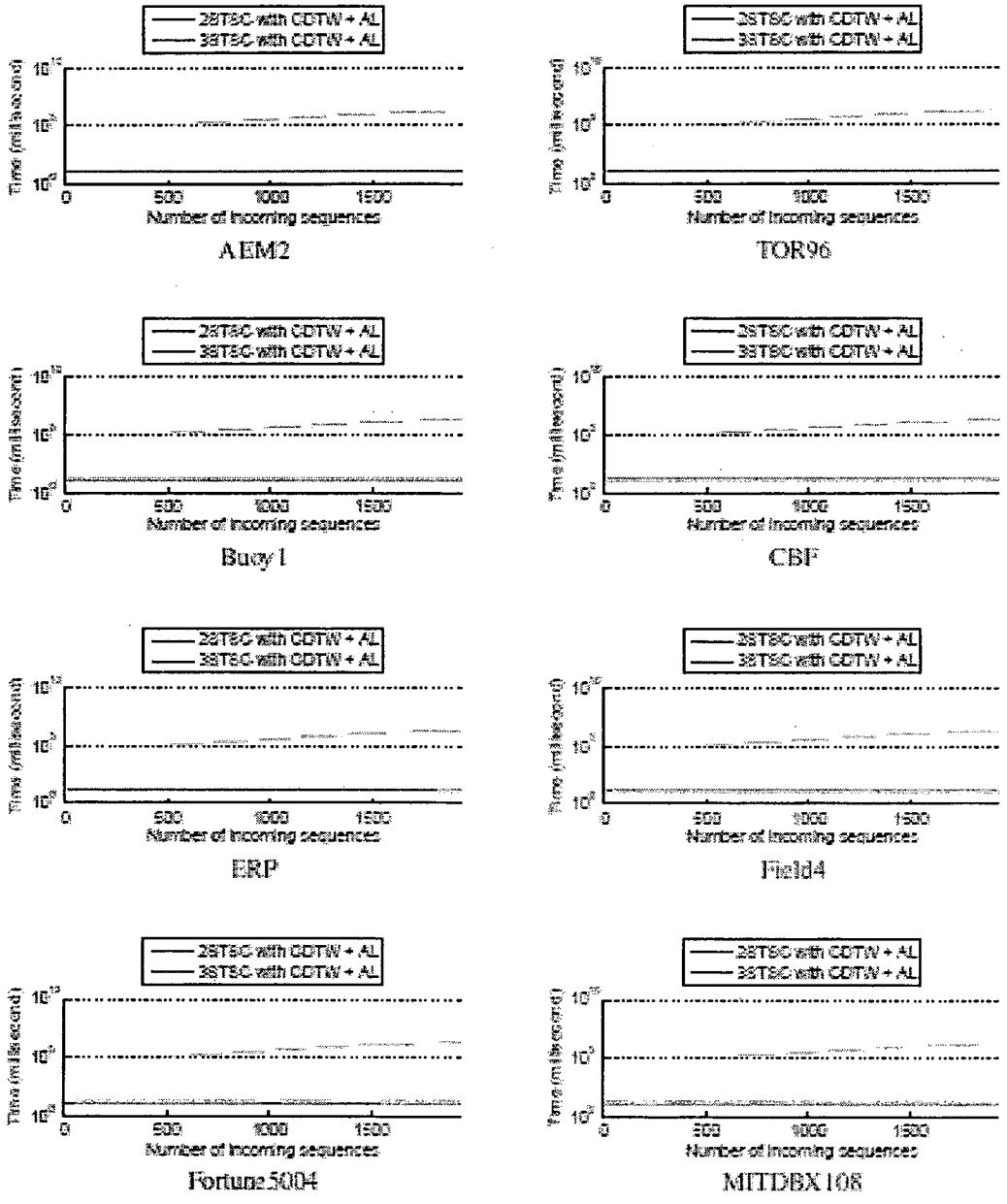
รูปที่ 4. 37 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน CDTW และ complete linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 3 และ w = 128



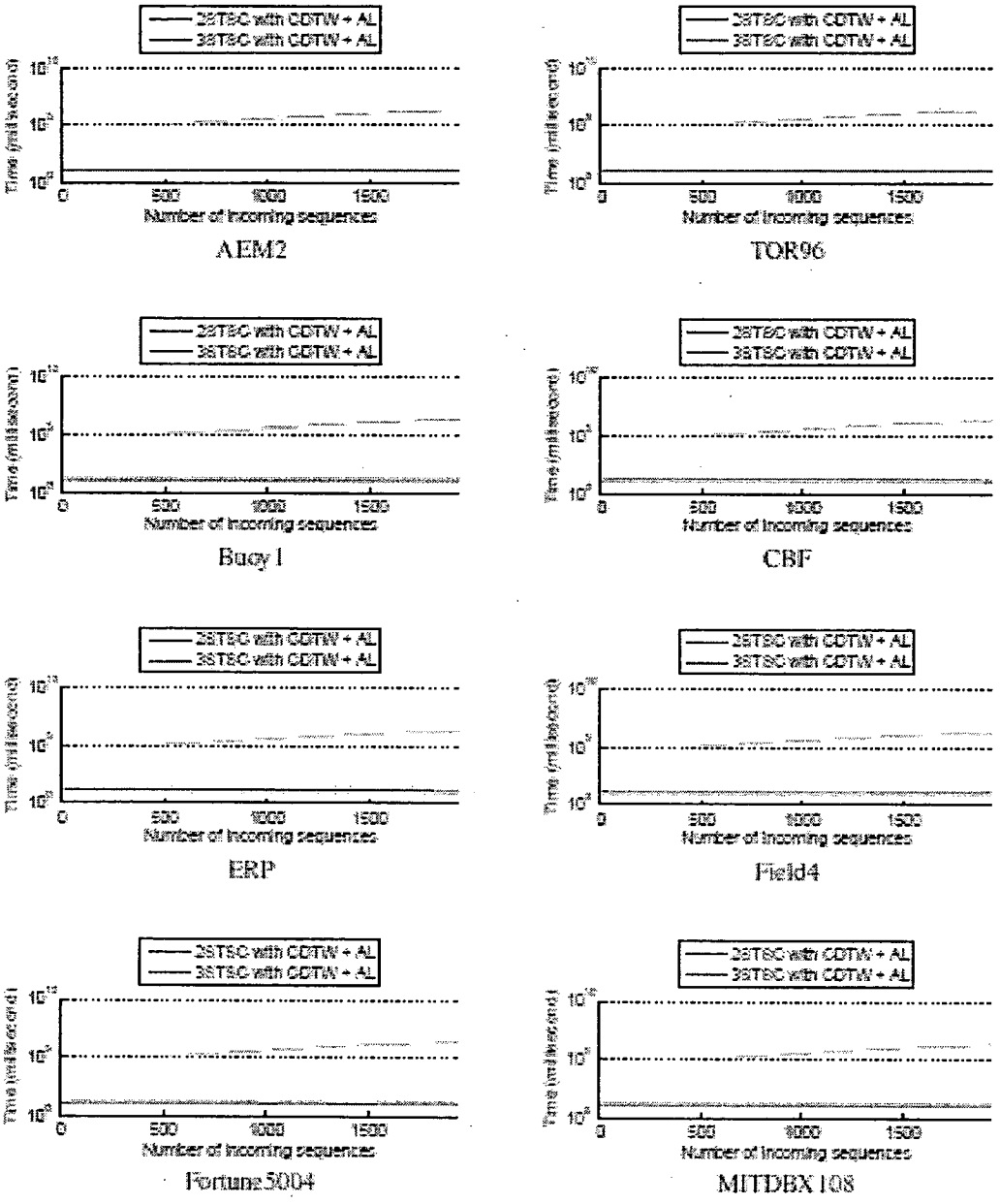
รูปที่ 4. 38 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน CDTW และ average linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 3 และ w = 64



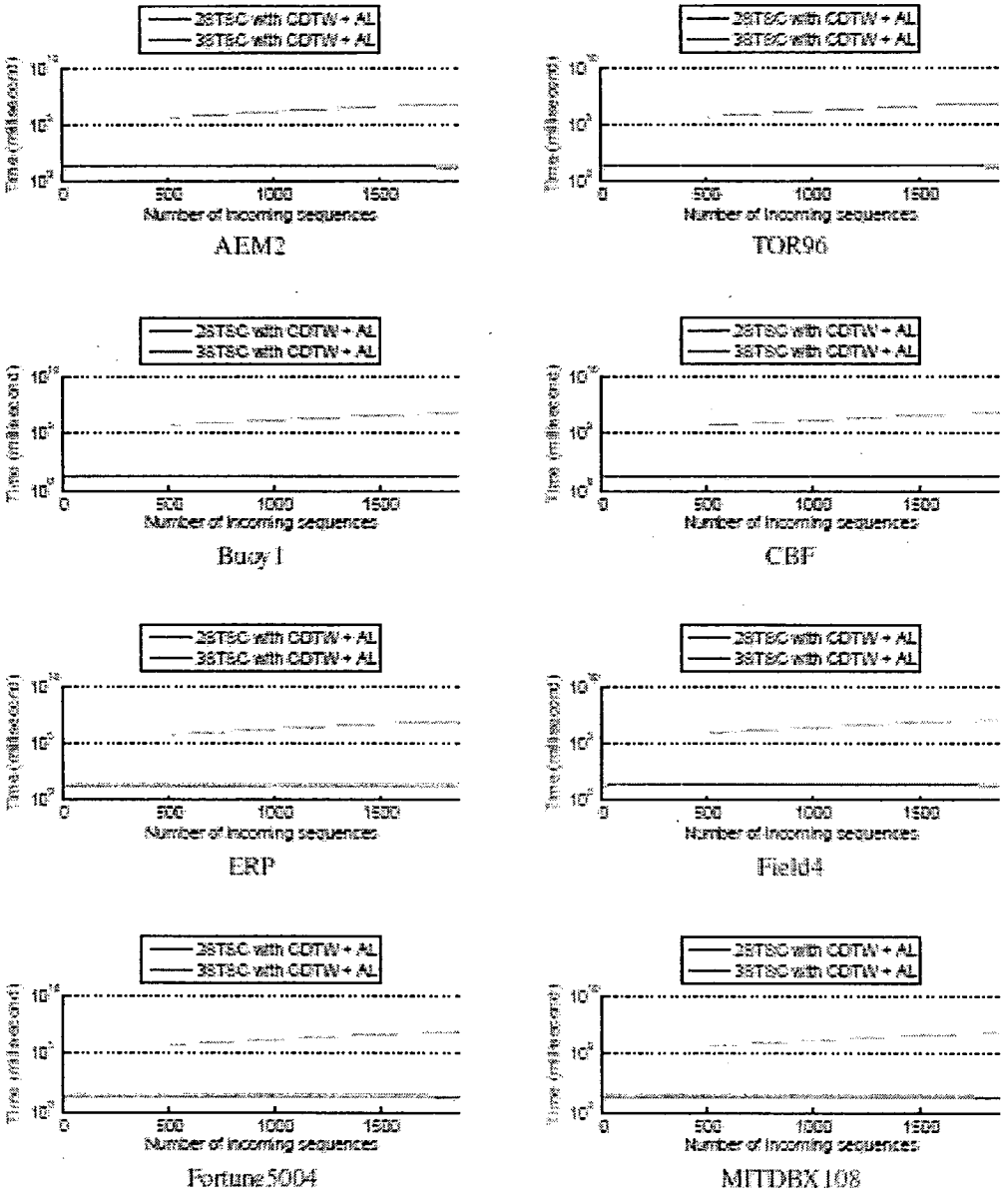
รูปที่ 4. 39 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน CDTW และ average linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 3 และ w = 32



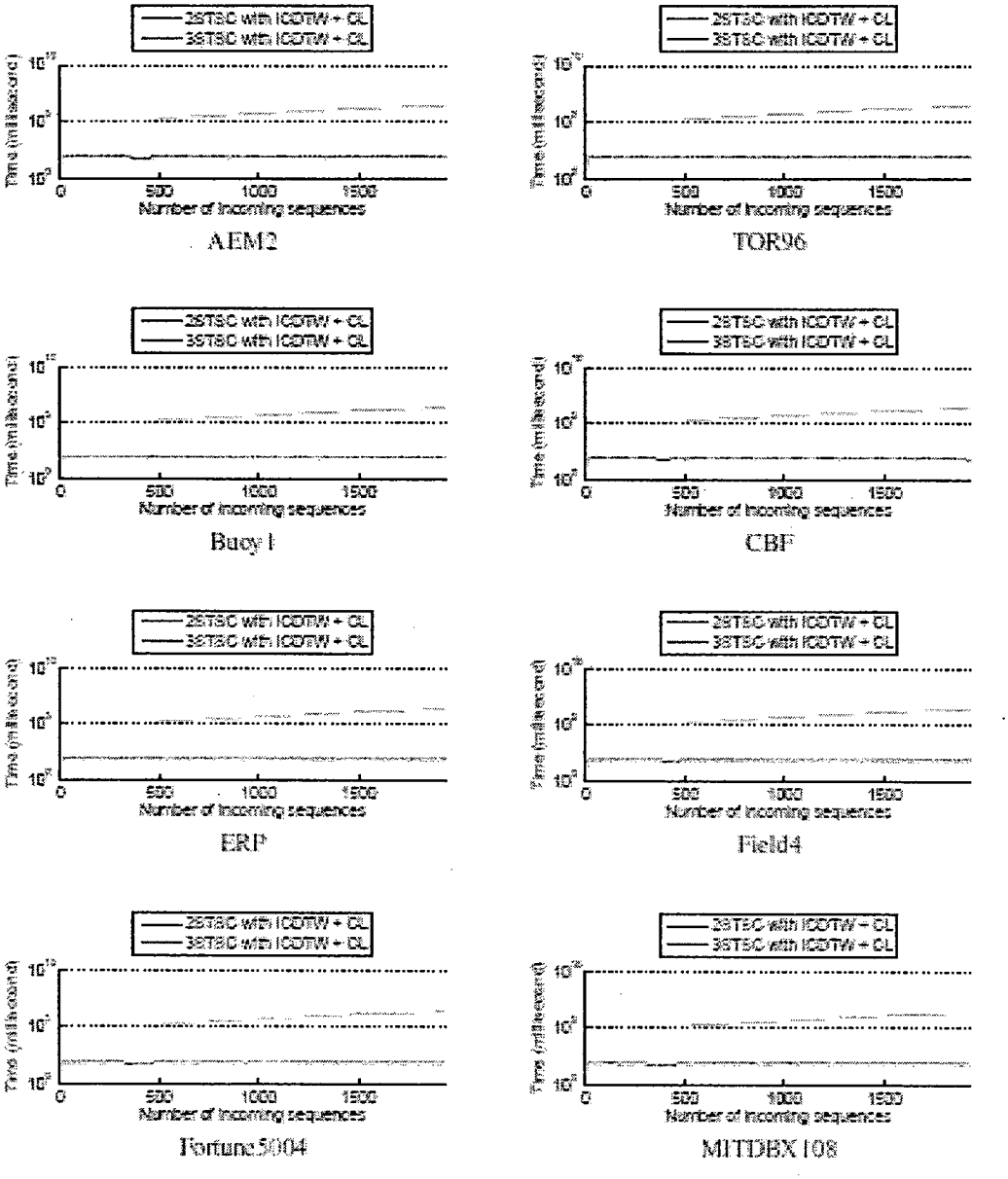
รูปที่ 4. 40 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน CDTW และ average linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 5 และ w = 64



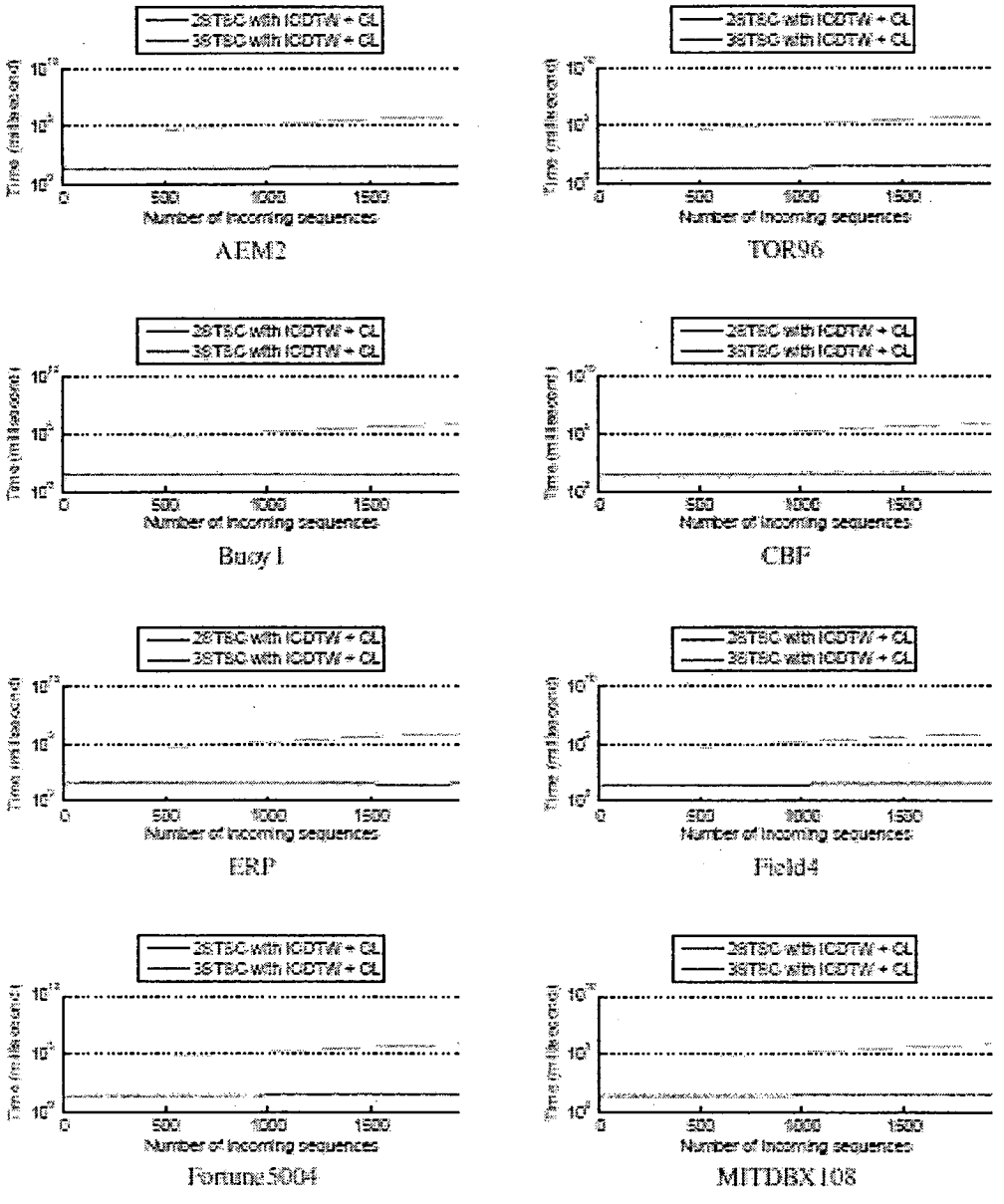
รูปที่ 4. 41 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน CDTW และ average linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 7 และ w = 64



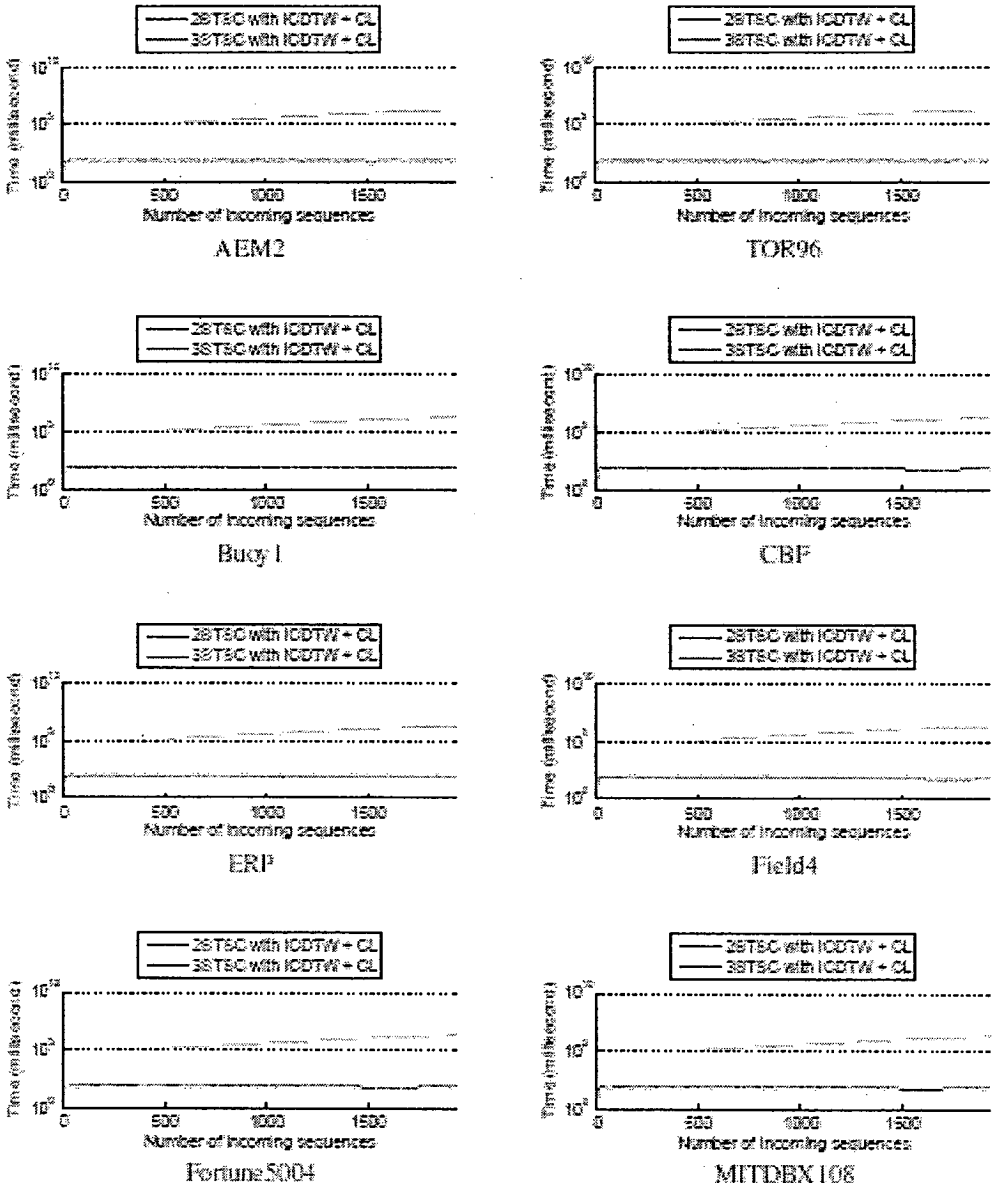
รูปที่ 4. 42 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน CDTW และ average linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 3 และ w = 128



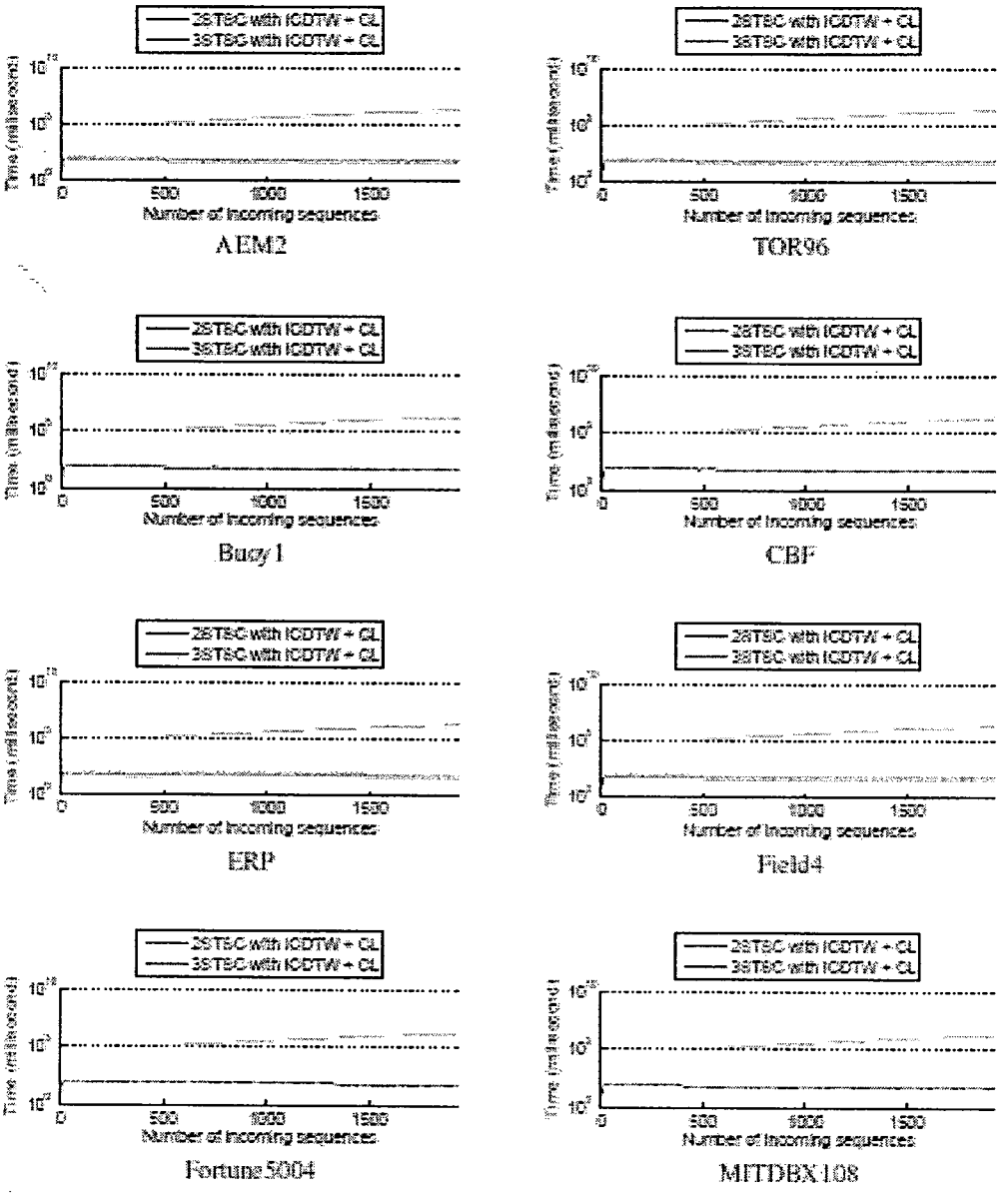
รูปที่ 4. 43 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน ICDTW และ complete linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 3 และ w = 64



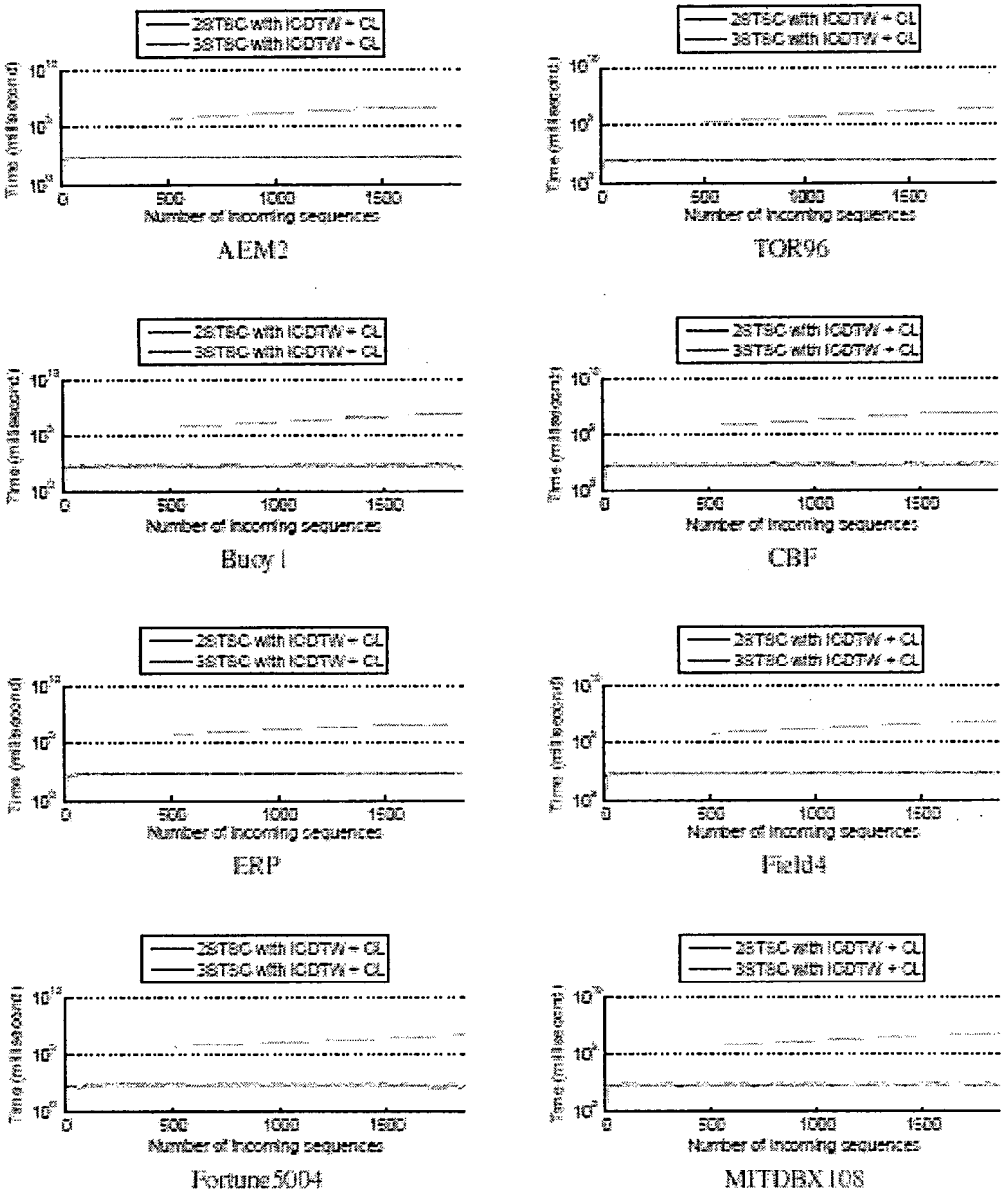
รูปที่ 4. 44 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน ICDTW และ complete linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 3 และ w = 32



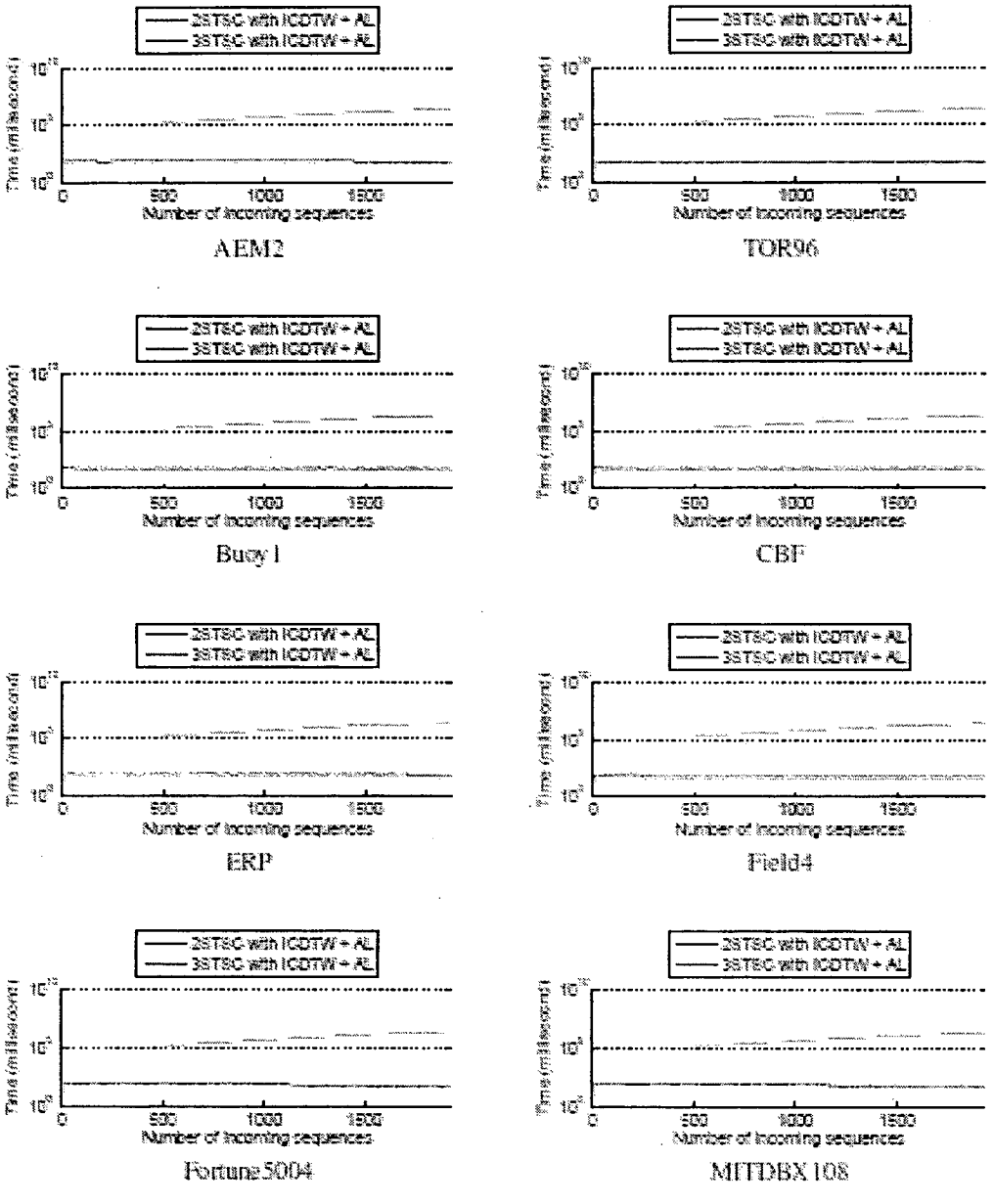
รูปที่ 4. 45 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน ICDTW และ complete linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 5 และ w = 64



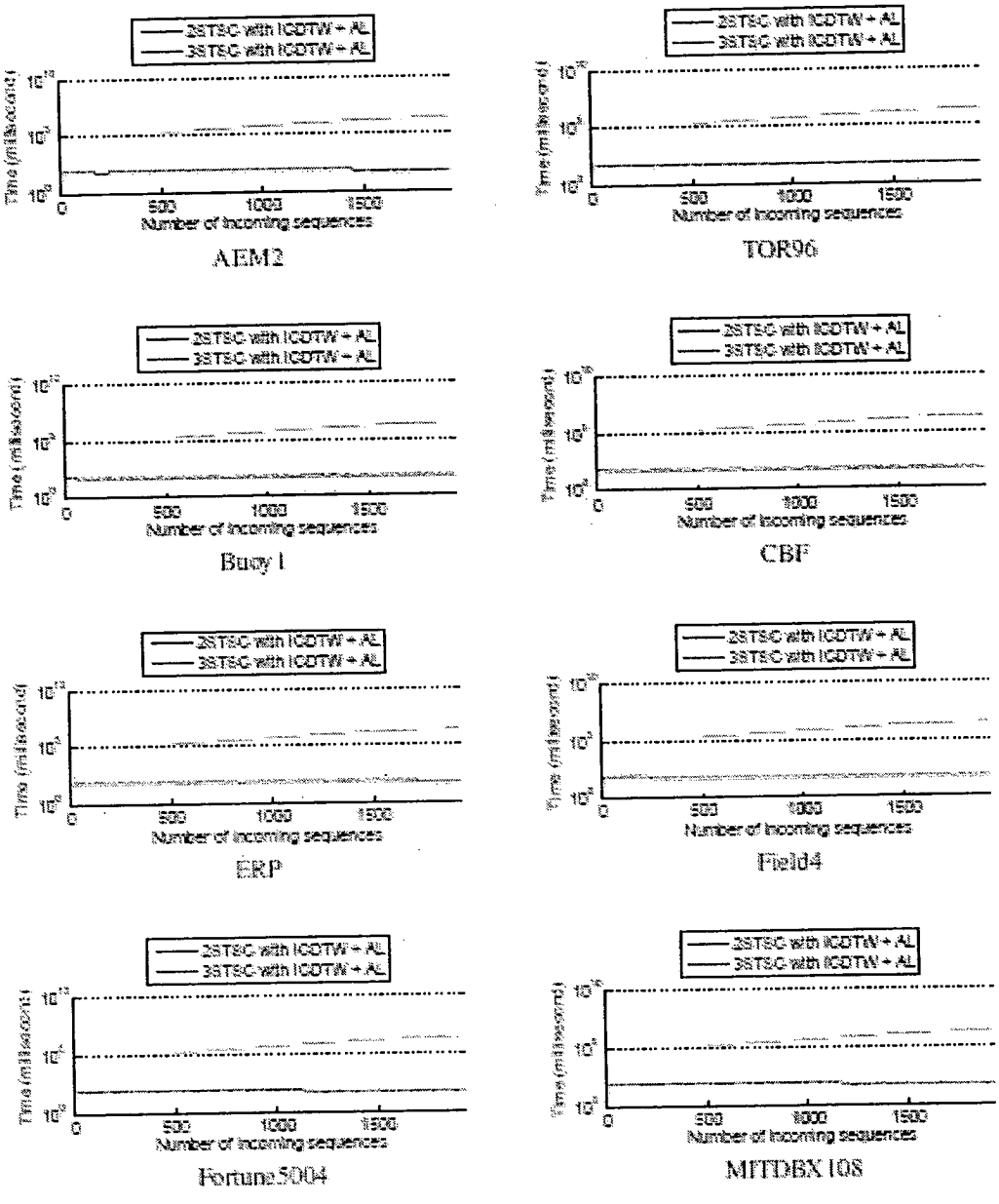
รูปที่ 4. 46 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน ICDTW และ complete linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 7 และ w = 64



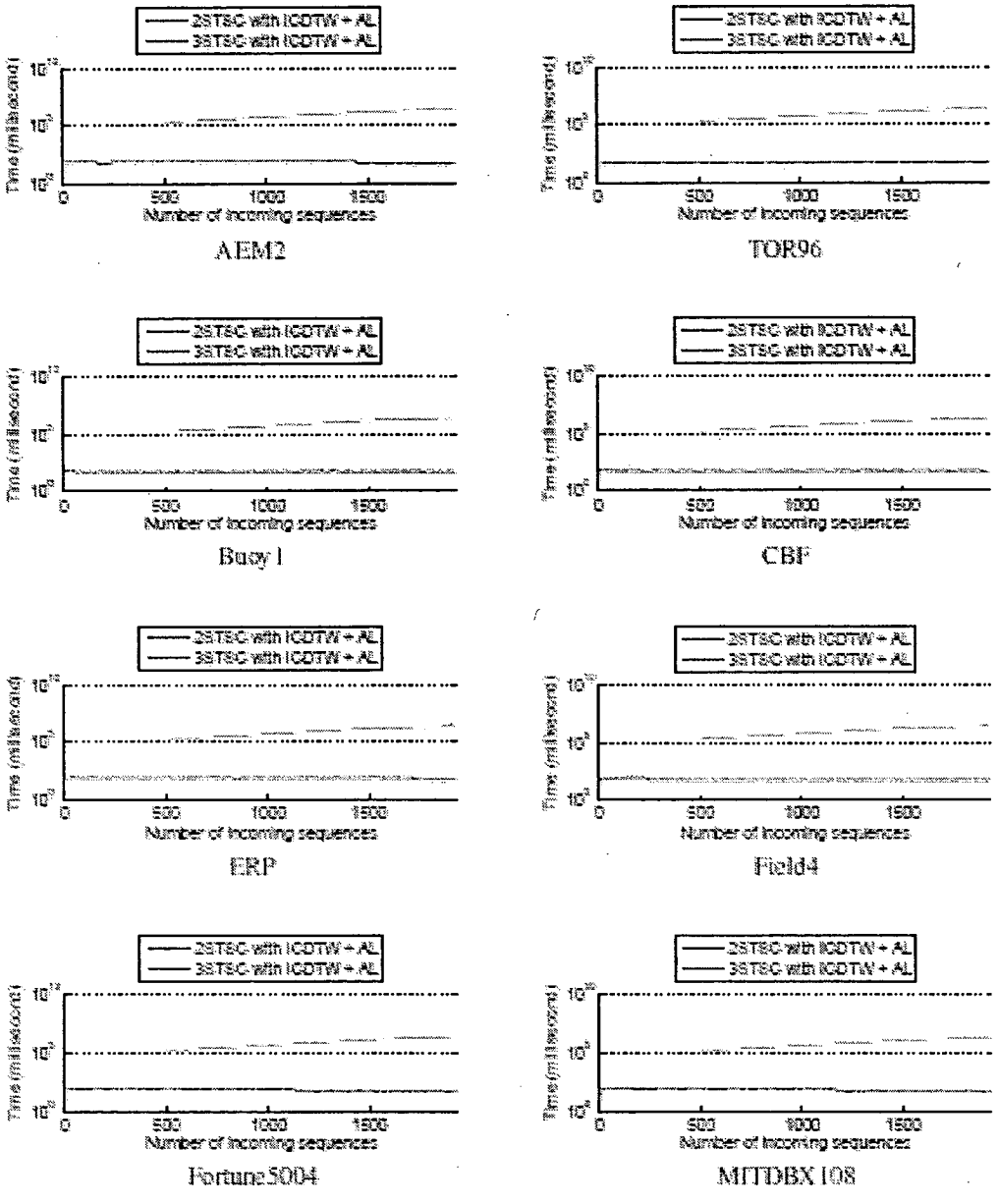
รูปที่ 4. 47 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน ICDTW และ complete linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 3 และ w = 128



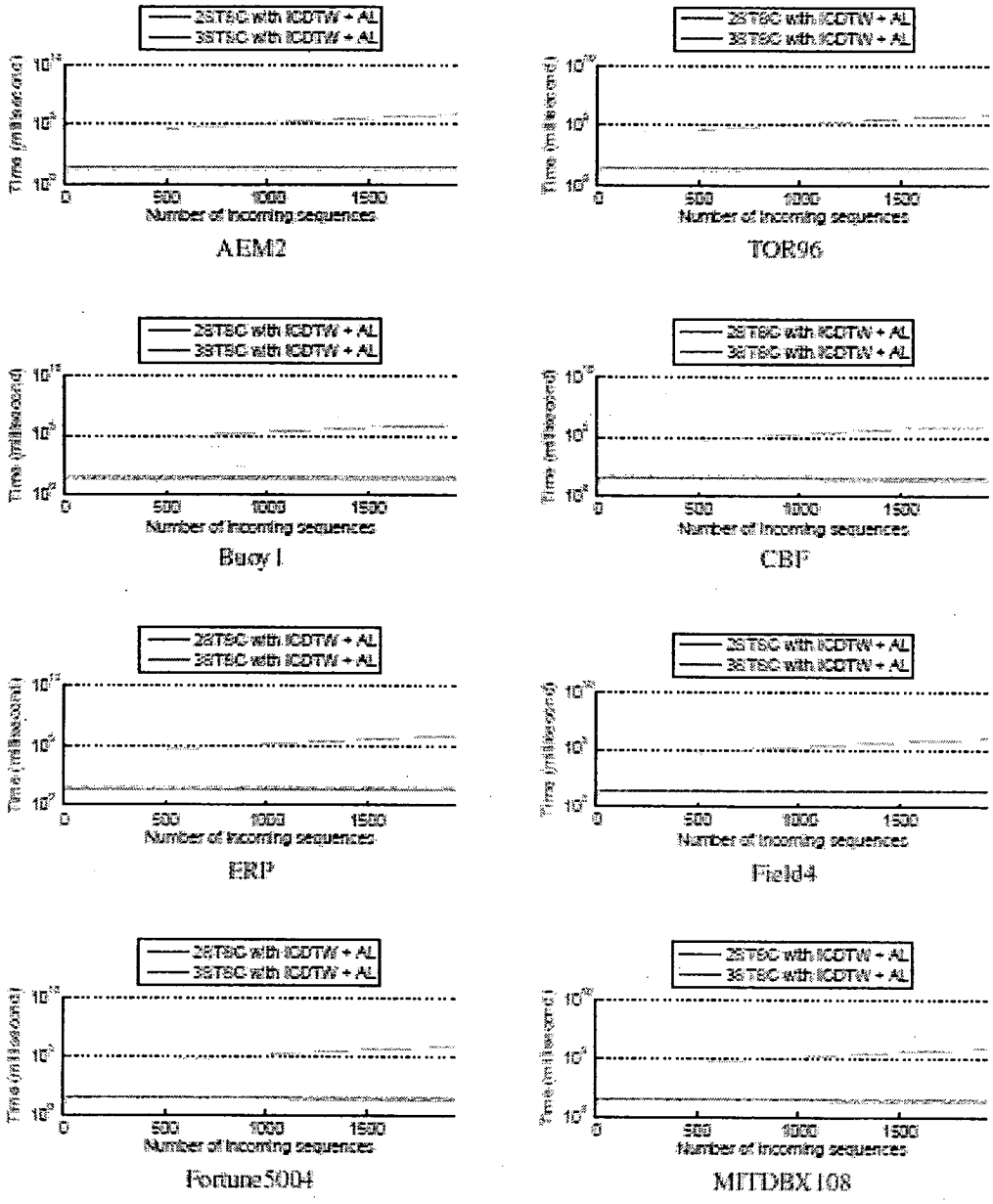
รูปที่ 4. 48 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน ICDTW และ average linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 3 และ w = 64



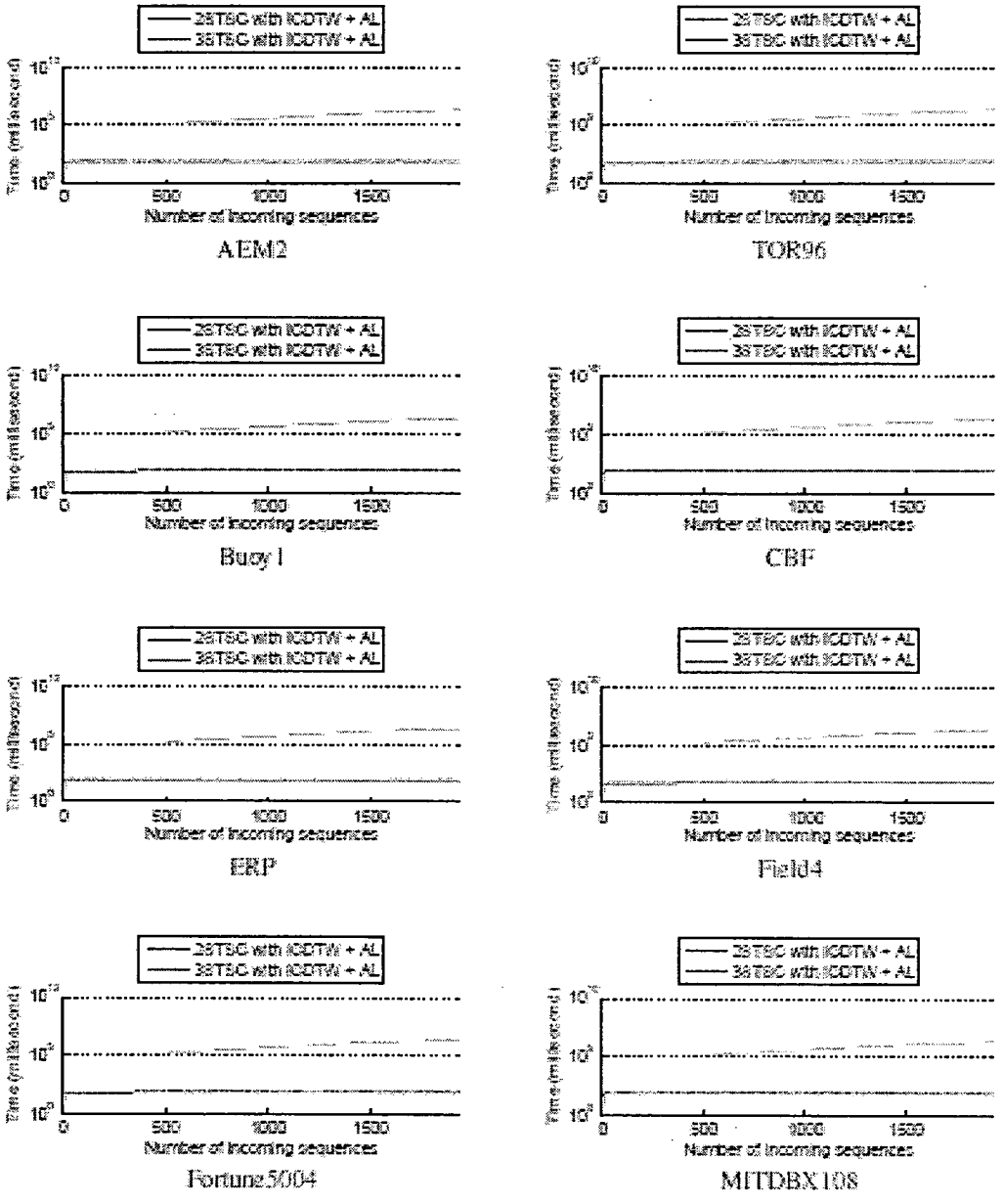
รูปที่ 4. 49 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน ICDTW และ average linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 3 และ w = 32



รูปที่ 4. 50 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน ICDTW และ average linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 5 และ w = 64



รูปที่ 4. 51 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน ICDTW และ average linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 7 และ w = 64



รูปที่ 4. 52 : เปรียบเทียบเวลาที่ใช้ในการคำนวณของ 3STSC และ 2STSC โดยใช้ฟังก์ชัน ICDTW และ average linkage เมื่อมีลำดับย่อยใหม่เพิ่มเข้ามา กำหนดให้ k = 3 และ w = 128

การทดลองที่ 2

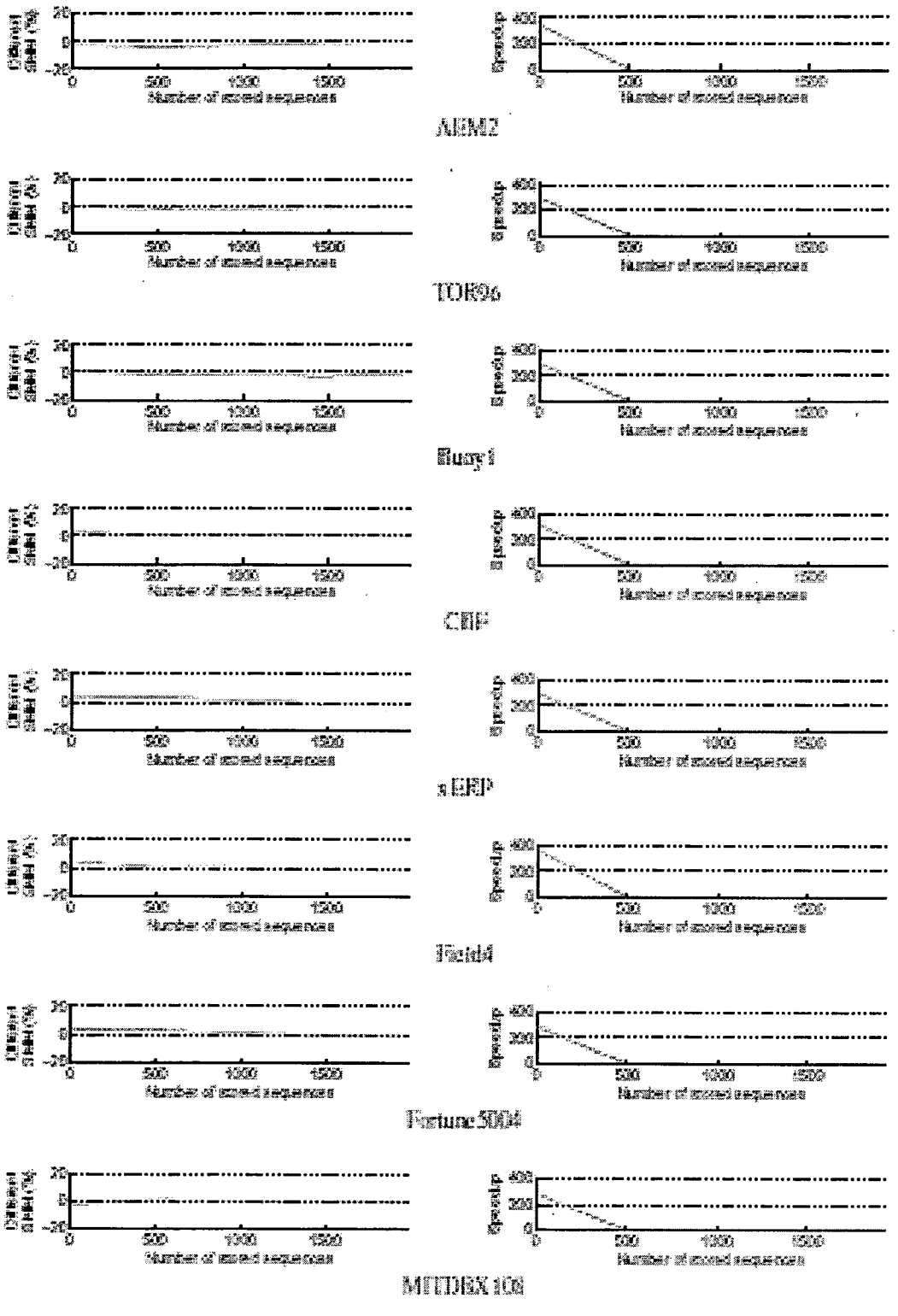
การทดลองที่ 2 นี้เป็นการทดลองเพื่อแสดงให้เห็นถึงประสิทธิผลของผลลัพธ์การจัดกลุ่มจากอัลกอริทึม 3STSC โดยแปรค่าจำนวนของลำดับย่อยสูงสุดที่ยอมให้เก็บไว้ รวมถึงจำนวนของกลุ่มข้อมูล และขนาดของ sliding window ซึ่งจะเห็นได้จากผลการทดลองว่าประสิทธิผลของผลลัพธ์จะดีขึ้นเรื่อย ๆ หากมีศักยภาพในการคำนวณและพื้นที่จัดเก็บที่เพิ่มมากขึ้น หากแต่เวลาที่ใช้ในการคำนวณก็จะเพิ่มขึ้นเป็นเงาตามตัว ทั้งนี้ในการวัดผล จะใช้มาตรวัดแบบ Shape-based Meaningfulness Measurement (SMM) ซึ่งมีนิยามดังนี้

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

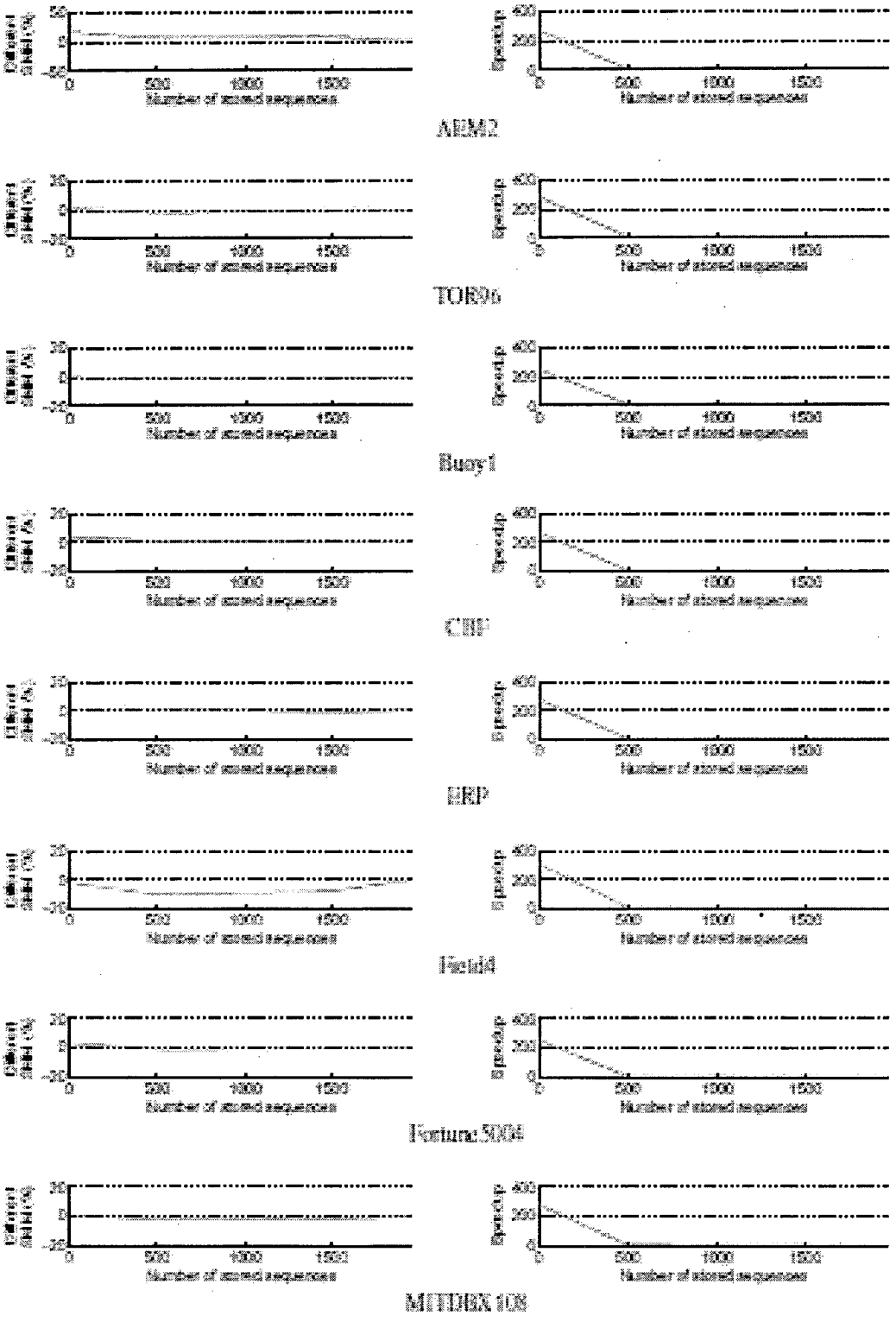
where $Distance(S_i, R_j)$ is a DTW distance between two sequences S_i and R_j.

ค่าของ SMM จะอยู่ในช่วง (0 ... ∞) และเป็นค่าสัมพัทธ์ ซึ่งจะต้องนำค่า SMM ที่ได้จากสองอัลกอริทึมที่ใช้พารามิเตอร์และชุดข้อมูลชุดเดียวกันมาเปรียบเทียบกันว่า อัลกอริทึมการจัดกลุ่มลำดับย่อยแบบใดที่ให้ค่าผลลัพธ์การจัดกลุ่มที่มีความหมายมากกว่ากัน

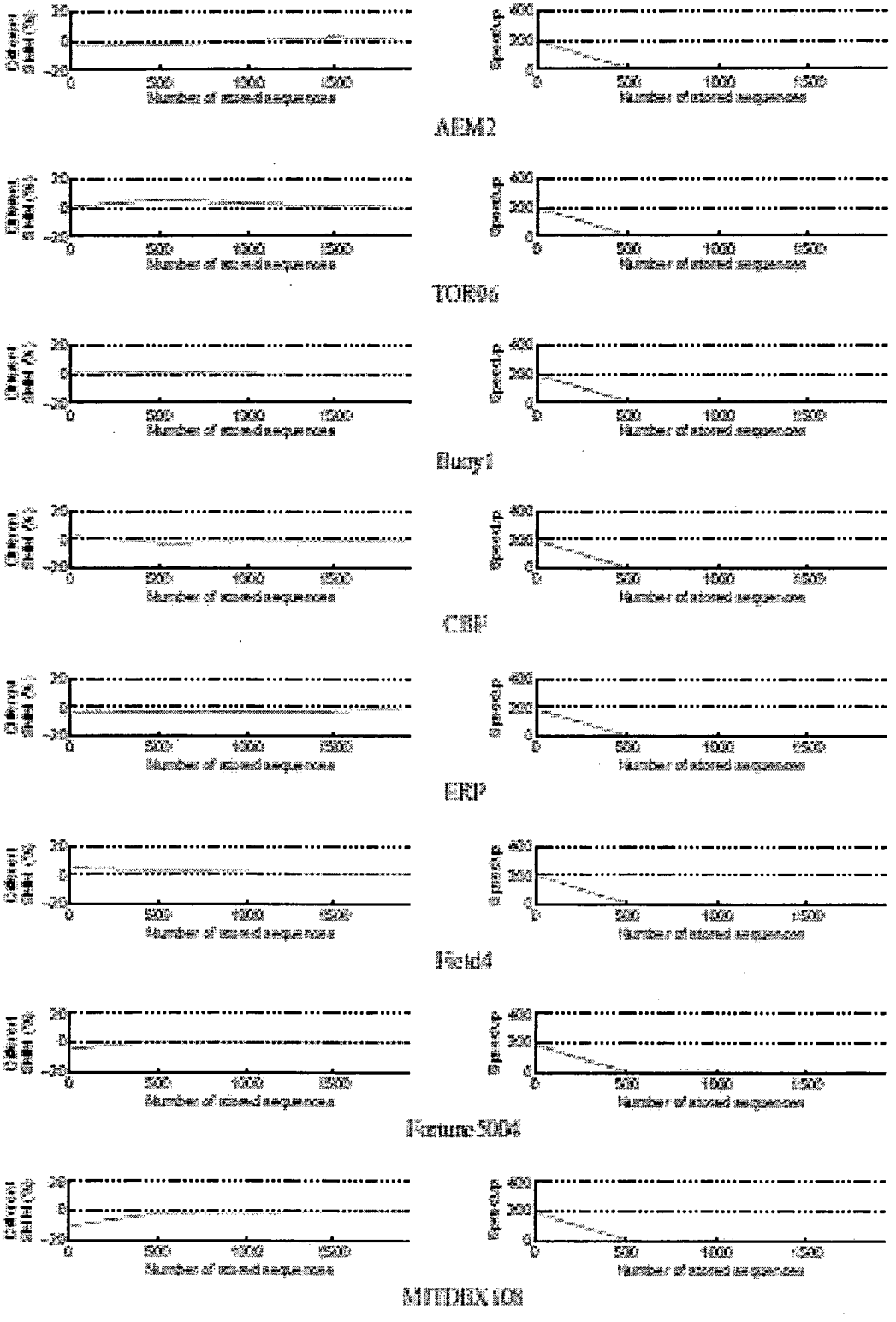
ในการทดลองนี้ จะใช้ฟังก์ชันในการหาระยะทางสำหรับการจัดกลุ่มแบบลำดับขั้น (k-hierarchical clustering) สองแบบ คือ complete linkage และ average linkage และใช้ฟังก์ชันการเฉลี่ยสองฟังก์ชัน คือ CDTW และ ICDTW รูปที่ 4.53 – 4.57, รูปที่ 4.58 – 4.62, รูปที่ 4.63 – 4.67 และ รูปที่ 4.68 – 4.72 แสดงเวลาที่ใช้สำหรับอัลกอริทึม 3STSC สำหรับตัวแปรเซตต่าง ๆ คือ (complete linkage, CDTW), (average linkage, CDTW), (complete linkage, ICDTW) และ (average linkage, ICDTW) ตามลำดับ ซึ่งจะเห็นได้จากผลการทดลองว่า ค่า SMM ของทั้ง 3STSC และ 2STSC นั้นมีค่าที่ใกล้เคียงกัน ซึ่งตีความได้ว่าอัลกอริทึม 3STSC สามารถให้ผลลัพธ์การจัดกลุ่มที่มีความหมาย โดยที่สามารถใช้เวลาลดลงกว่า 400 เท่า



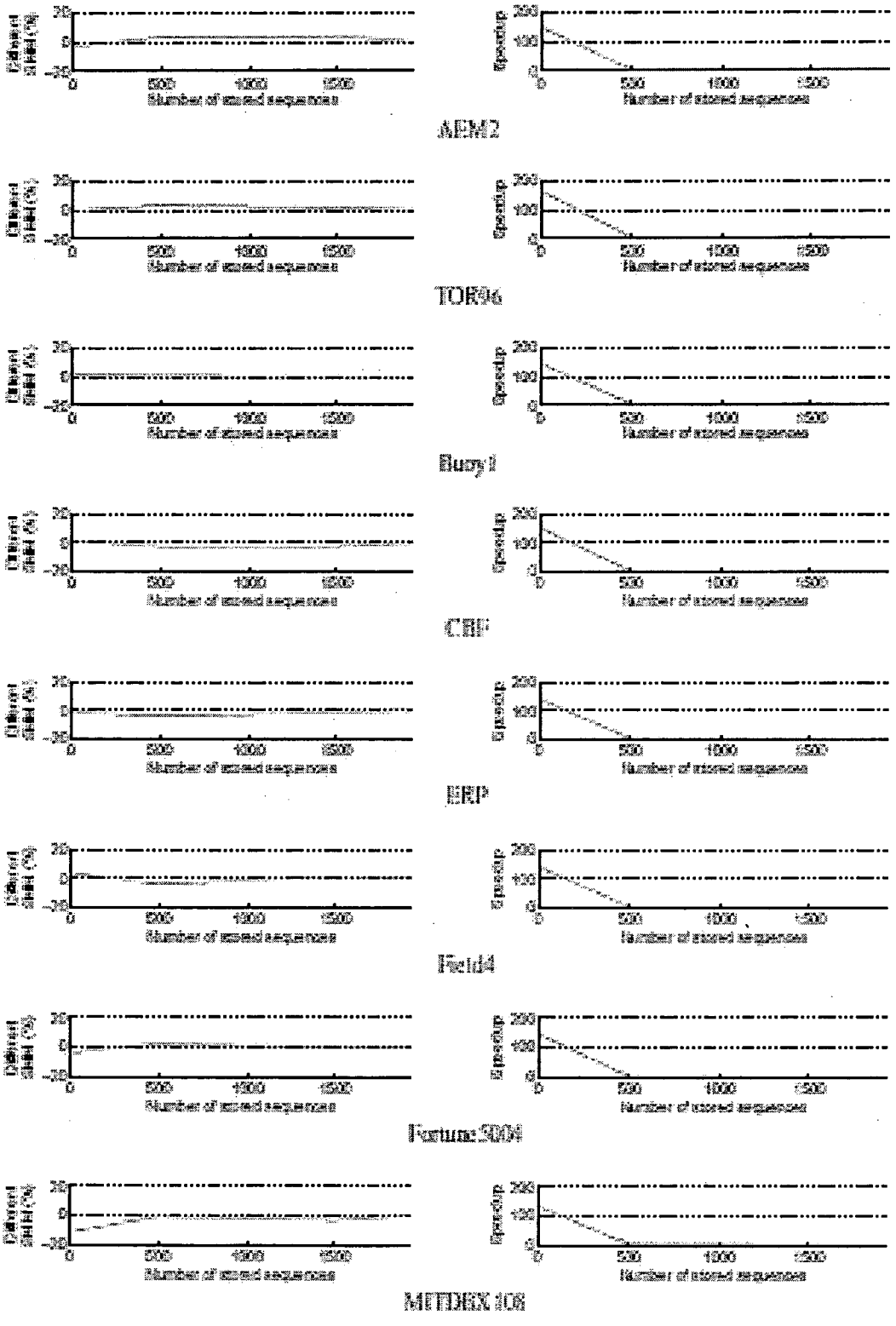
รูปที่ 4. 53 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน CDTW และ complete linkage โดยใช้ค่า k=3, w=64 และจำนวนลำดับย่อยที่ต่างๆ กัน



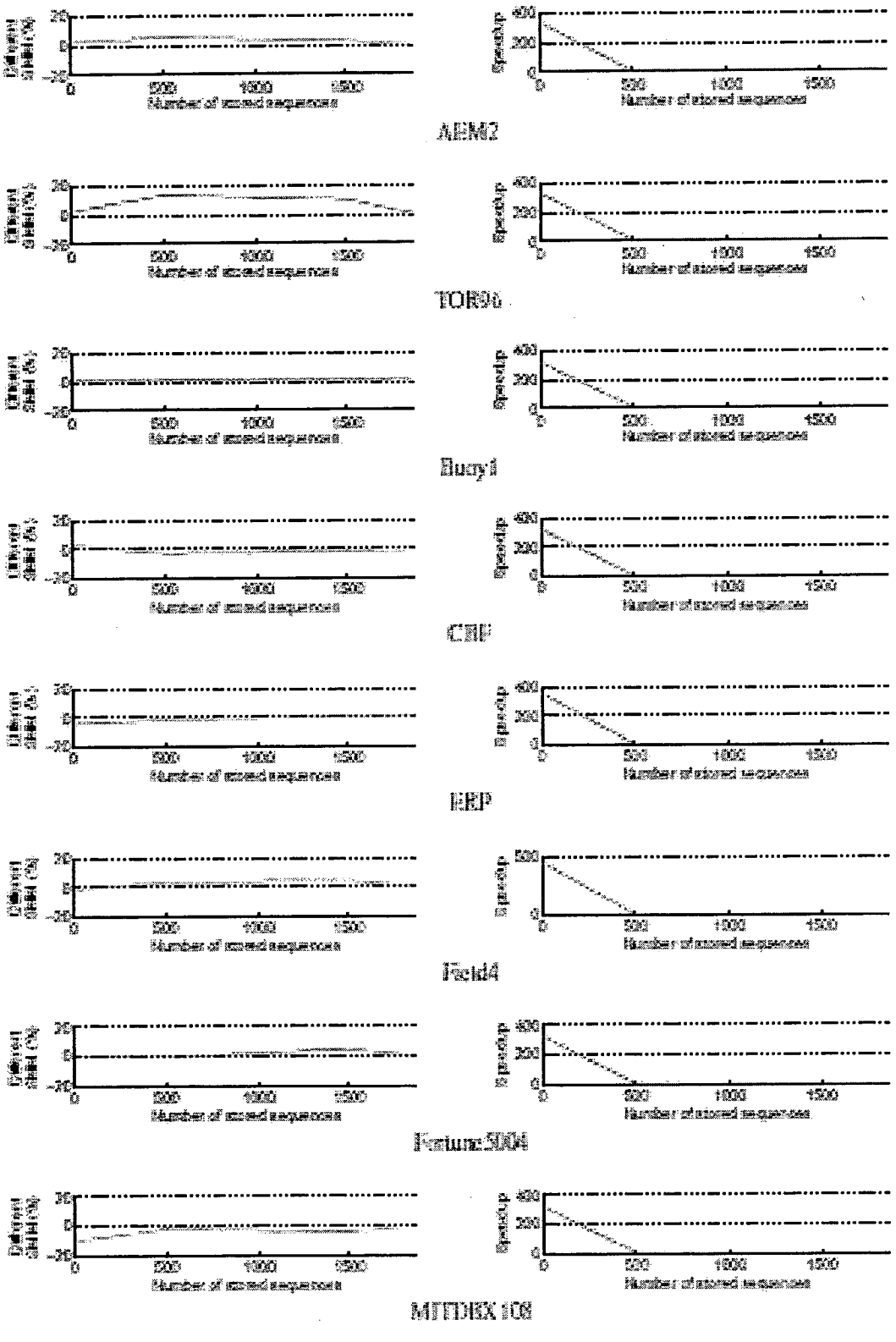
รูปที่ 4. 54 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน CDTW และ complete linkage โดยใช้ค่า k=3, w=32 และจำนวนลำดับย่อยที่ต่างๆ กัน



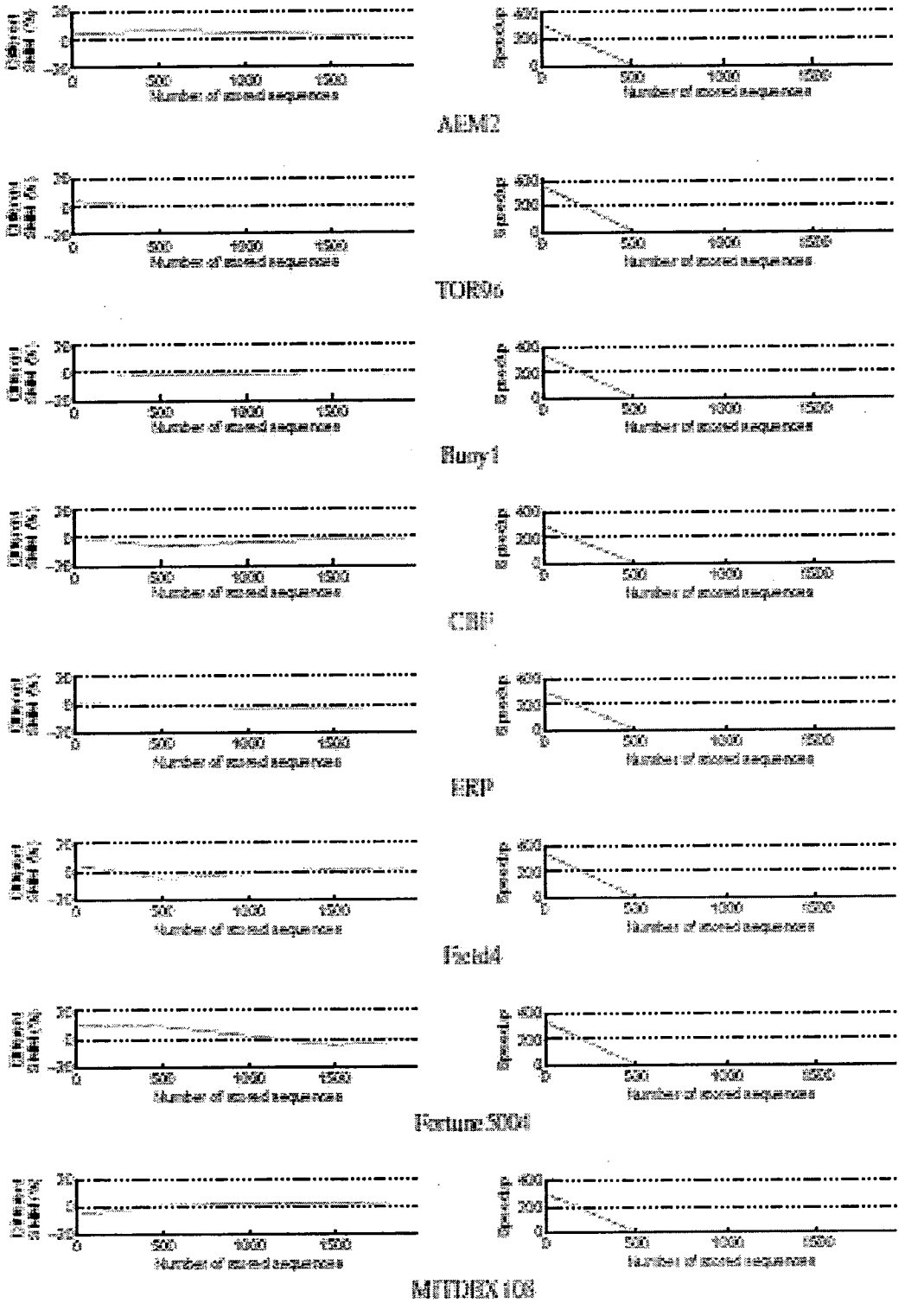
รูปที่ 4. 55 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน CDTW และ complete linkage โดยใช้ค่า k=5, w=64 และจำนวนลำดับย่อยที่ต่างๆ กัน



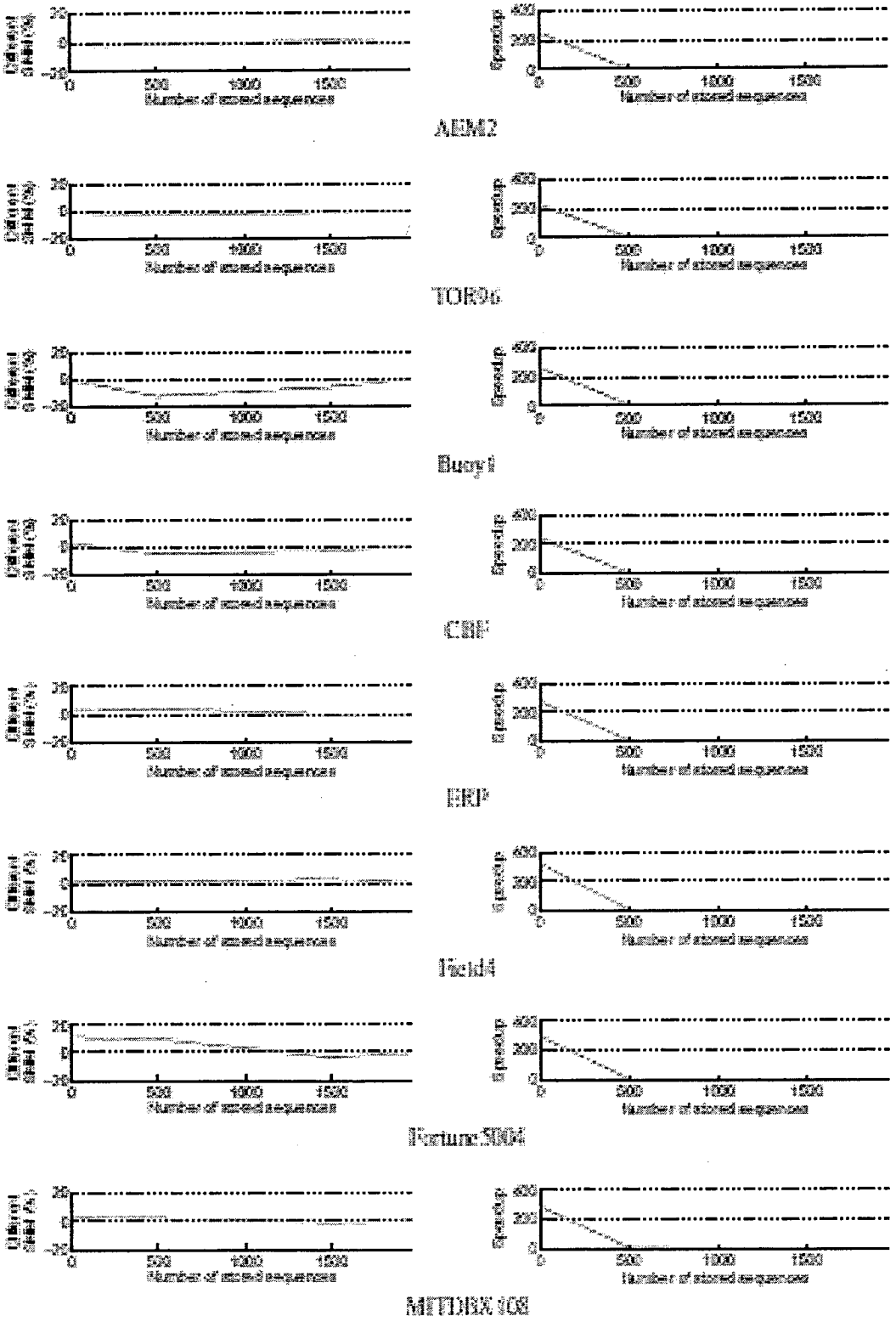
รูปที่ 4. 56 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน CDTW และ complete linkage โดยใช้ค่า k=7, w=64 และจำนวนลำดับย่อยที่ต่างๆ กัน



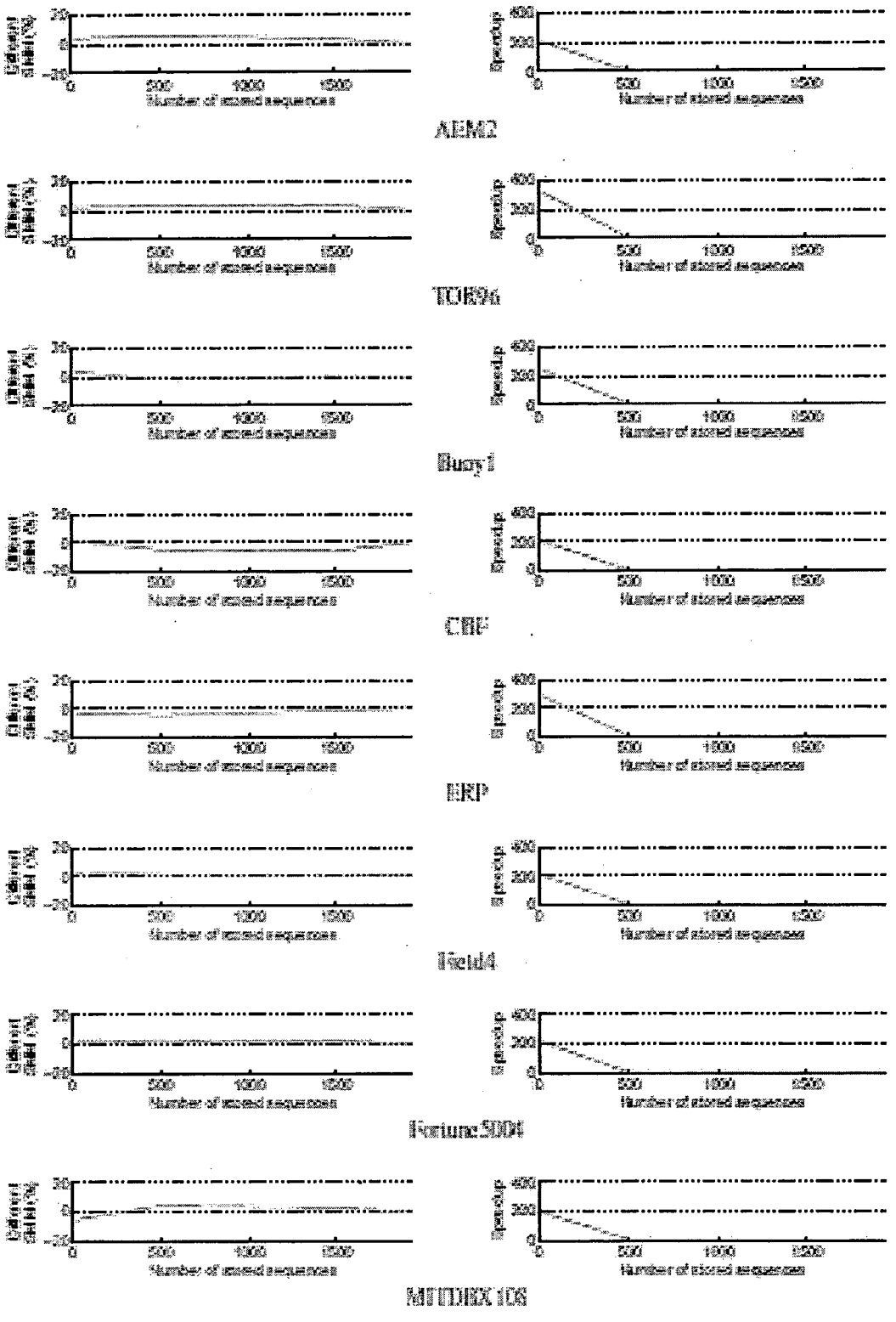
รูปที่ 4. 57 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน CDTW และ complete linkage โดยใช้ค่า k=3, w=128 และจำนวนลำดับย่อยที่ต่างๆ กัน



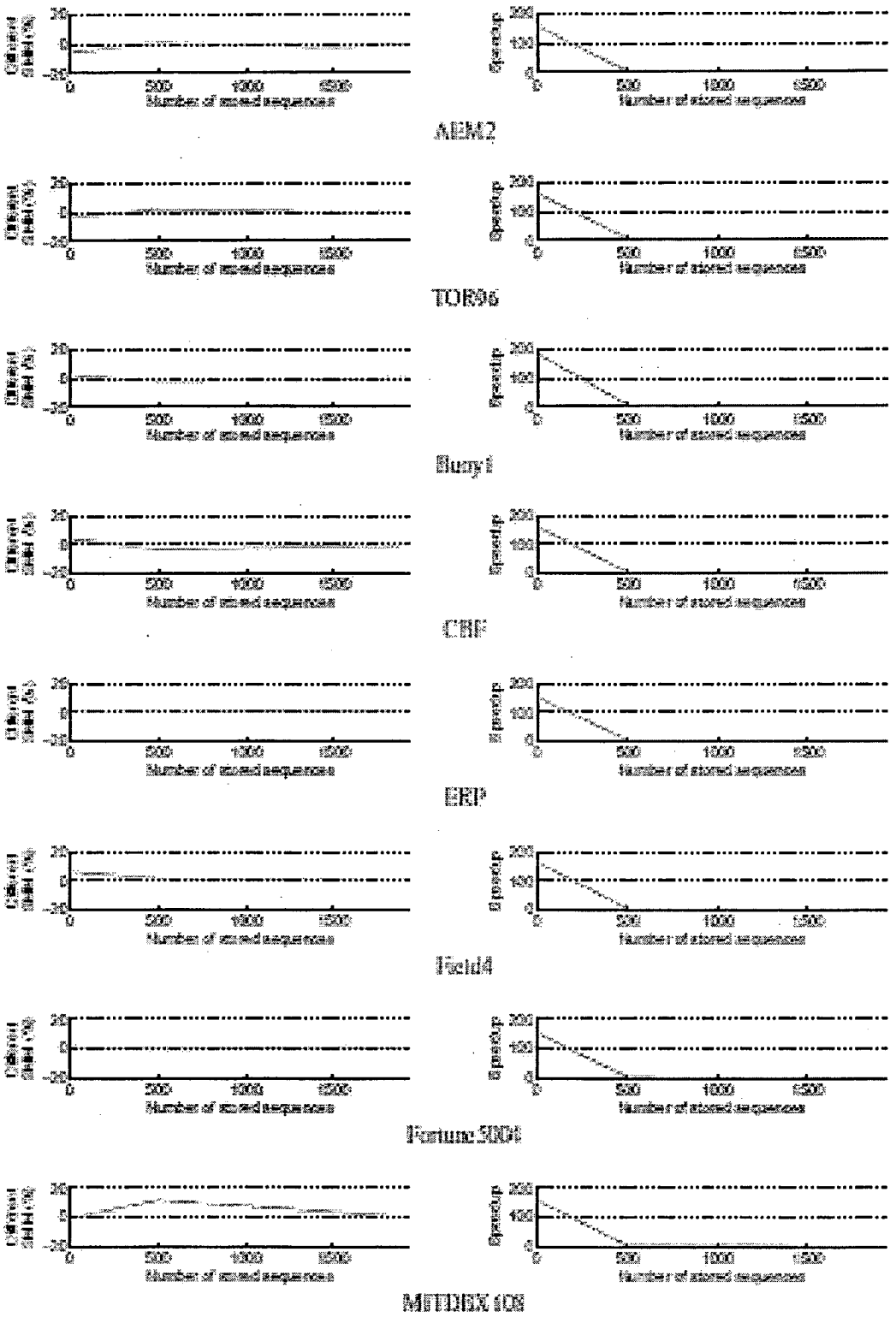
รูปที่ 4. 58 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน CDTW และ average linkage โดยใช้ค่า k=3, w=64 และจำนวนลำดับย่อยที่ต่างๆ กัน



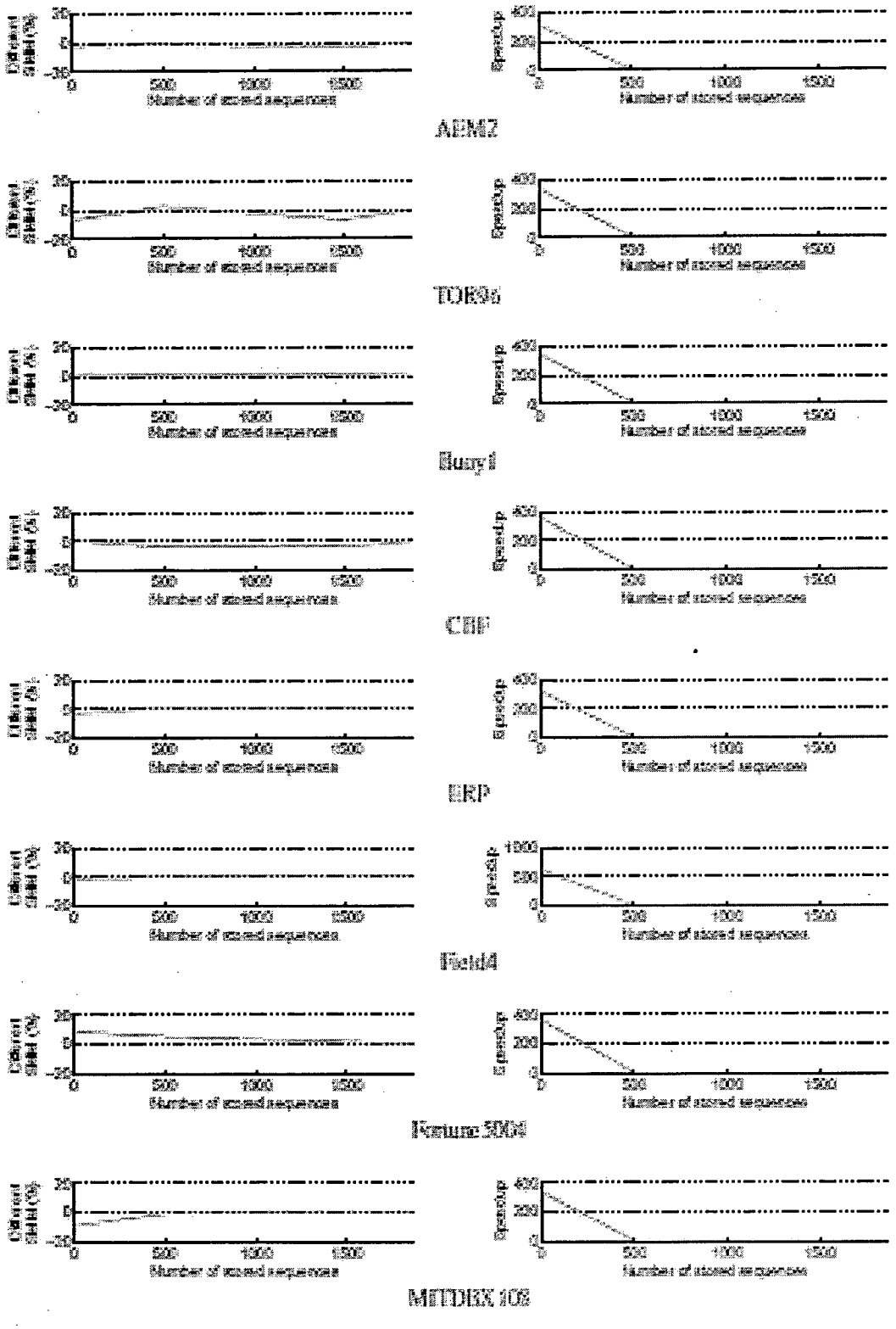
รูปที่ 4. 59 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน CDTW และ average linkage โดยใช้ค่า k=3, w=32 และจำนวนลำดับย่อยที่ต่างๆ กัน



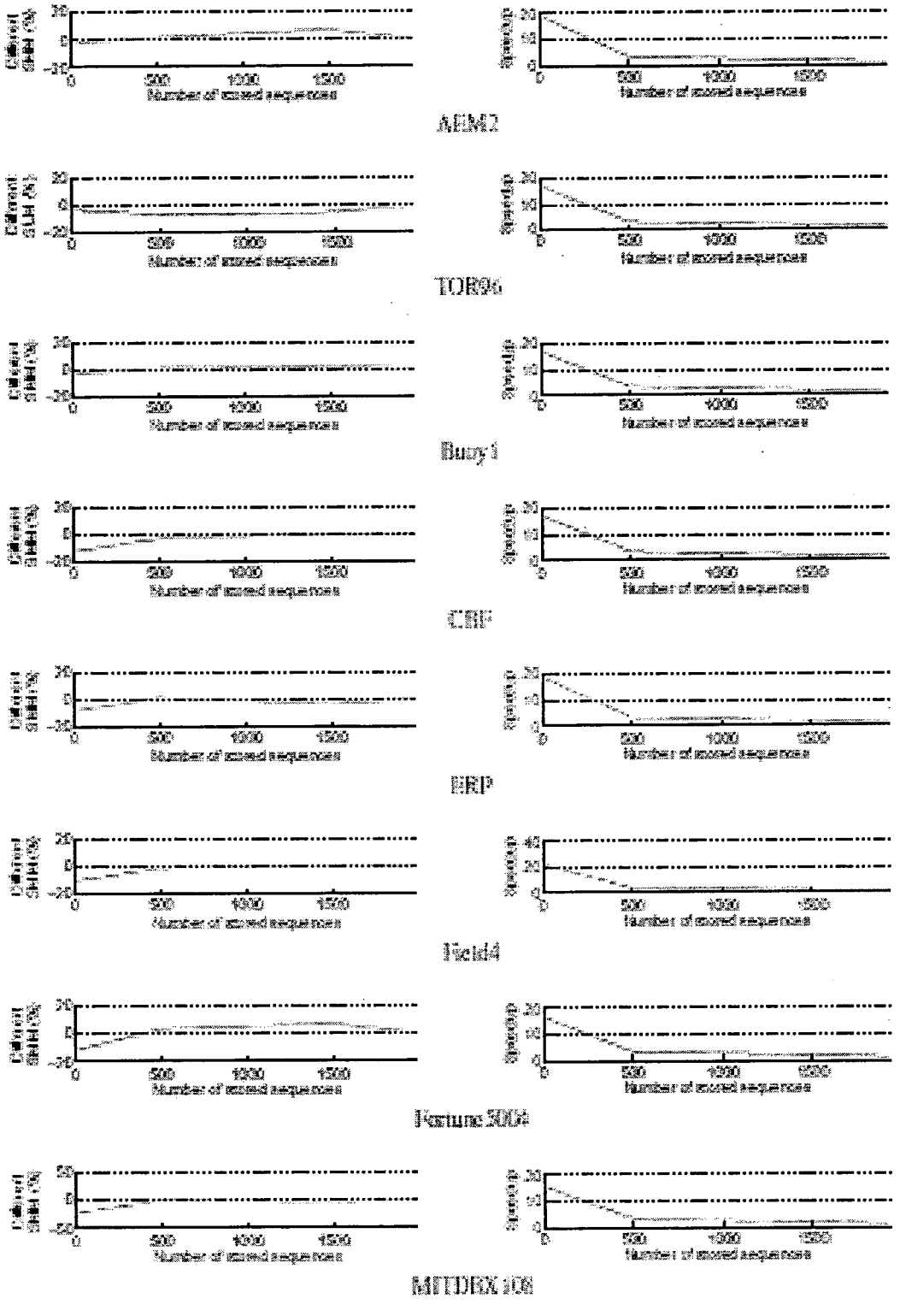
รูปที่ 4. 60 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน CDTW และ average linkage โดยใช้ค่า k=5, w=64 และจำนวนลำดับย่อยที่ต่างๆ กัน



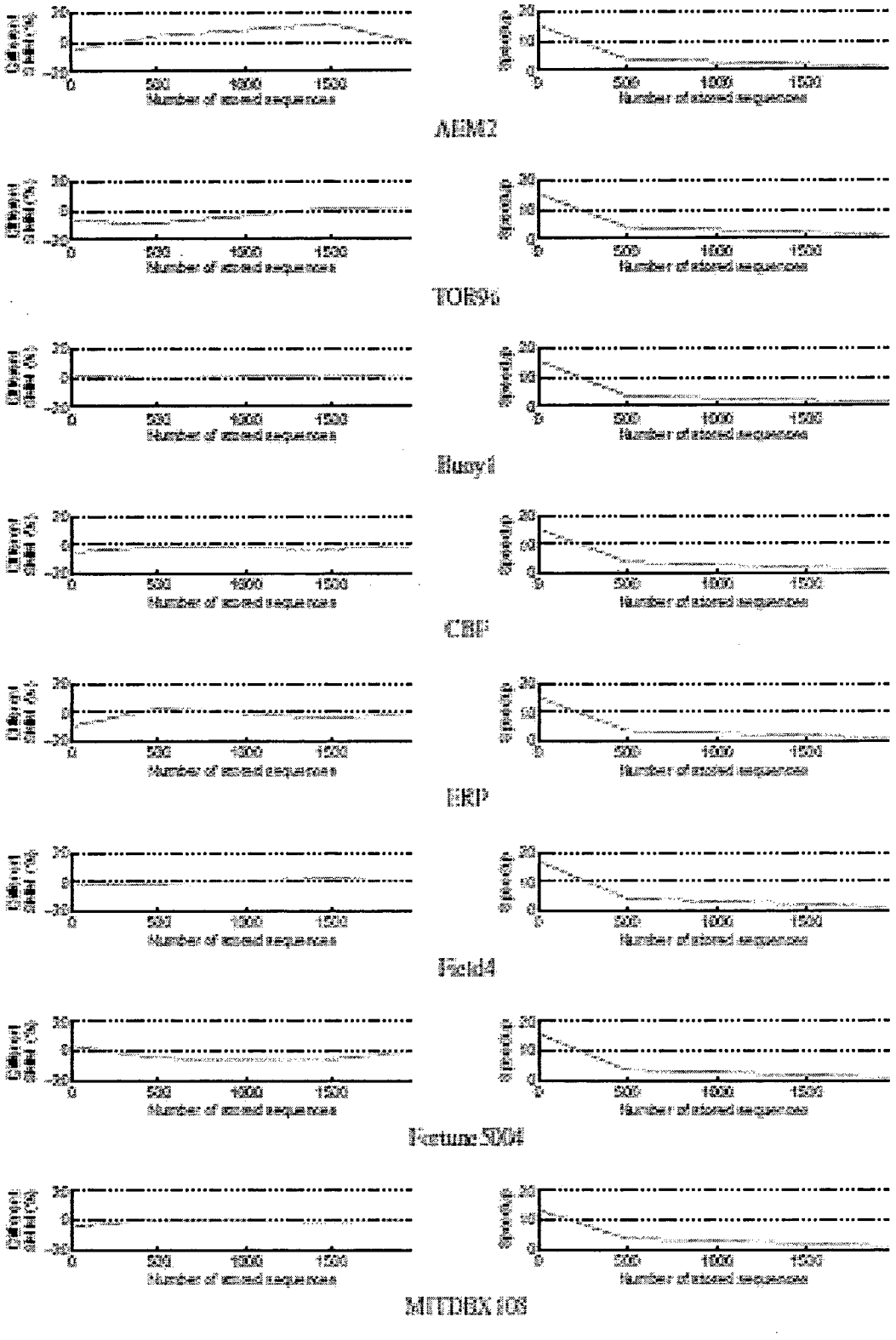
รูปที่ 4. 61 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน CDTW และ average linkage โดยใช้ค่า k=7, w=64 และจำนวนลำดับย่อยที่ต่างๆ กัน



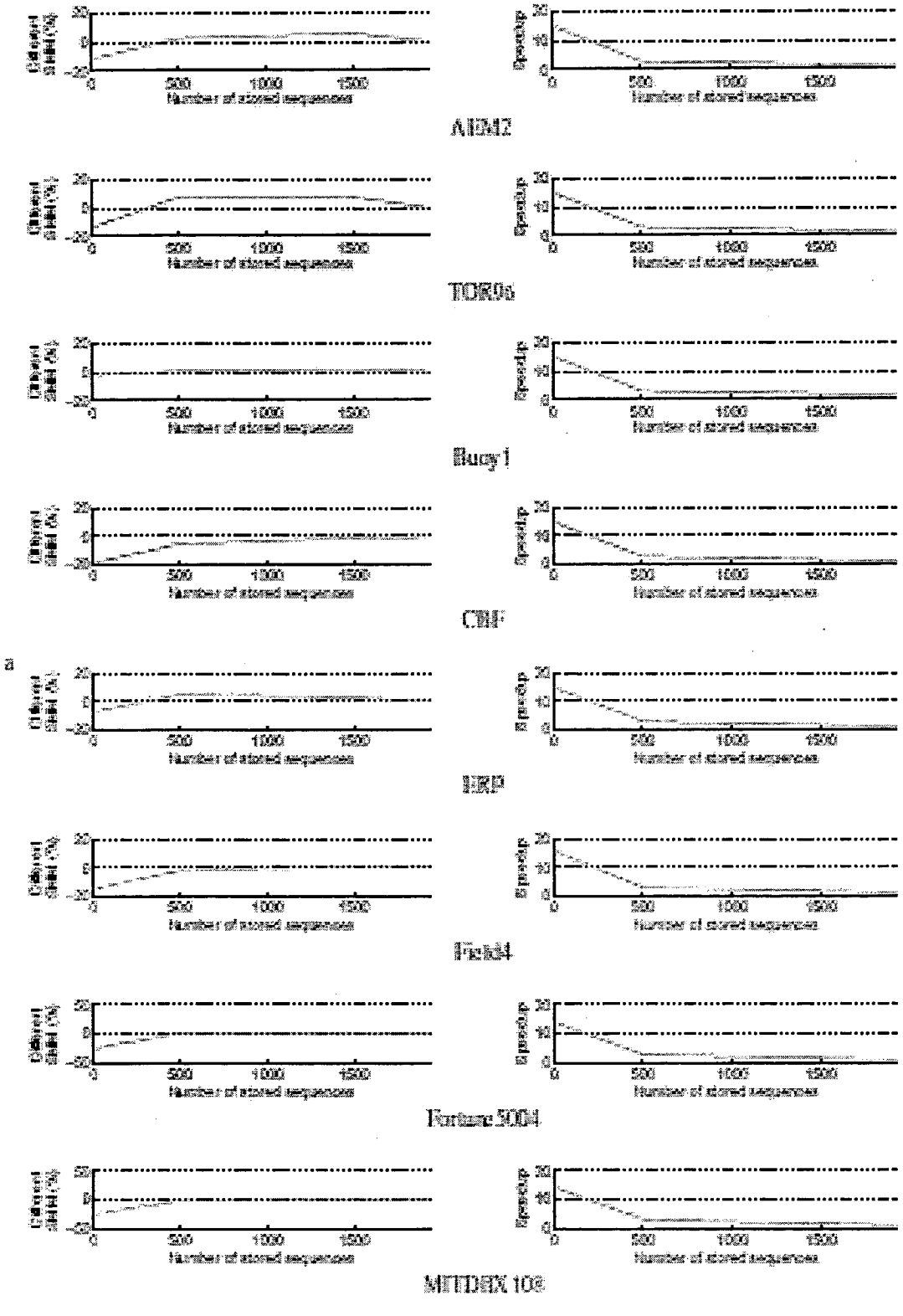
รูปที่ 4. 62 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน CDTW และ average linkage โดยใช้ค่า k=3, w=128 และจำนวนลำดับย่อยที่ต่างๆ กัน



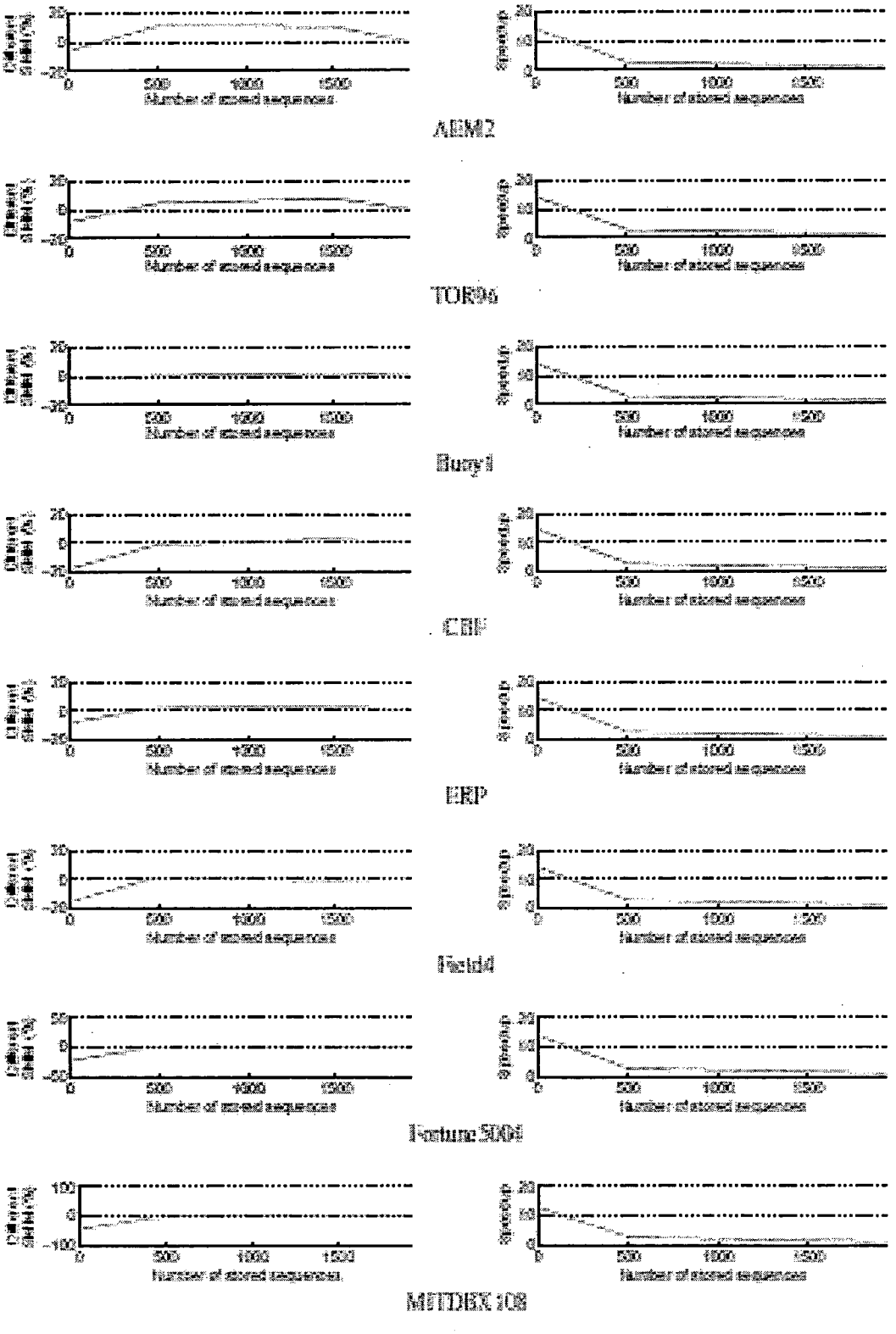
รูปที่ 4. 63 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน ICDTW และ complete linkage โดยใช้ค่า k=3, w=64 และจำนวนลำดับย่อยที่ต่างๆ กัน



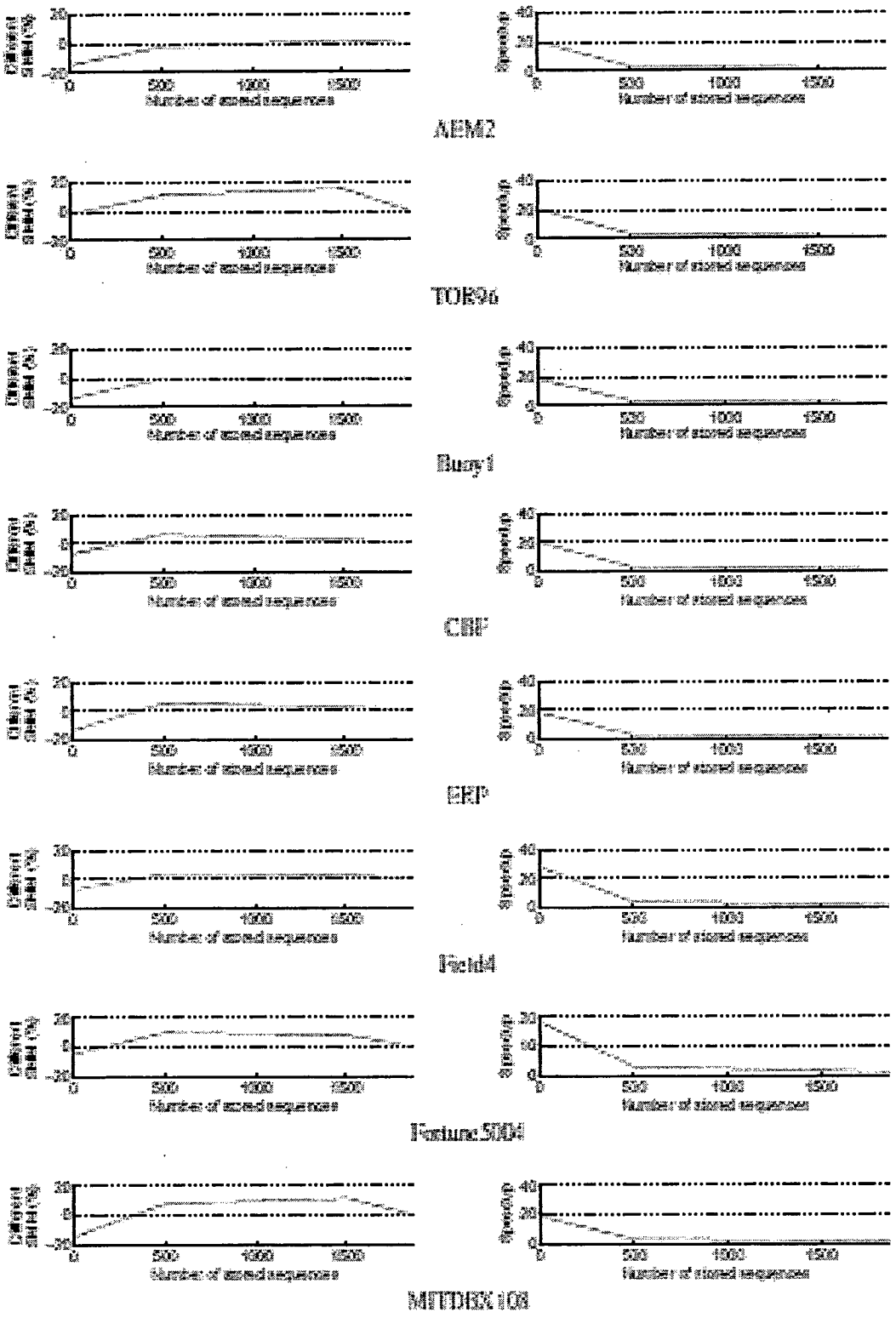
รูปที่ 4. 64 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน ICDTW และ complete linkage โดยใช้ค่า k=3, w=32 และจำนวนลำดับย่อยที่ต่างๆ กัน



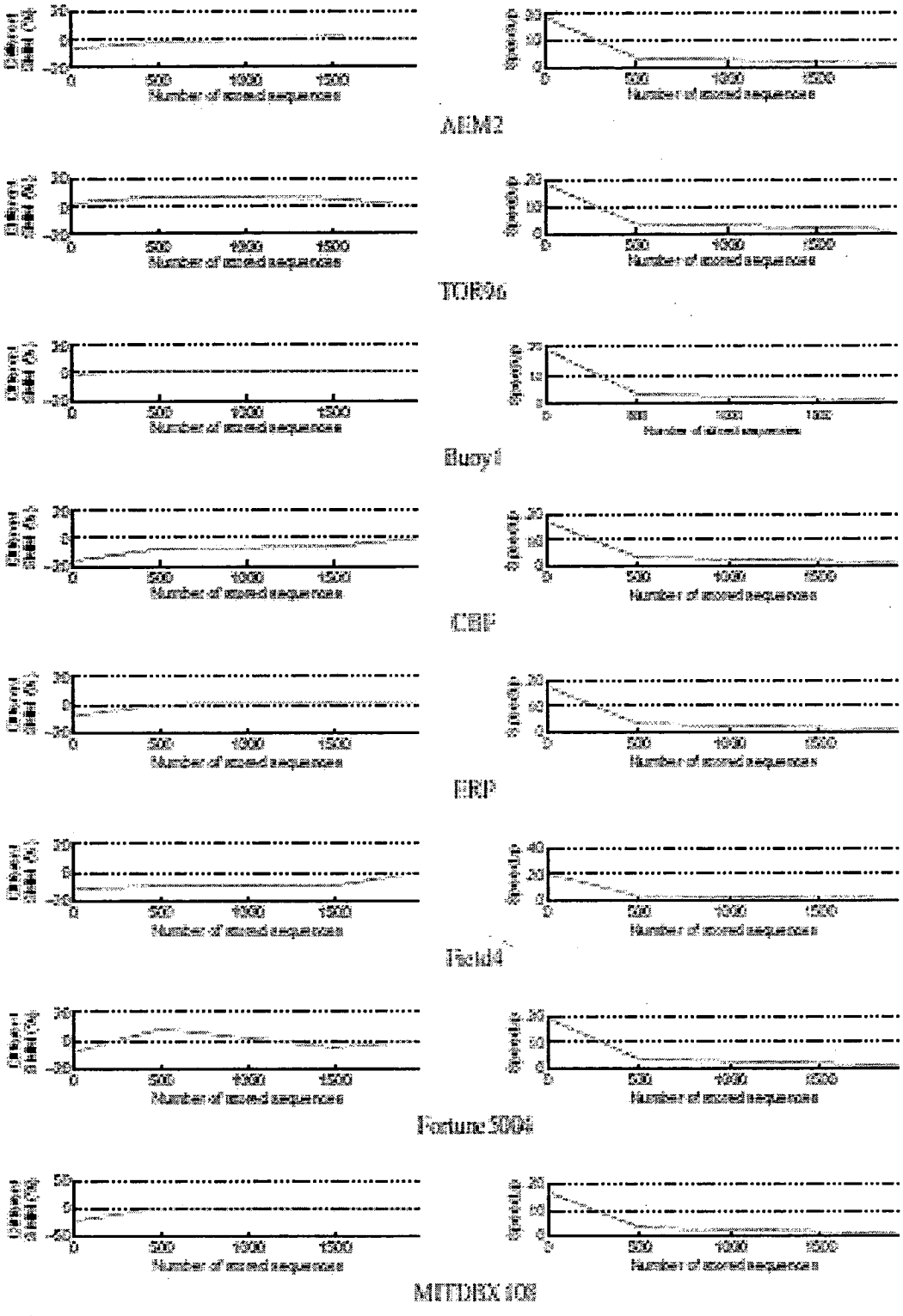
รูปที่ 4. 65 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน ICDTW และ complete linkage โดยใช้ค่า k=5, w=64 และจำนวนลำดับย่อยที่ต่างๆ กัน



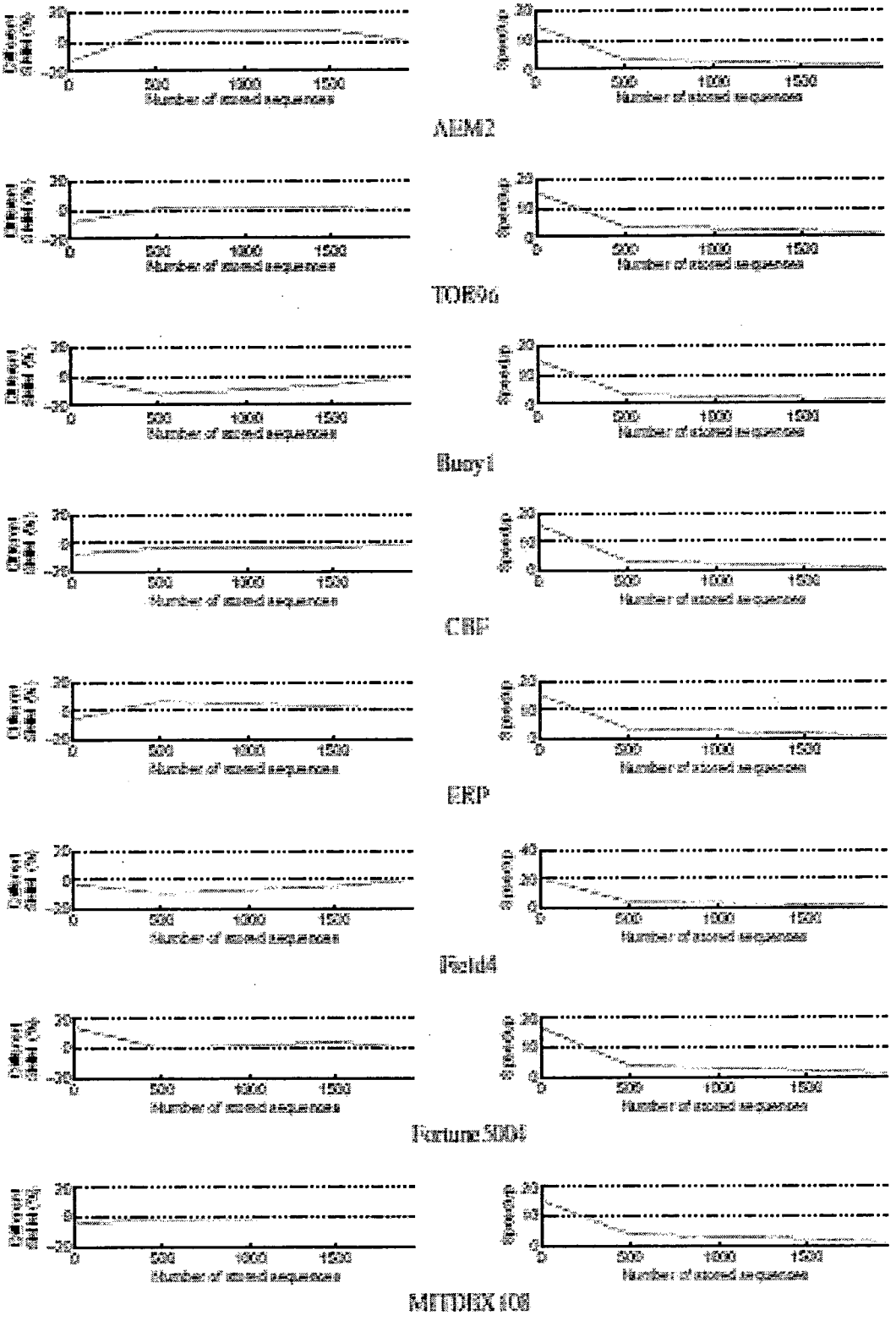
รูปที่ 4. 66 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน ICDTW และ complete linkage โดยใช้ค่า k=7, w=64 และจำนวนลำดับย่อยที่ต่างๆ กัน



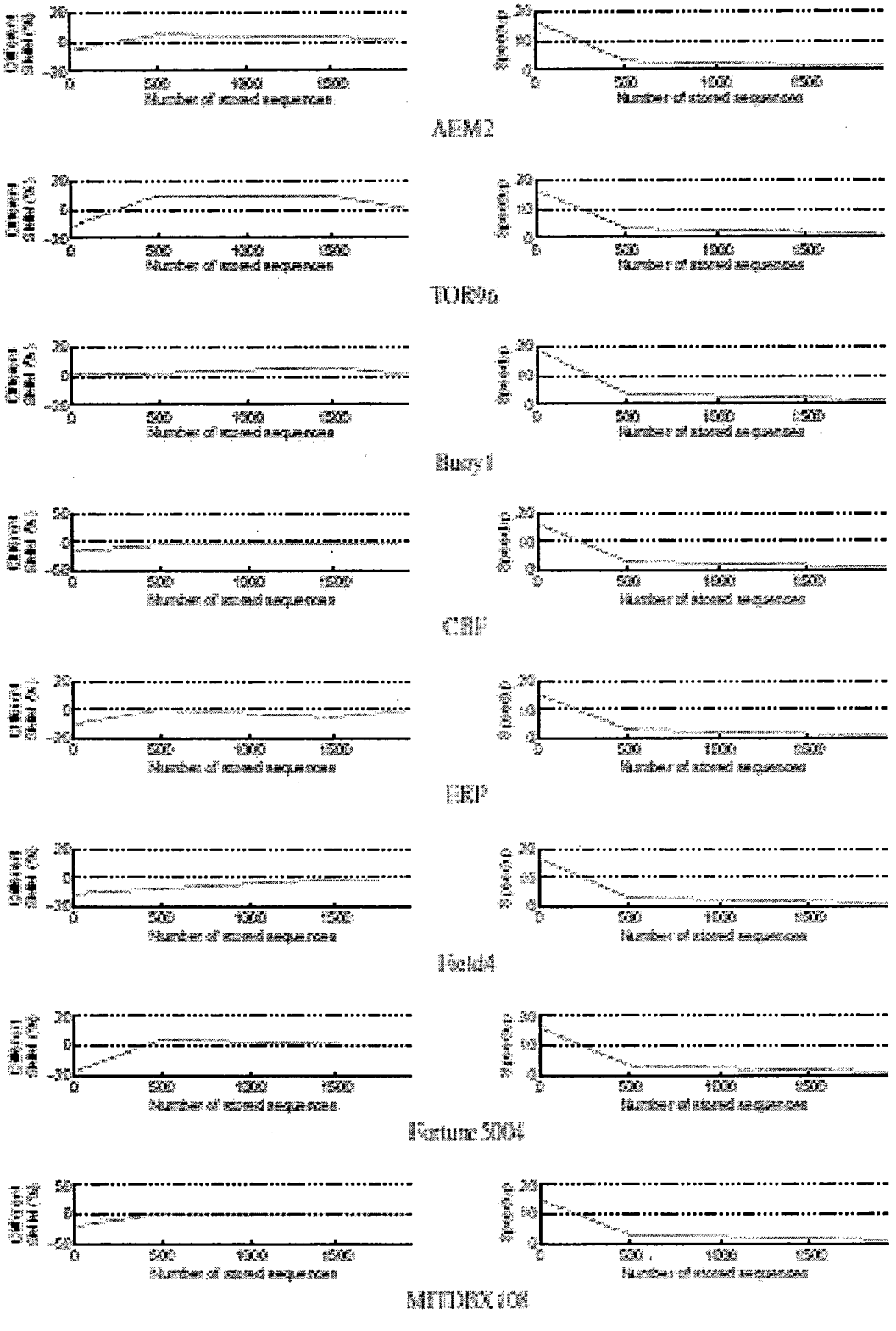
รูปที่ 4. 67 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน ICDTW และ complete linkage โดยใช้ค่า k=3, w=128 และจำนวนลำดับย่อยที่ต่างๆ กัน



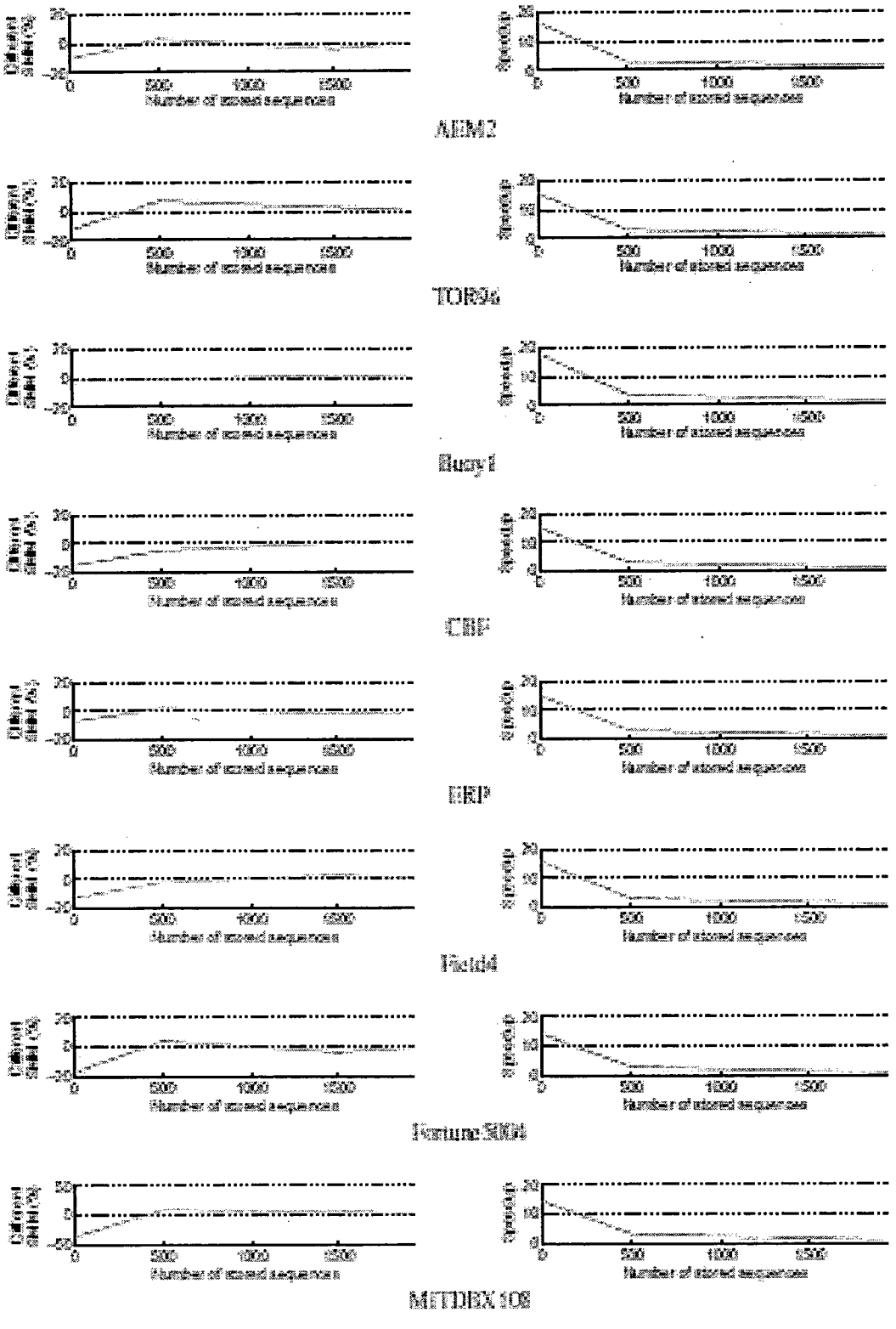
รูปที่ 4. 68 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน ICDTW และ average linkage โดยใช้ค่า k=3, w=64 และจำนวนลำดับย่อยที่ต่างๆ กัน



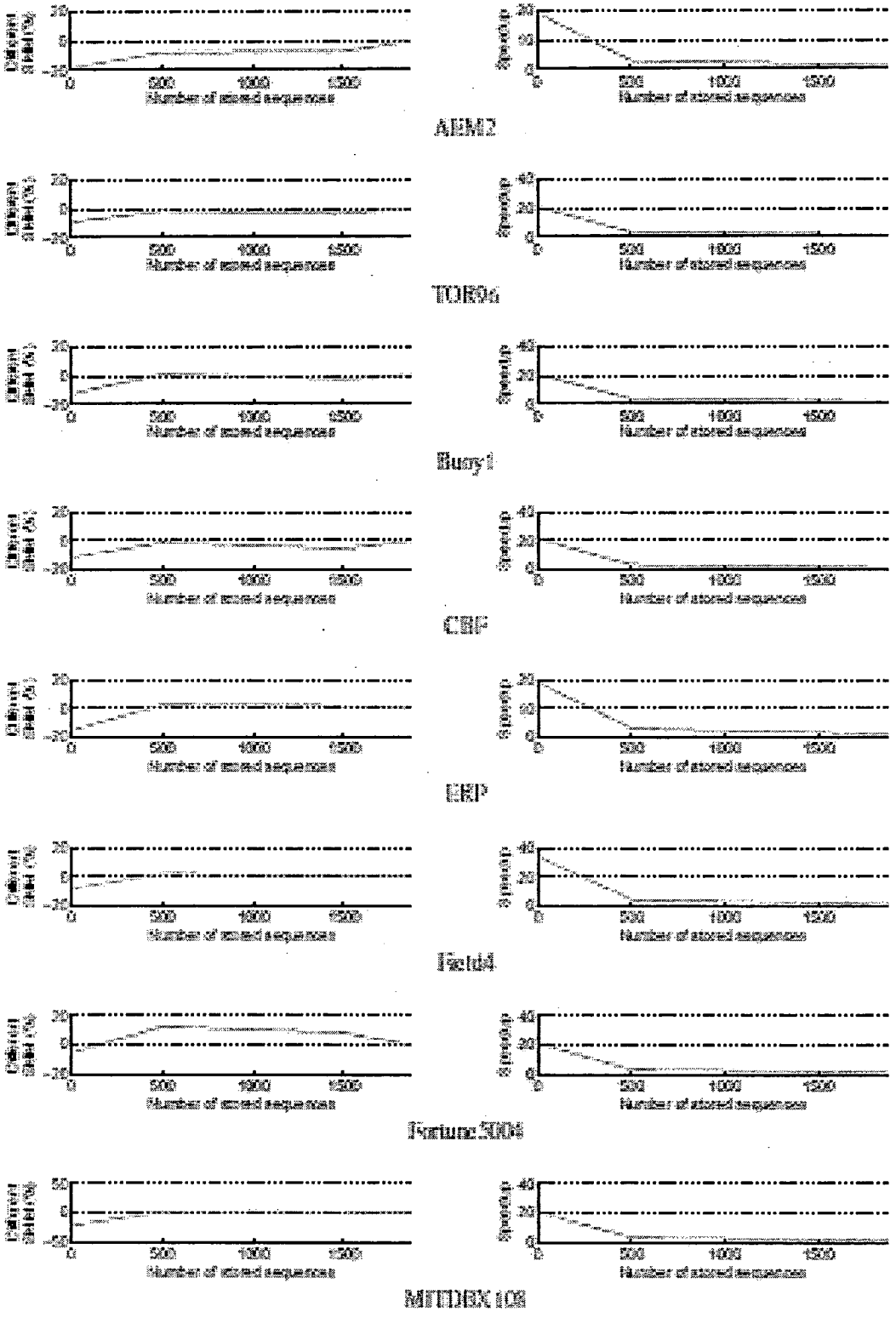
รูปที่ 4. 69 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน ICDTW และ average linkage โดยใช้ค่า k=3, w=32 และจำนวนลำดับย่อยที่ต่างๆ กัน



รูปที่ 4. 70 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน ICDTW และ average linkage โดยใช้ค่า k=5, w=64 และจำนวนลำดับย่อยที่ต่างๆ กัน



รูปที่ 4. 71 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน ICDTW และ average linkage โดยใช้ค่า k=7, w=64 และจำนวนลำดับย่อยที่ต่างๆ กัน



รูปที่ 4. 72 : เปรียบเทียบค่า SMM และ speedup ของอัลกอริทึม 3STSC โดยใช้ฟังก์ชัน ICDTW และ average linkage โดยใช้ค่า k=3, w=128 และจำนวนลำดับย่อยที่ต่างๆ กัน